А.Левенчук

Руководство по системному мышлению

Введение

Системное мышление помогает бороться со сложностью в самых разных проектах: оно даёт возможность выделять своим вниманием всё важное, на время отбрасывая неважное, но при этом удерживать целостность ситуации, учитывая взаимовлияния этих по отдельности продуманных важных моментов. Системное мышление происходит путём мышления моделированием (с текстами на формальных языках или данными в таблицах) и письмом (с текстами на естественных языках, но с отслеживанием типов объектов и видов отношений объектов в этих текстах), поэтому внимание не только наводится на важные предметы, но и удерживается на них всё время проекта: записанное не так легко забыть в суете. А ещё системное мышление даёт возможность заниматься творчеством: в его неформальной части позволяет делать догадки о возможных решениях по изменению мира к лучшему, а в формальной части — критиковать эти решения, чтобы пережившие критику решения взять всерьёз, то есть положить основой для коллективных действий. Системное мышление управляет вниманием многих людей в сложных коллективных проектах, привлекая это коллективное внимание к важному и экономя время путём исключения из мышления неважного.

Для инженеров-менеджеров самых разных специализаций системное мышление даёт возможность надёжно удержать в голове и записях их проекты во всей их цветущей сложности, связать теорию и жизнь. Для опытных инженеров, менеджеров, технологических и корпоративных предпринимателей, исследователей, людей творческих профессий, общественных деятелей системное

мышление позволяет разложить их знание жизни по полочкам. Это мышление-шпаргалка, оно заставит подумать о забытом, а плохой проект похоронит быстро и не даст вложить в него много ресурсов.

У нас не руководство по системной аналитике (часто путают с системным мышлением), ибо системному синтезу в нём уделено не меньше внимания. Системное мышление практично, культурно, стильно, деятельно, ибо создание и развитие систем происходит самыми разными методами, практиками, культурами и стилями работы, видами деятельности/труда/инженерии, но разговариваем мы обо всех этих методах работы при помощи понятий системного мышления. Это фундаментальное, универсальное мышление.

В руководстве также указано, как системно мыслить более-менее одинаково о предметах самой разной сложности, таких как предметы из инертного вещества, живые существа, киберфизические системы, личности, организации, сообщества, общества и даже человечество в целом. Это происходит вследствие того, что в системном мышлении с одной стороны уделяется огромное внимание масштабу объектов, а с другой стороны — сами системные рассуждения не зависят от масштабов объектов, они безмасштабны, просто масштабы являются их предметом. Системное мышление универсально и безмасштабно, поэтому в одном рассуждении можно удерживать внимание на самых разных объектах самого разного масштаба: именно для такой увязки разных объектов в сложных проектах системное мышление и предназначено.

Системное мышление не зависит от того, какие агенты с сильным интеллектом им занимаются — люди ли, AI-агенты или даже коллективные агенты (организации из множества людей, AI-агентов). Оно неантропоцентрично, и наше руководство это учитывает.

Освоение руководства по системному мышлению подразумевает некоторый опыт участия в сложных коллективных проектах из реальной жизни (не простых «учебных проектах» на одного человека, а «настоящих», коллективных рабочих, где результаты кому-то нужны настолько, что за них готовы заплатить!), опыт столкновения со сложностью в жизни лицом к лицу — у осваивающего системное мышление «сложность» относится главным образом к примерам из жизни, а не к «кейсам из учебника», ибо «учебные кейсы» слабо привязываются к жизненным ситуациям. На стажировке по системного мышления на основе нашего руководства вы будете много моделировать (увы, в книжном издании руководства заданий на моделирование нет), то есть не столько «учить системное мышление», сколько практиковать его, «системно мыслить моделированием». Дальше на работе и даже «просто в жизни» вы будете делать ровно то же самое, что на стажировке: заполнять таблички, это и будет ваше системное мышление. Надеемся, что вы даже не будете замечать, что «системно мыслите», по форме это будет просто «заполнение довольно простых табличек, хотя каждый раз над этим заполнением надо будет немного подумать».

Руководство задаёт для мышления определённые «рельсы», которые позволяют после некоторой тренировки быстро и автоматически оценивать ситуацию в реальных коллективных проектах. Системное мышление позволяет лишний раз не «изобретать велосипед» по борьбе со сложностью, вместо трудного и медленного «мыслительного бездорожья» происходит лёгкое и быстрое «мышление по рельсам», беглое задействование лучших придуманных цивилизацией приёмов

мышления. Часть того, что для других людей покажется творчеством, для системно мыслящего человека — это беглое применение мыслительных шаблонов, экономящее время для неизбежного реального творчества. В жизни всё время будут ситуации, которых не было в учебниках по той предметной области, которой вы занимаетесь. Руководство по системному мышлению поможет как раз в этих ситуациях. Как математики не изобретают каждый раз идею интеграла, так и системные мыслители не изобретают каждый раз идею различия функционального и конструктивного описания системы, идею различения целевой системы и системы создания и ещё пару десятков других полезных идей. И это они делают в самых разных проектах, для самых разных систем, соединяя в мышлении ситуации, описанные в самых разных прикладных учебниках.

Основная задача руководства — компактно собрать в одном тексте («свод знаний», по аналогии со «сводом законов» как собрании текущей конфигурации нормативных текстов с учётом всех обновлений, дополнений, исключений, подразумевающий также некоторую полноту охвата предметной области и официальный статус — с этим разбирается методолог, это методологическая работа) и объяснить (преодолеть «понятийное расстояние»[1] для понимания, работа методиста) «мыслительный минимум» по системному мышлению, обычно рассыпанный по самым разным источникам знания. Специфика нашего руководства в том, что его содержание базируется не столько на традиционной академической литературе прошлого века по общей теории систем или традиционных учебниках для менеджеров, сколько на современной инженерной и научной литературе 2017-2023 годов, а также в чуть меньшей степени на международных стандартах и публичных документах системной инженерии и инженерии предприятий, разработанных или обновлённых за последние пять-шесть лет (ISO 15288:2023, ISO 42010:2022, ISO 15926-2:2003, IEC 81346-1:2022, OMG Essence:2024), причём тут в нашем руководстве делается поправка на то, что эти международные стандарты и публичные документы идеологически связаны с крупными правительственными проектами с госфинансированием, поэтому там даются определения важных понятий, но переутяжелённый инженерный процесс. Мы каждый раз указываем на то, что не следует брать из этих стандартов в коммерческие инженерные проекты.

В руководстве учтены последние результаты исследований по физическим основаниям биологии, включая объяснительные теории эволюции, происходящей на разных системных уровнях. Это представлено в нашем руководстве не слишком подробно (подробней это можно найти в руководстве по интеллект-стеку), но даны все необходимые ссылки на литературу, чтобы дальше любознательным инженерам-менеджерам можно было разобраться самостоятельно. Материал руководства даёт безмасштабную (от молекул до человечества в целом) версию системного мышления, а также деантропоморфизированную (нет предположения, что рациональный агент — это обязательно человек-индивид) версию методологического мышления.

Руководство подготовлено на основе опыта преподавания системного мышления как в многочисленных вузах (ежегодно с 2012 года), так и в системах повышения квалификации инженеров, менеджеров, технологических предпринимателей, преподавателей и людей самых разных других занятий, в том числе и особенно в Мастерской инженеров-менеджеров (МИМ, ранее Школе системного менеджмента,

[1] http://wiki.lesswrong.com/wiki/Inferential_distance

ШСМ)[2], где автор является научным руководителем. Стажировка по нашему руководству подразумевает прохождение большое количества вопросников с авторскими ответами, а также выполнение заданий на системное моделирование (моделирование ситуаций в рабочих проектах стажёров). Наш опыт показывает, что само по себе чтение руководства как «учебника» не учит системному мышлению, а вот системное мышление само по себе (в формате мышления моделированием с использованием структурированных текстов и мышления письмом с использованием неструктурированных текстов на естественных языках) учит, если оно основывается на использовании руководства именно как руководства: для выполнения работы, в том числе выполнения заданий в ходе стажировки.

Изложение системного подхода даётся универсально для инженеров «железных» и программных систем, менеджеров и владельцев бизнесов, деятелей культуры, политиков и людей, которые занимают самые разные другие прикладные трудовые роли. Но мастерство методологии, системной инженерии, инженерии личности, системного менеджмента надо получать из отдельных руководств по этим методам, освоение системного мышления для этого обязательный пререквизит. А ещё нужно будет отдельно осваивать самые разные прикладные методы работы, для них будут отдельные прикладные руководства. «Как мне быть в моём проекте вот с такой проблемой» обычно решается с помощью системного мышления, организующего применение прикладного мастерства в самых разных видах труда, но не заменяющего само это прикладное мастерство. Системное мышление позволит вам быстро разобраться с вашими проблемами игры на рояле, но вот играть на рояле учиться всё равно придётся — и то же относится, например, к инженерии робототехнических систем. Проблемы вы будете решать быстрее, но учиться делать эти системы всё равно придётся, чудес не бывает.

Чтобы усилить свой интеллект, вам нужно освоить работу по множеству руководств для инженеров-менеджеров, ибо одно руководство по системному мышлению не закрывает весь интеллект-стек фундаментальных методов мышления, (системная инженерия, методология, риторика, этика, эстетика, познание/исследования, рациональность, логика, алгоритмика, онтология, теория понятий, физика, математика, семантика, собранность, понятизация). Но руководство по системному мышлению уточняет много важных мест из этих методов, поэтому руководство вполне можно отнести к набору руководств по усилению интеллекта.

Текст руководства в этом издании был фактически переписан в десятый раз: два первых варианта были в 2014 и 2015 году для курса «Системноинженерное мышление», затем книга «Системное мышление» с обобщением для инженерного, менеджерского и предпринимательского труда вышла в феврале 2018 года в Ridero (и одновременно вышел курс в Coursera), четвёртая переработка была в июле 2019 года (было вписано порядка 120 новых страниц и упрощена терминология), и был открыт онлайн-курс в Школе системного менеджмента. Пятая переработка была в августе 2020 года, в ней был учтён опыт ведения занятий и выхода курса «Образование для образованных 2020», где уточнено положение системного мышления в общем интеллект-стеке с другими фундаментальными методами в составе интеллект-стека. В шестой версии 2021 года была опять упрощена терминология и внесены изменения для поддержания актуальности материала.

В седьмой редакции 2022 года основное изменение — отделение материала про «не

[2] http://system-school.ru/

жизненный, не цикл» в отдельный курс методологии, а также дополнение курса материалами по безмасштабному и деантропоморфизированному мышлению, подкреплённому современными находками не только инженерии и менеджмента, но и физики с биологией. Курс получил название «Практическое системное мышление», чтобы отразить проактивный/enactive характер деятельности агентов, меняющих мир. Это изложение уже опиралось на третье поколение системного подхода. Деятельность/труд/практика/инженерия в этой версии курса стали синонимами, относящимися к работе систем создания над целевыми системами.

После выпуска седьмой редакции в 2022 году были разработаны курсы «Методология», «Системная инженерия» и «Системный менеджмент», в которых были отражены изменения в методах системной инженерии и менеджмента, произошедшие в последние годы: прежде всего переход к «непрерывному всему», исчезновение инженерии требований, окончательное отделение архитектуры от разработки (в том числе разделение концепции системы и архитектуры). Поэтому в январе 2023 года текст «Системного мышления» был переписан в восьмой раз, чтобы привести его в соответствие с новыми методами системной инженерии и менеджмента. Вернулось более короткое название «Системное мышление», исчез раздел «Требования и архитектура», появился новый раздел, отражающий вместо глоссария онтологию третьего поколения системного подхода. Уточнена онтика сервиса.

В мае 2023 были внесены ещё изменения, по факту начата девятая переписка: поведение (процессы) какой-то системы прямо запрещено считать системами (но, если очень надо, рекомендуется собрать изо всех взаимодействующих в ходе этих процессов систем новую систему, у которой тем самым появятся новые свойства, и она будет вести себя по-новому в силу эмерджентности). В связи с этим переписан пример социального танца (слово «танец» табуировано, танцует теперь танцор). Театральная метафора стала метафорой ролевой игры. Исправлено множество опечаток и мелких ошибок. Исправлено название курсов «Онтологика и коммуникация» и «Собранность» на объединённое «Моделирование и собранность». И даже вернулось старое название курса «Системное мышление» (пару лет курс назывался «Практическое системное мышление»). А в декабре 2023 года переписка продолжилась в целях улучшения понимания, добавилось множество вопросов, упражнений, заданий. Учтено появление курса «Инженерия личности». Курс был разбит на более мелкие разделы, их стало не 8, а 12. Курс «Моделирование и собранность» был переименован в «Рациональную работу».

Внесено множество уточнений и разъяснений: уточнено понятие аффорданса, дана «системная мантра», предостережено о возможности путаницы разных типов отношений для подролей — специализация и композиция, основным термином вместо «практики» стал «метод» и добавлена синонимия метода (культура, стиль, стратегия и даже сервис), введено понятие «предмет метода/сервиса» и его синонимия с альфами, термин «технологии» табуировали и сменили на «инструменты», добавили синонимию дисциплины/теории/объяснения/знания/алгоритмы, интеллект-стек теперь стек методов мышления (а не дисциплин), уточнено понятие техно-эволюции, предложено четыре основных альфы вместо трёх. Объём текста увеличен в ходе переписки примерно вдвое, главным образом за счёт дополнительных разъяснений и примеров. Добавлены также новые задания. В исследовательском разделе «12. Вместо глоссария: онтология системного подхода третьего поколения» даётся

гипотеза, что четвёртое поколение системного подхода появится после того, как появится объединённая классическая семантика на базе семиотических (знаковых, локальных) представлений и распределённых (как в нейросетях) представлений.

В ноябре 2024 года началась работа над десятой редакцией текста, граф создания стал графом создателей, появился подраздел с примерами групповых объектов как системы, внесено больше двухсот исправлений по присланным замечаниям и вопросам к тексту. В мае 2025 года «курс» стал «руководством» (что потребовало внесения исправлений в примерно тысяче мест по тексту), чтобы стать ближе к рабочему документу, а не учебному. А ещё в текст руководства внесены добавления по тому, как понимать системную мантру/канву/чеклист/сюжет. Это стало десятой версией текста. Руководство вполне рабочее, предназначено для использования прежде всего в рабочем контексте, на предприятиях и организациях, в проектах разного масштаба, хотя его по-прежнему можно использовать в вузах в качестве учебника.

Руководство вводит понятия концепции использования системы и концепции системы, архитектуры, обоснований успешности, конфигурации, управления работами, других традиционных и относительно новых понятий системной инженерии и системного менеджмента, непосредственно следующих из системного подхода. Но руководство не рассказывает о том, как разработать качественные концепцию системы и архитектуру, как тщательно обосновать успешность системы, то есть руководство не содержит описания методов современной системной инженерии и прикладных методов инженерии киберфизических систем, программной инженерии, системного менеджмента, культурного строительства, технологического предпринимательства, социальной инженерии. Изучение таких методов даже на кругозорном уровне обычно требует дополнительных долгосрочных усилий и отдельных руководств. Руководство по системному мышлению может быть освоено в ходе программы стажировки по рабочему развитию, куда входит в ряд других руководств. В этой программе стажёр осваивает методы в том, как:

- **наводить своё внимание** и внимание других **агентов** на объекты из явной спецификации (онтологии), то есть наводить внимание понятийно, а не спонтанно, а также **удерживать** понятийно наведённое внимание на самых разных временных масштабах, в том числе включая коллективное внимание в таких агентах как команда или целое предприятие путём лидерства, или влияния на сообщество или общество (руководство по рациональной работе),
- наводить внимание на **системы** (наше руководство по системному мышлению, о котором мы сейчас говорим),
- наводить внимание на **методы,** которыми ведут работы **системы-создатели** (руководство по методологии),
- структурировать **деятельности и роли**, которые обязательно удерживаются во внимании в ходе **проектов системной инженерии**, (руководство по системной инженерии),
- обучать **мастерству** выполнения деятельности (руководство по инженерии личности),
- обращать внимание себя и **организации** (команды, коллектива, предприятия) на объекты, описываемые **специализацией** системной онтологии для таких систем-создателей, как организации (руководство по системному менеджменту).

- усиливать свой **интеллект/мыслительное мастерство**, используя лучшие на сегодня (SoTA, state-of-the-art) методы мышления (руководство по интеллект-стеку).

Руководство требует от инженеров-менеджеров знания английского языка. Сам текст руководства/guide на русском, но большинство ссылок на первоисточники даётся на англоязычные материалы: первоисточников много (около 400 отсылок к литературе), и они современные, их не успели перевести и издать на русском языке.

В руководстве принят способ изложения, в котором наставления повторяются несколько раз в разных местах текста — с разным уровнем подробностей. Принцип «сказано один раз в одном месте» (как в энциклопедиях и справочниках/handbook или пошаговых инструкциях/manuals) в руководстве (guide/руководство/путеводитель, акцент на разъяснения направление к правильным и важным местам) намеренно не соблюдается. Это увеличивает объём текста, но заметно облегчает освоение мастерства системного мышления за счёт интервального повторения наставлений. Общий объём руководства с учётом заданий примерно 2.5 Мзнака с пробелами, так что руководство короче, чем вполне посильный для многих поколений школьников даже не в выпускном классе роман «Война и мир», в котором 3.0 Мзнаков[3].

После освоения системного мышления по нашему руководству мы рекомендуем освоить методологию по одноимённому руководству, а затем освоить работу по инженерным методам согласно руководствам по системной инженерии, инженерии личности, системному менеджменту. Это направление можно назвать **«дьявол в деталях»**: повышать своё мастерство инженерии-менеджмента по отдельным инженерным, менеджерским, творческим, социальным методам работы, то есть повышать прикладное мастерство. Это традиционные карьерные ходы профессионализации в инженерии, менеджменте, другим прикладным специальностям в их связи с реальной жизнью. Системное мышление позволит удерживать целостность изучаемого набора кругозорных и прикладных методов работы, а также переносить накопленный опыт из проекта в проект. Это мастерство инженера, менеджера, предпринимателя/основателя компаний, деятеля искусств, общественного деятеля/политика и т.д.: человека, который овладел системным мышлением и использует его для изменения мира к лучшему.

Но можно предложить и альтернативный вариант, **«ангел в абстракциях»** («знание принципов освобождает от знания фактов»): обобщить предлагаемое системное мышление с целью поднятия беглости в использовании его приёмов и распространения его на самые разные виды систем — для экспансии системного мышления на новые методы работы, новые классы систем (например, системы машинного обучения и искусственного интеллекта, системы из молодёжных субкультур и т.д.). По этому направлению можно углублять свои знания интеллект-стека фундаментальных методов мышления, изложенные в отдельном руководстве. Это образование человека, который занимается бесконечным познанием/исследованиями, бесконечным развитием, выход с квалификаций по рабочему развитию на квалификацию исследователя. Хорошим подспорьем на этом пути будет изучение литературы, которая упоминается в руководстве по интеллект-стеку, особенно литературы из последнего раздела руководства.

Активное участие в подготовке наставлений руководства по системному мышлению

[3] https://otvet.mail.ru/question/54668490

приняли преподаватели, аспиранты и студенты кафедры технологического предпринимательства МФТИ, директорат Русского отделения INCOSE, сообщество мастерской инженеров-менеджеров (МИМ, ранее Школы системного менеджмента). Без их активного участия это руководство вряд ли появилось бы на свет в его текущем виде.

Большое спасибо за принципиальные вопросы, получившие отражение в руководстве, участникам учебных программ по системной инженерии в УрФУ, МФТИ, МИФИ, СФУ, МИРЭА-РТУ, участникам стажировок и отдельных разборов/review рабочих ситуаций в мастерской инженеров-менеджеров (МИМ, ранее ШСМ). Руководство неоднократно обсуждалось на заседаниях Русских отделений INCOSE и SEMAT, автор выражает благодарность членам этих международных организаций за многочисленные замечания и предложения. Много ценных замечаний было представлено читателями блога автора (http://ailev.livejournal.com, трансляции блога есть в телеграме, мордокниге, вконтакте, фрифиде). Спасибо Роману Варьянко за оперативную корректуру и ценные содержательные замечания, Виктору Агроскину за помощь в отладке онтологии руководства, Юлии Чайковской за предложение удачных терминов.

В телеграм есть чат поддержки руководства, он общий для всей программы рабочего развития мастерской инженеров-менеджеров: https://t.me/systemsthinking_course

В издательстве Ridero по курсу выпущен двухтомный (ввиду большого объёма) вариант руководства в формате курса[4], его можно использовать в качестве сувенирной продукции для любителей электронных и бумажных книг, а также как подарочный вариант. Текст книг меняется реже текста руководства (невозможно быстро заменить книгу), в книге нет вопросов с авторскими ответами. Осваивать руководство и сверяться с ним в работе мы рекомендуем по онлайн-варианту.

Оглавление

Введение 1

1. О мышлении 15

Понятия, используемые в системном мышлении 15

Системное мышление как сильное/универсальное/общее 19

Эмерджентность и разделение труда 25

Особенности подачи учебного материала 28

Вопросы. Как устроен текст руководства? 35

Разные мышления 36

Вопросы. Типы интеллекта и мышления 40

Критерии сильного мышления 41

Заменяет ли системное мышление прикладное мышление? 47

Место системного мышления среди других мышлений: интеллект-стек 49

[4] Том 1: https://ridero.ru/books/sistemnoe_myshlenie_2024_tom_1/, Том 2: https://ridero.ru/books/sistemnoe_myshlenie_2024_tom_2/.

Вопросы. Системный анализ, необходимость прикладного мышления. 55
Состав интеллект-стека 56
Интеллект-стек и системное мышление 61
Моделирование: понятия. 68
Мыслительный минимум современного человека: учим один раз, используем во всех проектах 69
Моделирование мыслительного мастерства 76
Задания: мышление и интеллект-стек. 77
2. Наш вариант системного мышления: третье поколение 78
Варианты системного подхода. 78
Моделирование: пропущенные системные уровни 87
Системная инженерия: инженерия на основе системного подхода 88
Вопросы. Успешность системы. 99
Системность — это не систематичность 99
Опора на стандарты и публичные документы системной инженерии и менеджмента в нашем варианте системного подхода 101
Основные понятия системного подхода 104
Терминология 106
Слова-термины важны и не важны. 110
Определения: гробик для умершей мысли 112
Задания: обходимся без определений и не спорим о терминах. 116
Как выбирались термины для наших руководств 116
Вопросы. Терминология 119
3. Творчество: догадки и их критика. 121
Формальность системного мышления 121
Системное творчество. 124
Вопросы: аффордансы. 131
Задание: эволюция. 132
Предметные специализации системного мышления 132
Моделирование. Предметные области 142
Можно ли научить мышлению? 143
Метанойя. 146
Задания. Контринтуитивность и метанойя. 150
Состояния ученика в ходе обучения мышлению 150
Особенности ответов на тренажёрные вопросы (quiz) 157
Переход к использованию мышления 162
Задания: невидимость системного мышления для окружающих. 168

Искушение сложностью 168
Моделирование: выбор рабочего проекта 174
Применимость системного мышления: везде и всегда 175
Задания: системное творчество 179
4. Воплощение и описание системы 180
Воплощение, описание и документация системы 180
Моделирование: воплощения, описания, документы с описаниями 188
Описания 188
Моделирование: присвоение типа системе по её описаниям 195
Как договориться: не обобщать, а конкретизировать 195
Моделирование: заземление абстрактных текстов 197
Отношение состава 197
Вопросы. Воплощение и описание систем 202
Отверстия 206
Поведение системы: процессы/изменения 208
Поведение — это не система 212
Вопросы. Процессы как физические объекты 219
Компьютерные программы как системы 220
Целевая система в проектах разработки софта 226
Мастерство как система 230
Вопросы на физичность отношения состава 235
Задания: системы в физическом мире 237
5. Методы/функции/практики/деятельности/культуры/методы работы и роли 237
Машинка типов 237
Ролевые объекты/роли и «игра по роли»/функция/назначение в окружении 240
От целевой системы через метод работы к роли создателя, но не наоборот 247
Онтика изобретательства: конструктивы как аффордансы для ролей 253
Моделирование: функциональный/ролевой объект (роль) и его функция, материальный/конструктивный объект как аффорданс 256
Физическое мышление в системном мышлении 257
Функциональные и конструктивные рассмотрения одной и той же системы 262
Вопросы. Конструктивные и функциональные объекты 266
Производство систем как конструктивов в их потенциальных ролях 267
Мантра системного мышления 271
Задание по следованию «мантре системного мышления» 273
Второе и третье поколения системного подхода 273
Методология: агенты, роли, предметы интереса, интересы 280

Методологическое моделирование 286
Моделирование: аннотирование типами с использованием нотации :: 290
Вопросы. Конфликт интересов 290
Роли аналитиков и инженеров. Должности 291
Моделирование: аналитика и инженерия 296
Внешние и внутренние проектные роли 297
Моделирование: внешние проектные роли и успешность системы и её проекта 302
Культурная обусловленность проектных ролей 303
Игровая метафора 311
Мышление об агентах (людях и AI): прежде всего они исполнители ролей 320
Вопросы. Проектные роли 332
6. Интересы 338
Предметы интереса и интересы 338
Не склеивайте важную характеристику и предпочтение 344
Список методов описания из CPS PWG Cyber-Physical Systems (CPS) Framework 348
Разговаривайте не с людьми, разговаривайте с ролями 352
Моделирование: использование чеклиста для отслеживания ролей 355
Вопросы. Проектные роли и их предметы интереса 355
Позиция 358
Лидерство 363
Моделирование: роли коллег по организации 366
Внешние и внутренние роли 367
Организационные места, организации, оргзвенья 372
Начальники: это джокеры, ожидайте от них исполнения любых ролей 376
Моделирование: роли начальника 380
Звание и степень мастерства 380
Сколько всего проектных ролей 383
Ошибки в определении ролей 387
Моделирование: кто был на последнем совещании 391
Задания: исполнение ролей в жизни 391
7. Системные уровни 391
Не всё системы, что ими называют 391
Вопросы. Что считать системой 396
Системное разбиение 398
Эмерджентность и мета-системный переход 405
Вопросы. Системное разбиение и эмерджентность 411
Неустроенность и мета-системный переход 413

Системный подход против и редукционизма, и холизма. 424
Целевая система и коллективное системное мышление 428
Форматы описания системных уровней....432
Системные уровни в системной инженерии. Пример вычислительной техники. 435
Моделирование: системные уровни 438
4D системность: паттерны в пространстве-времени, ритмы....438
Моделирование ритмики 444
Задания по ритмике 445
Системные уровни выделяются вниманием. Пример методов телесной работы....445
Выделение системных уровней вниманием. Пример социальных танцев 457
Системные уровни. Продолжение примера социальных танцев: уровни выше танцора....462
Разделение труда между создателями. Наша система. Граф создателей. 471
Вопросы. Целевая система и соотношение систем....481
Рекурсивное применение системного мышления: рекурсивное управление вниманием....486
Задания по системным уровням 491
8. Графы создателей....491
Отношения создания....491
Моделирование: граф создателей 496
Концепция использования 496
Концепция системы и архитектура 505
Моделирование: концепция использования и концепция системы....507
Рекурсивное рассмотрение в системном мышлении 508
Примеры системной терминологии 515
Моделирование: краткое описание системы....521
Разделение труда и системные уровни 521
Вопросы. Концепция использования и концепция системы....525
Моделирование: концепция использования 528
Системы систем 529
Интеллектуальные агенты в системах 533
Системная этика 537
Государственное строительство и госпроекты 550
Будущее и проактивность деятельности....556
Сложность и меры сложности....563
Задания: системность мышления окружающих вас людей 566
Проблема наборов/collections как систем: чаще всего нет, но иногда — да. 566
9. Целевая система и её надсистема 574
Сначала найти целевую систему....574

Целевая система зависит от того, кто её ищет....583
Моделирование: целевые системы....592
Системы-продукты и сервисы/методы провайдеров/создателей....593
Пример парикмахерской как сервис-провайдера....597
Сервис-ориентация. Мир провайдеров....604
Примеры сервисов и их провайдеров....609
Моделирование: сервисы в компании....623
Ты — член команды....623
Моделирование: целевая система и ваша система....629
Признаки целевой системы....629
Частые ошибки в выявлении целевой системы....638
Моделирование: проверка на ошибки выявления целевой системы....647
Именование системы....648
Вопросы. Именование системы....652
Надсистема: её тоже нужно выявить!....655
Системный подход: для всех видов систем, не только для целевой....659
Вопросы. Системы создания....663
Задания: вы — сервис-провайдер....666
10. Как описывать системы....666
Фундаментальность/трансдисциплинарность системного мышления....666
Четыре основных вида описания системы на базе четырёх способов разбиения системы....669
Функциональный анализ и модульный синтез....676
Одна система, но множество описаний, множество имён: это нормально!....681
Альтернативные варианты основных видов разбиения системы на части....685
Несовпадение функционального и конструктивного разбиений системы....688
Моделирование: функциональные и конструктивные разбиения....694
Разработка концепции системы: функциональный анализ и модульный синтез....694
Вопросы. Функциональные, конструктивные, пространственные, стоимостные части - 1....701
Вопросы. Функциональные, конструктивные, пространственные, стоимостные части — 2....706
Моделирование: изобретение (поиск аффорданса)....709
Альфы и артефакты/«рабочие продукты»....709
Вопросы. Рабочие продукты....718
Яблоки из жизни, яблоки из задачи....721
Моделирование: классификация типов мета-модели типами мета-мета-модели....727
Альфы....727
Системное описание....739

Моделирование: описания, методы описания, предметы интереса описания, заинтересованные стороны 746
Системное описание (предметов) деятельности предприятия 746
Системное описание, описание, метод описания 750
Вопросы. Описания и важные характеристики 758
Задания: мета-моделирование 759
11. Системное моделирование 760
Описания и методы описания, модели и мета-модели 760
Моделирование: цепочки классификации (операционная модель, мета-модель, мета-мета-модель) 767
Системная модель, мета-модель, мета-мета-модель 768
Мульти-модель. Мега-модель. 775
Вопросы. Множественность описаний в системном моделировании 782
Понятие конфигурации системы 788
Функциональные описания: принципиальные схемы и сценарии 793
Конструктивные/продуктовые описания 801
Вопросы. Функциональные, конструктивные, пространственные, стоимостные части — 3 806
Платформы и платформенные/инструментальные стеки 811
Моделирование: уровни платформенного стека 821
Прикладная функциональность против универсальной конструкции 821
Вопросы. Наличие описания функциональности 829
Оргзвенья 830
Моделирование: назначение оргзвеньев на оргроли, выполняющие практики 834
Необходимость хорошей модульности 834
Борьба со сложностью в мышлении 836
Задания: системное моделирование 840
12. Вместо глоссария: онтология системного подхода третьего поколения 840
Онтология системы как набор описаний для понятийного наведения внимания 840
Первое поколение системного подхода: система в её окружении во время operations/эксплуатации 843
Второе поколение системного подхода: целевая система в её окружении, которую создают системы-создатели в их графе 844
Continuous everything в инженерии как техно-эволюция 845
Безмасштабные описания физических систем 848
Конструктивизм: системы выполняют операции как с физическими объектами, так и с абстрактными объектами 850
Третье поколение системного подхода: учёт времени эволюции 852
Что дальше 855

1. О мышлении

Понятия, используемые в системном мышлении

Человечество вырвалось из царства природы. Масса людей сегодня составляет 300 миллионов тонн. Это вдвое больше массы всех больших позвоночных, которые жили на Земле до появления человеческой цивилизации. Масса вещества, переработанного людьми, оценивается в 30 триллионов тонн, что больше 50 кг на каждый квадратный метр поверхности Земли[5]. И это следствие того, что люди в человечестве за последнюю пару тысяч лет усилили интеллект и накопили критическую массу объяснений, позволяющую эффективно «спасаться» от самых разных внешних напастей.

В первых трёх подразделах руководства мы приведём небольшую онтологию (объекты и отношения), которая поможет обсудить место системного мышления в общем мышлении. В руководстве будет не раз будет упомянуто, что опираться на выдачу точных определений можно только в математике. Но несколько подразделов будут похожи на «выдачу определений», чтение будет примерно таким, каким читают тексты словарей/глоссариев.

Мы не ожидаем, что вы вот прямо сейчас точно, как в математике, поймёте всё в этой небольшой онтологии — полное понимание придёт позднее, главное тут не застревать на каждом подразделе и упорно двигаться вперёд по освоению наших руководств (мы будем в них повторять и разъяснять каждое понятие, давать примеры и показывать ошибки их использования). Так, в нашем руководстве по системному мышлению будут повторяться и разъясняться многие понятия из руководства по рациональной работе. К концу освоения наставлений всех руководств программы рабочего развития у вас сложится современная картина мира, основанная на системном мышлении.

Чтобы понять место системного мышления среди других мышлений, нам потребуется разобраться, что же такое мышление — хотя повторим, мы не ожидаем, что по текстам первых трёх разделов вы поймёте, что это. Просто продолжайте осваивать руководства по программе рабочего развития, понимание придёт! А дальше можно будет углубить это понимание, проходя руководства программы исследовательского развития, прежде всего — руководство по интеллект-стеку фундаментальных методов мышления.

На мехмате МГУ у преподавателей есть поговорка: «Студент не понимает, не понимает, а потом привыкает». Вот и у нас — мы понимаем, что вы не компьютер, поэтому загрузить три страницы определений разных понятий в мозг осваивающего наше руководство так, чтобы потом свободно ими оперировать, сходу вы не сможете. Но зато вы получите представление о том, как вообще обсуждают интеллект, мышление,

---

[5] Zalasiewicz и др., «Scale and diversity of the physical technosphere», Zalasiewicz и др., http://journals.sagepub.com/doi/full/10.1177/2053019616677743

мастерство мышления, методы мышления.

Для нас тут главное — это показать место системного мышления в общем мышлении даже не человека — а всех агентов (и мы считаем, что из руководства по рациональной работе вы уже знаете, что такое «агент», но всё-таки потом мы ещё много раз повторим, что это). Заодно текст первого раздела покажет и пример использования самого системного мышления, а также рассуждения методологии как «учения о методах работы». Мы не ожидаем, что вы сразу поймёте здесь написанное, но к концу освоения руководств по рабочему развитию вам покажется, что это всё естественно и даже не требует дополнительных разъяснений.

Так что не застревайте на этой онтологии, не застревайте на первых трёх подразделах, но и не «проходите мимо». Последний раздел нашего руководства — это примерно такой же **синопсис** (краткое изложение онтологии без объяснений и примеров), он имеет название «Вместо глоссария: онтология системного подхода третьего поколения» с более-менее формальным изложением, он посвящён содержанию системного подхода третьего поколения. Но в первом разделе мы даём другую онтологию как набор понятий и отношений между ними для обсуждения места системного мышления среди самых разных других фундаментальных и прикладных мышлений.

Дальше текст существенно использует знание о типах объектов и их отношений из руководства по рациональной работе.

«Интеллектуальный агент»/«агент в узком смысле» может быть человеком, AI-агентом, организацией. Агент — это **создатель**::система, выполняющий универсальные физические преобразования других систем. Создатель — это система, которая выполняет всевозможные трансформации объектов физического мира. Понятие создателя построено как расширение понятия компьютера/вычислителя, который выполняет преобразования информации. Как и универсальный (тьюринг-полный) компьютер может выполнять все возможные вычисления при наличии достаточных ресурсов памяти и времени, так и универсальный создатель может выполнять все возможные трансформации физических объектов — при наличии необходимой энергии и времени. Мы можем считать, что в создателя входит компьютер/вычислитель, задающий программу преобразований. Человек — это универсальный создатель, вычислитель в котором реализован главным образом мозгом. Подробно обо всём этом будет говориться и в нашем руководстве по системному мышлению, и последующих руководствах программы рабочего развития инженеров-менеджеров.

Агент «состоит из»/«отношение композиции»:

- **Организм** («хардвер» агента — тело, «аппаратная часть»), сюда мы также относим **экзотело** с **экзокортексом**, и даже находящиеся в **собственности** агента **ресурсы,** например, какой-то **инструментарий/tools/instruments (набор инструментов/аппаратуры/оборудования),** увеличивающий аппаратные возможности агента. А если это коллективный агент, то речь идёт о всех телах вместе.
- **Личность** («софт» агента, «прошивка»/«программа») состоит из набора всех **мастерств** агента. Мастерство с задействованием организма выполняет **работы** каким-то специфическим для этого мастерства **методом/способом**. Личность состоит из:

- **Жизненное мастерство,** включающее
  - **Интеллект** состоит из
    - **Мастерство**::вычислитель, исполняющий **метод/практику/способ/культуру мышления** 1 (например, понятизация::метод) из набора методов сильного мышления (далее этот набор методов мы будем называть **«интеллект-стек»).** Метод/практика мышления 1, состоит из
      - подмастерство мышления с использованием понятий из **теории/дисциплины** этого метода мышления 1
      - Подмастерство задействования **инструментария** (набор инструментов/tools/instruments/аппаратуры/hardware, чаще всего в случае мастерства в составе интеллекта — это моделер) для этого метода/практики 1
    - Мастерство мышления по методу/практике 2 из методов мышления интеллект-стека 2 (например, мастерство мышления семантика)
    - …
  - **Мастерство ухода за собой,** состоит из
    - Подмастерство мышления с использованием понятий из знаний/теорий/дисциплин ухода за собой (как спать, как есть, как лечиться, как завести семью, …)
    - Подмастерство применения инструментария/tools/instruments/аппаратуры/hardware ухода за собой (как пользоваться зубной щёткой, …)
- Набор **прикладных мастерств** как вычислителей, умеющих достигать каких-то конкретных результатов в какой-то прикладной предметной области (например, авиастроении, менеджменте, прыжках на батуте)
  - Набор прикладных подмастерств мышления с использованием понятий из знаний/теорий/дисциплин прикладных предметных областей
  - Набор прикладных подмастерств применения прикладного инструментария/аппаратуры

Мышление может пониматься в двух значениях (мы будем использовать оба):

- В узком смысле — это вычисления интеллекта (то есть вычисления набора мастерств в интеллект-стеке фундаментальных методов мышления)
- В широком смысле — вычисления агента-создателя по любому виду его мастерства (включая прикладное мастерство).

Понятие метода/способа/практики работы в его отличии от собственно работы — это предмет отдельного руководства по методологии. Методология — это учение о

методе. Сам разговор о методе труден, ибо путаются разные виды отношений, а ещё метод описывается теорией/правилами/алгоритмами, которые чаще всего плохо выразимы пошаговоыми процедурами, и там те же проблемы понимания, что связаны с пониманием функционального программирования в алгоритмике и функционального анализа в математике.

Так, мышление — это метод/функция/практика/культура работы интеллекта. Но уместно спрашивать: «какой метод мышления ты применил»? Это не тавтология «какой метод метода ты применил», а попытка выразить на бытовом языке метонимию: имеется в виду работа мышления::метод и попытка выяснить, какой же вариант/вид этого мышления::метода был задействован.

Мы различаем работу (достижение какого-то результата) и метод/способ/практику работы. Мы различаем также метод/способ/практику работы, теорию/знания/дисциплину/алгоритм этой работы, инструментарий/аппаратура/оборудование для работы. Метод/практика работы — это сам способ, которым идёт работа (паттерны поведения), а не описание способа работы. А вот теория/объяснение/алгоритм — это как раз описание способа работы, обычно не слишком зависимое от используемого инструментария/аппаратуры/оборудования, но тесно с ним связанного. Повторимся: для всех этих тонкостей есть отдельное руководство по методологии, мы советуем освоить его следующим.

Итак, в узком значении — **мышление это вычисление::функция/процесс/метод работы интеллекта::«часть личности»::«набор мастерства по интеллект-стеку методов»::«набор вычислителей, работающих по интеллект-стеку фундаментальных методов/практик/способов мышления».** Набор этих методов обсуждается в руководстве по интеллект-стеку, мы очень кратко перечислим их ниже по тексту.

Особенность мышления::вычислений именно интеллекта в отличие от вычислений разных других видов мастерства — направленность на решение новых и новых классов проблем. Мышление в узком смысле (вычисления по интеллект-стеку методов) направлено на **познание** как создание новых и новых объяснений/теорий/знаний для новых и новых методов, позволяющих агентам затем и в одиночку, и коллективно с помощью самого разного инструментария этих новых методов изменять мир к лучшему, участвуя в бесконечной **техно-эволюции** (и даже вмешиваясь в **биологическую/дарвиновскую эволюцию**).

Системное мышление/systems thinking — это использование понятий системного подхода (система, подсистема и надсистема, эмерджентное свойство, системный уровень и т.д., мы будем их раскрывать на протяжении всего текста) в мышлении в широком смысле. Эти понятия помогают отмоделировать мир как состоящий из **важных объектов**: взаимодействующих вложенных друг в друга **систем**. Эти системы — **физические объекты**, устойчивые по отношению к воздействиям окружающей их непрерывно меняющейся и не всегда дружественной **среды**. Системы как-то удерживают **границу** между собой и средой, не смешиваются. На длинном **масштабе времени** эти системы ещё и **эволюционируют**, добиваясь всё большей устойчивости к неприятным для них **сюрпризам** со стороны окружающей среды.

Современный интеллект — человеко-машинный вычислитель, и часто он

коллективен, то есть мышление ведётся группой людей, поддержанных компьютерами. И даже без компьютеров человеческий биологический интеллект усиливается внешней надёжной памятью в виде заметок на бумаге или бумажных книг. Системное мышление подразумевает использование записей, мышление не всё происходит в биологическом мозге. **Не пишете/моделируете в компьютере или хотя бы на бумаге — не думаете системно.** Системно мыслить сейчас умеет не только человек, но и компьютер с программой искусственного интеллекта, если их этому научили. Системно мыслить (то есть использовать в мышлении и связанном с ним моделировании понятий системного подхода) может даже организация, хотя что такое «мышление» для организации и каков там мыслящий интеллект — это до сих пор вопрос споров.

Ещё один вопрос споров — только ли «вычисления» у интеллекта, ибо мастерство мышления по методам интеллект-стека включает и методы работы с инструментами/tools/instruments. Это может включать в мышление и какие-то физические процессы, которые физики могут сводить к информационным взаимодействиям (то есть с натяжкой можно сказать, что это «вычисления»), но в быту и даже в инженерии о них так уже не принято думать. Скажем, чтобы проверить мышление о том, что творится на обратной стороне Луны, можно создать и послать туда инструмент: космический корабль, который сделает замеры каких-то характеристик — и дальше мышление продолжится. Включать или не включать создание инструмента (микроскопа, ракеты, синхрофазотрона, многоэтажного дома, генномодифицированного помидора, термометра) в мышление, непонятно. Создатель — это обобщение понятия вычислителя/компьютера до агента, меняющего мир, так что в принципе, довольно легко отождествлять «чистое мышление как только вычисление» и мышление, которое включает задействование инструментов для изменения окружающего мира. Инструментом может быть и лопата, и компьютер, и робот, и даже другой человек, которого выучивают для выполнения каких-то работ по выбранным специально для этого методам.

Главное в том, что современный интеллект основывается на фундаментальных (из интеллект-стека) методах мышления, опирающихся на знания/теории/дисциплины, насквозь пронизанные понятиями системного подхода. При дополнении этих знаний инструментами-моделерами (в простейшем случае — экзокортекс в виде любого способа ведения записей, даже ручка-бумажка), интеллект справляется с рассуждениями о системах самого разного масштаба/физических размеров: от элементарных частиц до квазаров. В том числе интеллект справляется с ситуациями, когда одни создатели (например, люди) создают других создателей (например, роботов, или организации из людей) — и поскольку это всё системы, то для обсуждения таких ситуаций «создания и развития создателей создателями» тоже используются понятия системного подхода, это тоже будет системное мышление.

Системное мышление как сильное/универсальное/общее

Продолжим краткое изложение материала без пространных объяснений и примеров: мы просто даём это как справочный материал, чтобы вы могли оценить набор понятий, в которых удобно обсуждать само системное мышление в ряду других видов мышления. Конечно, потом и в нашем руководстве будут подробные разъяснения и примеры, а в следующих руководствах программы рабочего развития на эти темы будет исчерпывающая информация и ссылки на первоисточники. Не

застревайте на детальном понимании содержимого синопсиса в текущем разделе — но используйте для общего знакомства.

Мышление может быть **сильным/универсальным/общим** (мышление интеллекта, не ограничивается предметной областью) и **узким/неуниверсальным/прикладным** — мышление об объектах какой-то узкой предметной области (выращивание масленичных культур в целом и подсолнечника в частности, проведение оптических вычислений, обучение верховой езде). Эти понятия близки к тому, как определяют сам интеллект: сильный (сравнимый с человеческим, потенциально универсальный по отношению к предметной области) и прикладной — и тут споры о том, считать ли узкий интеллект именно интеллектом, или говорить о нём просто как о каком-то прикладном мастерстве. Нам это сейчас неважно, ибо интеллект мы определяем как набор специального вида мастерств — мастерство, осуществляющее ведение мышления по набору **фундаментальных/трансдисциплинарных** методов мышления, которые отвечают за универсальность интеллекта. Так что рассуждения по поводу прикладного мастерства (мастерства, которое может проводить мышление по прикладным теориям/дисциплинам/знаниям) проходят по той же линии, что и для фундаментального мыслительного мастерства, то есть интеллекта.

Набор методов мышления, которым следует сильный/общий интеллект, составил интеллект-стек. Stack/«стопка» (как в «стопке листов бумаги») означает, что каждый метод мышления и его знания/теорию/«научную/учебную дисциплину» и инструментарий (обычно это инструментарий моделирования — моделер) выше в стеке/стопке проще объяснять, используя методы мышления ниже её.

Вот интеллект-стек фундаментальных методов мышления сверху вниз, методы в нём названы по «научным/учебным дисциплинам»/«объяснительным теориям»/ «видам/областям знаний»: **инженерия, методология, риторика, этика, эстетика, познание/исследования, рациональность, логика, алгоритмика, онтология, теория понятий, физика, математика, семантика, собранность, понятизация.**

Методы мышления (и их знания/теории/дисциплины) в составе интеллект-стека **фундаментальны** (то есть «лежат в основании» прикладного мышления с использованием понятий дисциплин прикладных методов мышления). Фундаментальность методов мышления в составе интеллект-стека и дающих для них понятия теорий/дисциплин растёт к низу стека. Чем ниже в интеллект-стеке, тем более методы мышления и их дисциплины фундаментальны/«лежат в основании», понятия их более абстрактны по сравнению с понятиями прикладных дисциплин, для них труднее понять их влияние на какую-то ситуацию. Но в этом-то и секрет, что эти фундаментальные методы мышления полезны, когда встречаются **проблемы** — такие ситуации, когда непонятно, что делать: что и каким прикладным методом изменять в физическом мире, чтобы он изменился к лучшему, непонятно о чём думать, об объектах какой предметной области. Нужно самое абстрактное мышление об объектах, о методах изменения объектов, об организации коллективного мышления и коллективной работы.

Такие ситуации, когда надо задействовать интеллект и тем самым интеллект-стек фундаментальных методов мышления — часты. Проблем нет только у тех, кто

ничего не делает, но только в первый момент. Во второй момент бездельники получат от окружающей среды свои неприятные сюрпризы, окружающий мир никогда не спит.

Теории/дисциплины интеллект-стека, усиливающие интеллект (то есть делающие его как вычислитель более универсальным в предметах мышления и скоростным за счёт использования улучшенных алгоритмов, по которым идёт мышление), влияют на работу с прикладными методами, помогая разобраться с их понятиями.

Видимая часть мышления мастера как работы всех его мастерств (включая прикладные, интеллект, ухода за собой) — это прикладное мышление. Кто-то умеет строить дом, кто-то чинить синхрофазотрон. Результаты прикладного мышления распознаваемы, хорошо видимы! Но есть невидимая часть в мышлении: это проявление силы интеллекта как умение быстро и универсально решать ранее не встречавшиеся проблемы, **«переводить проблемы/problems в задачи/tasks»**.

Задача — это единица работы с известным методом/способом этой работы и известными потребными ресурсами, которая может быть поручена одному создателю, имеющему доступ к ресурсам и владеющему методом. Существенной частью сильного интеллекта тут является мастерство **стратегирования**: метод нахождения подходящей для преодоления проблемы **стратегии**::«метод работы» и после создания стратегии (выбранный метод преодоления проблемы) — мастерство планирования как составление плана использования ресурсов-конструктивов для реализации стратегии в виде набора задач, выполнение работ по которым реализует стратегию и тем самым преодолевает исходную проблему.

Системное мышление относится к мышлению интеллекта, оно проявляется в преодолении проблем, а не получении каких-то понятных результатов. С освоением системного мышления вы **станете умнее** (ваш **интеллект станет сильнее**), но это не значит, что вы научились делать какую-то конкретную работу: быть хорошим менеджером, певцом или инженером-теплотехником. Нет, вы просто будете быстрее преодолевать встречающиеся вам проблемы (в том числе проблемы, которые вам встретятся в менеджменте, пении, теплотехнической инженерии), но вот самим прикладным методам работы придётся учиться отдельно.

Каким-то общим методам работы (системная инженерия, инженерия личности как обучение, инженерия организации как системный менеджмент) вы научитесь в ходе освоения наших следующих руководств. При этом мы чётко будем различать: если ваш вопрос о том, как делать системные описания «вообще», это вопрос к руководству по системному мышлению. Если вопрос в том, как делать системные описания именно организации — вам надо обратиться к руководству по системному менеджменту, если надо системно описать мастерство — вам нужно руководство по инженерии личности.

Но если так, то зачем руководство по системному мышлению? В нём один раз рассказывается о самом типе «системное описание» и что там должно в этих описаниях обязательно присутствовать. Это руководство задаёт типы мета-мета-модели, а прикладные руководства (в том числе руководства инженерного цикла: программная инженерия, инженерия личности, инженерия организаций) — это руководства, задающие типы мета-моделей. В жизни же для решения проблем и проверки на отсутствие грубых ошибок надо типизировать понятия мета-модели (предметной области) типами мета-мета-модели, то есть типами интеллект-стека фундаментальных методов мышления. Если ваша модель организации вдруг

перестала быть системной, то вы об этом не узнаете из руководства по системному менеджменту, как создавать системную архитектуру программной системы узнаете из руководства по программной инженерии (хотя мы и касаемся этого вопроса и в нашем руководстве по системной инженерии, но ссылки там даём на литературу именно по программной инженерии). Не ждите ответа на вопросы по прикладным методам работы от руководств по фундаментальным методам мышления, в том числе от руководства по системному мышлению.

Руководства по фундаментальным методам мышления служат для другого. Так, после освоения нашего руководства по системному мышлению **полностью незнакомые вам ситуации станут как-то знакомыми, станут понятнее, будет проще в них действовать, это и будет проявление системного мышления.** Не знакомые с системным мышлением люди увидят паутину тесно взаимосвязанных непонятных и трудно выделяемых из окружения объектов, а вы как адепт системного мышления увидите набор малознакомых, но систем (система — это тип из мета-мета-модели мира, раскрываемой нашим руководством), связанных не «как-то», а более-менее понятно (понятными отношениями: прежде всего отношениями композиции и создания). И дальше вы разберётесь быстрее, потому что вроде как незнакомая ситуация будет для вас частично знакомой: вы будете знать типы важных объектов этой ситуации, и поэтому сможете внятно рассуждать.

Если у вас менеджерская проблема, то вы используете инженерию организации, а уж инженерия организации задействует системную инженерию, которая задействует методологию, а ещё как менеджер вы будете использовать методику и решать этические проблемы, мыслить рационально и логично. Редко кто отмечает, что деятельность агента в роли менеджера заключается в инженерии организаций. Не подчёркивают, что личность этого агента мыслит логично и руководствуется в своих решениях самыми свежими достижениями этики. Но это не значит, что в жизни всё так и происходит. И у какого агента-менеджера интеллект сильней, тот и будет лучшим менеджером: он просто дообучится нужным прикладным методам, когда будет встречаться с проблемами. Системное мышление поможет это сделать. Агент — это обычно был взрослый человек (не трёхлетний ребёнок), но сегодня таким агентом может быть AI, а может быть и целая организация из людей и искусственных интеллектов.

Тем самым интеллект-стек фундаментальных методов мышления (физика, математика, логика, методология и т.д.) используется в самых разных прикладных областях, от музыкальной композиции до робототехники. Нас меньше будут волновать полные (включающие не только теории/знания/дисциплины, но и инструментарий для поддержки мышления) методы мышления из интеллект-стека. Поэтому часто говорят не об интеллект-стеке фундаментальных методов мышления, а только о дисциплинах/теориях/знаниях методов в составе интеллект-стека. К тому же методы чаще всего называют по имени их теорий/знаний/«учебных/научных дисциплин», а не инструментов/аппаратуры (хотя бывают и исключения, типа «микроскопии»[6]), и это не случайно. Инструментарий поддержки теории/дисциплины какого-то метода можно легко заменить (например, один моделер заменить на совершенно другой), а поддерживаемая теория/дисциплина останется.

---

[6] https://ru.wikipedia.org/wiki/Микроскопия

Определяемое фундаментальными методами мышления в составе интеллект-стека системное мышление универсально, фундаментально, безмасштабно: оно задействуется в мышлении о самых разных прикладных предметных областях, оно находится на высоком уровне абстракции (мета-мета-модель мира, об этом будет много подробностей).

Мастерство в мышлении по интеллект-стеку фундаментальных методов нужно для решения проблем, с которыми обладатель интеллекта ещё не встречался. Прикладные методы работы помогают решать задачи, с которыми обладатель интеллекта уже встречался. Но как только ситуация в жизни как-то отличается от ситуации, описанной в прикладном методе или его объяснениях/дисциплине (будь это сантехника, приборостроение или выращивание генномодифицированной рыбы), потребуется задействовать интеллект — чтобы разобраться с тем, что там за отличие и как решать возникшую проблему.

Интеллект, усиленный фундаментальными знаниями/теориями/дисциплинами **универсален**, ибо помогает справляться с проблемами в самых разных прикладных предметных областях, от животноводства до ракетостроения. Поэтому объяснения/дисциплины интеллект-стека называют часто **трансдисциплинами**, это «дисциплины для рассуждения в ходе задействования прикладных методов» (trans-/транс- — это «находящиеся по ту сторону» от прикладных дисциплин, вернее, дисциплин прикладных методов). Обзор современного содержания трансдисциплин будет в руководстве по интеллект-стеку.

Эти фундаментальные знания/теории/дисциплины **безмасштабны,** в отличие от привязанных к определённым масштабам в пространстве и времени знаний прикладных дисциплин. Термин «безмасштабный/scaleless/scale-free» пришёл из физики и используется по отношению к дисциплинам/знаниям/объяснительным теориям, в которых отсутствует зависимость их применимости от размера обсуждаемых объектов. Фундаментальные дисциплины позволяют рассуждать об элементарных частицах, людях, авиалайнерах, горах, квазарах и галактиках. Системное мышление в составе этих фундаментальных/трансдисциплин не только универсально в части предметных областей, занимающихся разными свойствами предметов физического мира, но и безмасштабно в части рассуждений о самых разных ситуациях в физическом мире, рассматриваемых на разных масштабах размеров — от микромира субатомных частиц до космического мира галактических кластеров, а также самых разных масштабах времени — от фемтосекундных световых импульсов в лазерах до миллиардов лет, которые шла биологическая эволюция.

Прикладные дисциплины/знания/объяснительные теории используют понятия (типа мета-мета-модели) трансдисциплин интеллект-стека для того, чтобы обсуждать методы работы с системами как физическими объектами, разбитыми на уровни какого-то определённого масштаба/размера. Эти уровни «по масштабу/размеру» называют **системными уровнями**. Из маленьких молекул составляются более крупные клетки, из мелких клеток большие организмы, из множества разных организмов огромные популяции — это и есть деление систем живой природы на системные уровни. Из винтиков и металлических деталей собираются авиамоторы, из авиамоторов, фюзеляжей и крыльев — самолёты, из самолётов и аэропортов — авиатранспорт. Это системные уровни в инженерии.

Разные прикладные методы работы занимаются закономерностями поведения

систем из разных системных уровней. Молекулярная биология — белками, моторостроение — моторами, авиастроение — самолётами. Этими системными уровнями занимаются прикладные **методы/практики/стили/культуры** инженерии. **Инженерия** меняет физический мир к лучшему, прикладывая мышление о каких-то видах систем к изменению физического мира с целью создания и развития **целевых систем** — моторов, самолётов, коров, предприятий и даже обществ. Скажем, классическая системная инженерия — это инженерия киберфизических систем (роботов, авиалайнеров), медицина — это инженерия (сейчас главным образом ремонт) человеческих существ, обучение — это инженерия (главным образом развитие/модернизация) личности, менеджмент — это инженерия организаций.

Сильный интеллект реализует идеи «спасения» **агентов**, которые своей инженерией меняют мир, чтобы устранить проблемы сейчас и предотвратить их в будущем. Интеллект есть у чего угодно (агенты в широком смысле), только разной силы. Но вот интеллектуальные агенты — это пока только образованные люди и к ним приближаются современные системы искусственного интеллекта. Люди ещё и объединяются в организации вместе с AI-агентами. Человечество переработало по 50 кг материала на каждый квадратный метр поверхности земли, чтобы не умирать от голода, холода, болезней, не быть съеденными дикими животными и даже не быть уничтоженными соплеменниками. Не умереть от голода и холода — это уже больше задачи, это не проблемы. С болезнями — смотря какая болезнь, лечение некоторых болезней до сих пор проблема, некоторых — уже задача, но биологическое бессмертие — пока проблема. Уничтожение людей соплеменниками — пока тоже проблема, хотя ситуация уже много лучше, чем пару тысяч лет назад.

Смысл жизни — «спастись», в эволюционном смысле этого слова. **Жить сейчас и в будущем, для этого запастись знаниями самых разных методов работы и затем проводить работы, используя знания и инструментарий методов работы в инженерных проектах по созданию самых разных систем.**

Системное мышление — это мышление по той части знаний/теорий /дисциплин методов интеллект-стека, которые ответственны за мышление о проблемах, перевод проблем в задачи и в этом мышлении о проблемах и методах их решения используют понятия системного подхода.

Знания для решения задач обычно есть, знаний для решения проблем — нет. Интеллект как раз даёт мышление с использованием «знаний о знании», «знание трансдисциплин о прикладных дисциплинах». Помним, что часто о методах/способах/культурах/стилях работы говорят в отрыве от их инструментария — как об их прикладных знаниях/дисциплинах. Системный подход — это часть таких знаний, то есть часть «знаний о знаниях». Системный подход как объяснения на основе набора понятий мета-мета-модели содержится в дисциплинах/теориях/знаниях методов мышления интеллект-стека. Но мы ещё и понимаем, что голыми руками и голыми мозгами не работают, задействуют инструменты. Поэтому системное мышление требует не просто мастерства в части использования мозга как аппаратуры мышления, но и мастерства использования инструментария — моделеров, работающих с самыми разными описаниями систем.

Системный подход оказался настолько важен в инженерии, что в её наиболее

общем варианте инженерия стала называться **системная инженерия**. Впрочем, и о современной методологии можно смело говорить «системная методология», ибо в её основе тоже существенно использована идея системного подхода, да и физика использует понятие «системы» в качестве одного из основных, и сам термин пошёл оттуда. Другое дело, что физику «системной» не называют, там системы привычны.

А вот «системная биология» — это инженерия искусственной жизни, слово «системная» там по большому счёту не означает системного подхода, а означает просто «создаваемые живые объекты», ибо **в быту словом «система» часто называют что угодно, что достаточно сложно, и при этом меньше всего имеют в виду термин из системного мышления. «Система кошка зашла за систему угол» — это бытовой язык, а не язык понятий системного мышления.** Тип «система» можно запросто убрать, он тут ничего не даёт.

Итак:

- **Интеллект — это мастерство в методах превращения проблем, которые непонятно как преодолевать, в задачи, которые надо просто взять и сделать.**
- **Мышление — это функция интеллекта, его поведение, то, что делает интеллект. По сути мышление — это вычисление интеллекта, так что вычисление может проводиться по каким-то методам мышления. Методы мышления, которые ведут к усилению интеллекта, составляют интеллект-стек.**
- **Системное мышление — это некоторое подмножество мышления, занимающееся появлением новых свойств целого объекта-системы (эмерджентность) в результате взаимодействий частей этих целых объектов. Системное мышление основывается на понятиях системного подхода.**

Эмерджентность и разделение труда

В последнем абзаце предыдущего подраздела было введено понятие **эмерджентности** — появление/emergence на каждом системном уровне у систем свойств, которых не было у систем на предыдущем системном уровне. Часы могут показывать время, а вот шестерёнки, пружинки, корпус механических часов — нет, транзисторы в электронных часах — нет.

Новые/эмерджентные свойства заставляют использовать для объяснений на каждом масштабе/размере/системном уровне какую-то свою прикладную дисциплину/теорию. Системное мышление позволяет разобраться во всём хитросплетении нарезки мира на системы из разных системных уровней («нарезку» не надо понимать буквально, это «нарезка вниманием», выделение систем как фигур из фона). Тем самым в основе работы интеллекта над созданием и использованием разных прикладных методов работы с их дисциплинами/теориями/знаниями, обслуживающих понимание происходящего на разных системных уровнях, лежит именно системное мышление. Методы/способы/практики/стили системного мышления удивительно эффективно помогают разобраться со сложностью окружающего мира.

Разделение труда — это когда разные интеллектуальные агенты (люди и AI) специализируются на методах работы с каким-то отдельным масштабом систем. Скажем, агент в роли терапевта занимается в целом человеческим организмом,

состоящим из органов. А вот агент в роли пульмонолога — специализируется только на лечении лёгких. В основе разделения труда лежит как раз системное мышление: системы разных масштабов разделяются по системным уровням, а затем изучаются разными науками и изменяются разными видами инженерии.

Один агент не в состоянии глубоко освоить изменение всех систем мира на всех системных уровнях, ему банально не хватит вычислительных ресурсов для длительного обучения. Если учиться до профессионального уровня какому-то методу работы год (это очень консервативная оценка! Попробуйте за год научиться профессионально играть на фортепиано, или быть профессиональным архитектором организации!), то можно научиться за тридцать лет всего 30 методам работы — и когда дойдёшь до тридцатого, выяснится, что первый освоенный метод работы безнадёжно устарел. Но самих методов работы тысячи, десятки тысяч! Никакой жизни одного агента не хватит, чтобы на профессиональном уровне освоить их все.

Поэтому идёт разделение труда, разные агенты специализируются на разных методах создания и развития разных систем. **Чтобы разобраться, каким мастерством в каких методах работы должны владеть агенты в вашем конкретном проекте, вам нужно сначала нарезать мир на системы, выбрать из них важные для вашего проекта, чтобы потом выбрать метод работы с этими важными системами, и уже после этого выбрать агента, который сможет работать по выбранному вами методу. И вот тут пригодится системное мышление, тут должен работать универсальный и безмасштабный интеллект, ибо надо будет мыслить о системах, о методах работы с системами, об агентах, которые будут работать с системами (то есть переводить системы из одних состояний в другие).**

Перед тем, как заняться изучением системного мышления, нужно понять **общие критерии сильного мышления** как мышления сильного интеллекта. Эти критерии применимы и к системному мышлению. Также нужно ответить на вопрос: в чём разница между **системностью и систематичностью**, ибо их часто путают.

В мире существует много вариантов системного мышления, базирующихся на разных вариантах системного подхода, поэтому нужно будет понять, какой вариант прописан в нашем руководстве, и почему именно он.

Дальше в этом разделе будет некоторое количество разъяснений по тексту руководства:

- как относиться к терминологии (слова-термины важны, и не важны!),
- как выбрать степень строгости/формальности системного мышления,
- как убедиться, что системное мышление помогает, а не мешает свободе творчества (в курсе же приведены шаблоны эффективного мышления, которые отлично работают, но бывает ли «творчество по шаблону»? Да, конечно, творчество бывает и даже предпочтительно по шаблону! Без шаблона много легче породить банальность, а не что-то новое!).
- можно ли вообще научиться мышлению, или нужно просто родиться умным? Конечно, мышлению можно научиться, хотя родиться надо бы человеком (или нечеловеческой продвинутой AI-системой), а не кошкой.
- какие стадии освоения мышления?

И только после рассказа обо всём этом в следующем разделе мы начнём изучать

основные понятия системного подхода и связанные с ними мыслительные приёмы самых разных фундаментальных методов в составе интеллект-стека.

Сразу скажем, что наше руководство по системному мышлению примерно эквивалентно в плане освоения его материала руководству езды на велосипеде: чтение руководства как книги многое вам расскажет про системное мышление, но не факт, что после прочтения нашего руководства как книги вы станете системным мыслителем. Нужна практика, руководство предназначено для использования его в работе! В ходе освоения нашего руководства нужно использовать раскрытые в нём приёмы мышления в реальной жизни. Осваивать работу по руководству системного мышления трудней, чем проходить курс системного мышления по учебнику и учиться решать задачи из курса «для экзамена». Задачу из задачника в учебном курсе, учебный кейс ты просто решаешь, а в жизни задачу надо сначала поставить, рабочий кейс увидеть, и только потом решить — и ставить задачи труднее, чем их решать. Руководство по системному мышлению заставляет ставить задачи перед тем, как их решать, и делать это не в учебном, а в рабочем контексте.

Пререквизит к освоению системного мышления — освоение рациональной работы по соответствующему руководству. Вы после него:

- стали рационально работать, у вас появилось время на освоение системного мышления, вы смогли перестроить расписание,
- научились не путать слова/термины и понятия,
- научились отличать физические объекты от их описаний,
- не забываете записывать результаты моделирования,
- не требуете определений для понятий,
- находите объекты заданного типа в окружающем мире,
- удерживаете внимание на найденных объектах, когда вокруг обычная жизненная суета и мир вокруг мельтешит, и вам некогда сосредоточиться,
- находите из предложенной вам онтологии тип для объектов окружающего мира, если их тип неизвестен.

Но если в руководстве по рациональной работе речь шла о самых разных типах самых разных объектов, то в нашем руководстве по системному мышлению мы расскажем, что основные объекты, на которых надо удерживать внимание в ходе рабочих проектов — это объекты типа «система».

Следующим руководством для освоения в программе рабочего развития будет руководство по методологии, где понятия системного подхода будут использованы для обсуждения методов/деятельности/практики/инженерии/культуры/стиля/технологии создания и развития одними системами (создателями) других систем (создаваемых: целевой или вашей). Это живые системы «вырастают сами», но если речь идёт о создании чего-то интеллектуальными агентами (людьми с AI и их организациями), то нужно обсудить, как описывать паттерны/шаблоны/методы/практики/культуры работ по созданию систем. Основатель фирмы создаёт фирму, которая создаёт станок, который создаёт деталь. Чтобы организовать коллективную работу для всего этого сложного процесса, методологи должны создать руководство по предполагаемым в этих работах методам, а затем в участниках этих рабочих процессов (необязательно по этим руководствам будут идти люди, возможно это будет делать и достаточно вменяемый AI) надо будет создать прикладное мастерство, которое и будет затем всё это делать. Как описать эту коллективную инженерию, как об этом думать? Вот

этим и занимается методология.

Дальше системное мышление и системная методология будут использованы в руководстве по (безмасштабной, для всех видов систем, а также эволюционной/непрерывной) системной инженерии. И дальше будут руководства инженерии личности (как развивать личность, как научить личность новому мастерству) и руководство по системному менеджменту как руководство по инженерии организации и сообщества. Это всё руководства программы рабочего развития.

Конечно, системное мышление используется агентами и для ускорения понимания прикладных методов работы. Это часть поведения сильного интеллекта. Поэтому мы сначала учим работать по руководствам рациональной работы, системного мышления и методологии, и только затем переходим к освоению руководств по прикладным методам работы с «железными» и программными системами (софтом), личностями, сообществами, организациями.

Освоение системного мышления — это очередной шаг усиления интеллекта, то есть очередной шаг в изучении интеллект-стека фундаментальных/трансдисциплинарных методов мышления.

Повторим: не застревайте на содержании этих первых двух разделов, впрочем, и последующих разделов, и даже содержании всего нашего руководства. Осваивайте руководства, потратив разумное время на каждый подраздел, прямо в ходе работы по своему рабочему проекту, обязательно выполняя задания так, как вы выполняете рабочие задания на стажировке. Понимание обычно приходит не сразу после чтения текста подраздела, а чуть позже, иногда при чтении следующих подразделов, иногда при чтении следующих разделов, иногда при освоении следующих руководств, но чаще всего в ходе выполнения заданий и по мере приобретения опыта работы по руководствам.

В следующем разделе мы рассмотрим особенности подачи материала в наших руководствах. Главная мысль там: наше руководство не энциклопедия-википедия, не справочник/handbook («маленькая книжка, удобно помещающаяся в руку») со структурированной выжимкой по системному мышлению, не инструкция/ьфтгфд с пошаговым её исполнением. Оно устроено не как учебник математики с парой определений по каждой теме и десятком учебных задач для этой пары определений, ибо руководство для работы в полной неопределённости жизни — не учебник по строго определённым ментальным/математическим/идеальным объектам. Мы понимаем, что это руководство не для программирования классического логического компьютера в головах наших инженеров-менеджеров, но для направления на системное мышление нейронной сеть в мозгу инженеров-менеджеров (а то и не в мозгу, а в датацентре, ибо нашими текстами могут руководствоваться и AI-агенты).

Особенности подачи учебного материала

Вменяемость (характеристика интеллекта, показывающая, насколько агент способен после принятия рациональных аргументов поменять своё поведение) у людей лучше, чем у кошек. У кошек лучше, чем у муравьёв. Людям можно один раз объяснить, а кошку только надрессировать повторениями, чтобы добиться воспроизведения желаемого поведения. Муравьёв нельзя даже надрессировать.

Но как бы ни были люди вменяемы, жизнь показывает, что нельзя только

объяснениями научить ездить на велосипеде и автомобиле, нельзя только объяснениями научить делать что-нибудь сложное. Например, нельзя только объяснениями научить математике. Нужно решать задачи, делать упражнения. Просто прочесть учебник математики — не поможет. Нужны повторения действий с материалом учебника, удержание мозга в размышлениях на темы учебника.

Для освоения какого-то культурного (по лучшим образцам, в лучшем имеющемся стиле, а не абы как случающегося) мышления нужны упражнения, нужно выполнение большого числа повторений мыслительных операций. Нейронная сетка учится на большом числе повторений.

Это относится и к мыслительным операциям наведения внимания на какие-то важные объекты. В нашем руководстве этими важными объектами будут самые разные системы, которые надо в проекте распознавать, выявлять их границы, уточнять их тип, находить отношения между ними, уметь описывать их и их поведение, распознавать их состояния.

Поскольку надо удерживать внимание инженера-менеджера, стажирующегося по нашему руководству хотя бы минимально необходимое для хоть сколько-нибудь надёжного обучения нейросети его мозга время, то наше руководство/путеводитель/gudie представляют собой не хорошо структурированный справочник/reference/handbook, то есть результат работы методолога, выявляющего и описывающего формально и кратко метод системного мышления. Нет, эти материалы руководства (как и других наших руководств) в том числе отражают работу методиста как специалиста по обучению агентов с нейросетью, прежде всего людей. У людей учится «мокрая» нейронная сетка, для её обучения нужны многократные повторения предъявляемых для обучения понятий, используемых в разных контекстах. Мозг запоминает в том числе контексты, и по сопричастности к ним вспоминает и требуемые действия. Не понять один раз повторённое много раз одно и то же, а понять через повторение в разных контекстах — наше руководство, как путеводитель, ведёт к освоению системного мышления, учитывающего особенности обучения нейросетей (не только человеческих! Современные AI на базе LLM обучаются примерно так же, и там даются ещё и примеры рассуждений, в том числе и рассуждений с ошибками, тысячи и тысячи повторений).

Вот ровно это обилие повторений вы и видите в нашем руководстве. Материал руководства при этом, конечно, немного распухает (но всё-таки меньше, чем объём «Войны и мира», которую в школе проходят за одну четверть по литературе, не прерываясь при этом в изучении математики, физики, химии, русского языка и прочих предметов школьной программы, это точно не занимает много времени — хотя там и на вопросы по тексту отвечают, и ещё пишут домашние сочинения). Зато текст руководства позволяет освоить системное мышление, это проверено нашими инженерами-менеджерами. А вот однократное предъявление «точного определения каждого понятия, как в учебнике» не работает как в части собственно выдачи определения (про бесполезность, и даже вред определений в нашем руководстве есть специальный подраздел, но вы должны это уже понимать и по итогам освоения руководства по рациональной работе), так и в части освоения мышления по нашему руководству.

С одного предъявления какой-то мысли в коротком фрагменте текста научения нейросетей не происходит, в том числе научения такого, после которого можно

было бы выполнять какие-то рабочие задачи — просто не помнится материал, который нужен для этих задач, а наставник-человек или даже наставник-AI, который ткнёт изучающего системное мышление по нашему руководству стажёра в невспомненный короткий фрагмент текста, может отсутствовать.

При освоении руководства по системному мышлению, как и при освоении других методов работы по их руководствам, главное — это выполнять задания на текстовое (писать посты) и табличное (заполнять таблички с предписанными колонками) моделирование. Это мышление письмом/моделированием, удержание мозга на системных размышлениях о ситуации проекта. Не пишешь — не думаешь, не общаешься — не в клубе, не в культуре[7].

По факту стажёр, осваивающий системное мышление работает основное время не с руководством и к нему приложенной «рабочей тетрадью», а просто работает с моделером для рабочих проектов (пишет посты по итогам каких-то действий и размышлений, а также заполняет таблички в заданиях). Руководство тут — просто «контекстный хелп» для этого моделера, само системное мышление идёт в работе с моделером, а не в работе с руководством. Руководство только рассказывает о том, о чём надо не забыть подумать и какие ошибки в мышлении не стоило бы допускать. На каждый час времени стажировки по системному мышлению работа в моделере может занимать 50 минут, а чтение текста руководства — 10 минут. Так что чтение текста руководства — это может быть всего одна шестая от времени стажировки. А дальше? Всё то же самое, уже не стажироваться, а работать, только уже без регулярного обращения к тексту руководства! Всё то же самое, как и для любого другого рабочего руководства, стандарта деятельности, регламента работы и тому подобных способов предписания какого-то метода работы.

Дальше давайте посмотрим на этот расклад времени «один к пяти», ибо полезность разных способов освоения системного мышления с использованием нашего руководства всегда есть куда уменьшать. Так, **если вы пролистали текст, а не медленно его прочитали и затем выполняли задания по мышлению письмом/моделированием (это обычно чуть больше, чем втрое быстрее, чем «внимательно прочли», вместо десяти минут будет примерно три минуты), то вы потратите три минуты из шестидесяти, мозг ваш будет работать вместо часа над какой-то темой только три минуты — это будет для всего руководства освоение примерно 5% материала. Если внимательно прочли, не выполняя заданий, то освоили системное мышление на одну шестую часть от предполагаемого руководством, это примерно 17% материала. Надеяться на освоение системного мышления без выполнения заданий — нельзя, мозг учится только когда некоторое время удерживает внимание на какой-то теме, причём ещё и привязывает её к разным обстоятельствам реальной жизни, разным контекстам.**

Для примера сразу повторим только что сказанное про необходимость повторений в разных контекстах, чуть перефразируя. Основное в руководстве в части методики его освоения:

- Удержать «мокрую нейронную сетку» инженера-менеджера (то есть вас) достаточное время на работе с понятиями системного мышления, чтобы дать необходимое для обучения нейронной сетки время. Работу учителя можно автоматизировать, работу ученика — нет, ибо научение чему-то нейронной

[7] https://ailev.livejournal.com/1756090.html

сети требует большого числа повторений операций с каким-то набором понятий.

- Основные положения руководства должны каким-то образом повторяться в самых разных контекстах, чтобы быть эффективно воспринятыми. Это означает, что **бесполезно пятикратное или даже десятикратное перечитывание какого-то кусочка текста равно как бесполезно пятикратное моделирование одного и того же объекта одним и тем же способом. Надо идеи какого-то кусочка текста пять раз прочесть в других кусочках текста, равно как заданным способом надо отмоделировать пять разных объектов, чтобы повторения были полезны.**
- Чтобы преодолеть естественное забывание, надо повторить обучение несколько раз, при этом не подряд, а с некоторыми интервалами, учитывающими кривую забывания[8] (spaced repetition, интервальное повторение)[9].

В изложении материалов руководств программы рабочего развития инженеров-менеджеров (в том числе материала руководства по системному мышлению) используются методические приёмы, которые могут вызвать удивление у не знакомого с ними инженера-менеджера, привыкшего к справочникам и научной литературе. Нет, материалы нашего руководства устроены так, чтобы облегчить их освоение, в них[10]:

- Используются принципы «разнесения» (spacing) и «перемешки» (interleaving). В тексте идёт намеренное постоянное возвращение к каким-то темам через разные промежутки времени (spacing), одновременное обсуждение винегрета из трёх-четырёх тем (interleaving) вместо строго последовательного их изложения. Последовательного изложения тем, группировки нескольких примеров на одну тему (в образовании это blocking), как это обычно делается в справочной литературе — этого нет намеренно! Руководство/guide — это наставление, путеводитель, это не справочник! Если нужна справка, можно спросить AI, который даст справку по материалу руководства. Инженеры-менеджеры интуитивно считают, что упорядоченное сжатое изложение материала по одной теме (blocking) и легче в восприятии, и лучше для последующего вспоминания. Но это заблуждение, неверная интуиция. Жизнь контринтуитивна в этом вопросе: **эксперименты показывают, что перемешка (interleaving) и перерывы в подаче темы (spacing) замедляют освоение материала, ибо более трудны в восприятии, но зато существенно улучшают результаты обучения[11]**.
- некоторые фрагменты текста повторяются в разных местах руководства, иногда через пару абзацев, а иногда в соседних подразделах, а иногда и в других руководствах. И это не один раз — некоторые мысли повторяются десятки раз! Повторения тоже намеренны, текст учитывает необходимость

---

[8] https://ru.wikipedia.org/wiki/Кривая_забывания

[9] https://ru.wikipedia.org/wiki/Интервальные_повторения

[10] Есть отдельное руководство по обучению как инженерии мастерства, понимаемого как система, создаваемая и развиваемая в ходе инженерного процесса — руководство по инженерии личности.

[11] статья, в которой этот эффект описан в экспериментах для одного из классов задач подробно, но приводятся ссылки на аналогичные эффекты и для других классов задач: M. S. Birnbaum, E. L. Bjork, R. A. Bjork, Department of Psychology, University of California, Los Angeles, "Why interleaving enhances inductive learning: The roles of discrimination and retrieval", https://yadi.sk/i/UPRTP0DxRpw8Vg.

повторения для обучения нейронной сети инженера-менеджера (как человека, так и AI), ибо человеческий мозг, равно как AI-система на базе LLM на нейронной сети — не классическая компьютерная память, запоминающая всё с первого предъявления. Можно считать себя гением, но в лучшем случае это будет гениальное запоминание, но не освоение в части применения наставлений руководства. Наше руководство обычно осваивается за месяц-полтора без отрыва от других деятельностей, примерно так же, как школьники «проходят мимо» роман Л.Н.Толстого «Войну и мир», получая общее представление о его содержании. Повторим (мы уже об этом говорили, буквально страницу назад, вы помните? Кто прошёл по ссылке и уточнил, что это такое?), что к концу изучения последних разделов содержимое первых разделов успевает забыться, это описывается **кривыми забывания**[12]. Мы понимаем, что не все решатся систематически пройти руководство в режиме стажировки с наставником или даже самостажировки «для повторения», да ещё и в середине первого чтения, так что мы просто встроили немного повторения в однократное прочтение. Но мы уверены, что всё равно **будет крайне полезно простажироваться на работу по нашему руководству второй раз, причём даже простажироваться не столько на работу по этому одному руководству, сколько по всем руководствам программы рабочего развития мастерской инженеров-менеджеров** (в настоящий момент это руководства по рациональной работе, системному мышлению, методологии, системной инженерии, инженерии личности, системному менеджменту, и мы крайне рекомендуем ещё ознакомиться с руководством по интеллект-стеку из программы исследовательского развития). Почему надо освоить все эти руководства, причём в заданной последовательности? Потому как каждое следующее руководство в цепочке будет добавлять понимание материала предыдущих руководств. **Если будет трудность понимания материала какого-то подраздела, бесполезно повторять многократно его чтение (конечно, если вы его читали медленно, внимательно, а не «по диагонали»), просто идите дальше: наверняка через несколько страниц будет разъяснение, а если дошли до конца руководства, то разъяснение может случиться в следующем руководстве программы рабочего развития.**

- Дело не в том, что в текст нашего руководства встроено мало повторений. Дело в том, что понимание при повторном прохождении всей цепочки руководств программы рабочего развития инженеров-менеджеров будет совсем другим! **Опыт показывает, что второй раз текст руководства воспринимается как «вижу в первый раз»: вычитывается и понимается из него совсем другое.** Впрочем, этим наше руководство не отличается от любых других не самых простых текстов, от любых стандартов, регламентов, наставлений. Каждый сложный текст передаёт сложно связанный граф понятий. **При последовательном описании графа неминуемы «ссылки вперёд», на ещё не слишком понятые концепты. Тут есть только вариант двойного прохода: первый раз вы знакомитесь с понятиями, а второй раз понимаете все связи между уже знакомыми понятиями.** Описание связей между плохо понятыми концептами бесполезно: будет проигнорировано и забудется.

[12] https://ru.wikipedia.org/wiki/Кривая_забывания

- Терминология для основных понятий системного мышления намеренно не задана в руководстве однозначно: в явном виде везде используются **синонимические ряды/«паровозики»,** где синонимы все «с нюансами». Нельзя будет быстро пробежаться глазами по фразе, чтобы понять её! Глаз будет спотыкаться о длинный синонимический ряд. Метод/практика/practice/деятельность/стиль/культура/«вид труда» — такое будет встречаться в тексте повсеместно. Требуется каждый раз осознанно понимать, не какое слово написано, а какое понятие имеется в виду под этим рядом. Это тоже сделано в нашем руководстве намеренно, где-то к середине текста вы осознаете полезность такого подхода — хотя в начале изучения материала это примерно так же раздражает, как и повторения. Вы начнёте с любопытством присматриваться, что там помянуто в этих синонимических рядах. **В жизни вам не будут встречаться слова из наставлений руководства (не встречаете же вы в жизни «физические тела», хотя в учебнике физики речь именно о них), а в руководстве нет слов из жизни (в учебнике физики не описан полёт пустой бутылки в мусорную корзину, а описан полёт физического тела неведомо куда).** Наше руководство написано в языке мета-мета-модели, а жизнь — в языке мета-модели предметной области с очень редким употреблением типов мета-мета-модели, и эти типы обычно встречаются в другой терминологии, нежели вы ожидаете. Где вы ожидаете увидеть «метод работы» будет «культура», где вы ожидаете увидеть «культура» может быть «стиль». Наше руководство готовит к этой ситуации: **присваивание типов из руководства объектам из жизни делается не на основе схожести названий (слов), а на основе схожести понятий (ментальных моделей, стоящих за использованными словами-терминами)**. Чтение текста руководства становится более медленным и осознанным, но это хорошо: нейронной сетке вашего мозга от этого сплошная польза! Использование «косой черты»/slash для синонимических рядов довольно распространено в инженерии и англоязычных текстах, это только в литературных русскоязычных текстах она не приветствуется, мы об этом знаем[13], но в нашем случае игнорируем.

Обучение оказывается тем редким случаем, когда «разнесение», «распыление», «повторение» и нефиксированная терминология полезны для результата. Конечно, они увеличивают время на чтение текста, но уменьшают время на итоговое освоение мышления до степени применимости в работе. Мы всё это «неудобство для чтения» считаем полезным: оно увеличивает время, которое мозг работает с понятиями руководства, оно полностью соответствует рекомендациям наук обучения (learning sciences)[14]. Это полезное увеличение времени, это цель! Нам надо удержать вашу нейронную сетку в работе с понятиями наставлений руководства, результат освоения системного мышления при этом будет лучше.

В ходе освоения работы по руководству системного мышления (в книге этого нет, поэтому мы настоятельно советуем брать руководство в Aisystant, онлайн, а не просто читать книгу — книга имеет только сувенирное значение) вам потребуется

[13] https://bureau.ru/soviet/20140629/
[14] https://en.wikipedia.org/wiki/Learning_sciences

- отвечать на вопросы и давать объяснения этих ответов,
- заниматься моделированием в табличках,
- выполнять задания, записывая свои мысли по итогам их выполнения в постах.

Не ленитесь давать обоснования в ответах на вопросы: так вы поднимете беглость в использовании терминологии. Вместо «смутных ощущений» верности или неверности ответов, «мычания в уме» («эээээ» в уме как догадка по поводу ответа так и останется без имени, а ведь цель ответов на вопросы — овладеть терминологией, помнить аргументы) вам придётся выразить рассуждение о верности ответа каким-то текстом, и термины перестанут «вертеться на кончике языка», но не вспоминаться и поэтому оказываться бесполезными для системного мышления. Вам пригодится эта учебная тренировка в написании обоснований ответов на вопросы, когда вы будете общаться с коллегами в рабочей ситуации, буквально на следующий день, будете общаться с единомышленниками в кругу наших инженеров-менеджеров. Им не скажешь аргумент в форме «эээ, у меня ощущение, что ответ такой, но не могу вспомнить, почему». **Аргументация должна быть внятной и выражаться словами, а не «эээ» внутри вашей головы.** Скорее всего, вам придётся посмотреть в текст руководства, чтобы вспомнить аргумент, вспомнить термины и написать обоснование.

Вот типичный отзыв инженера-менеджера[15]:

> Да, похоже, что настоящий game changer — это только упражнения.
>
> Помню, текст 2015 года я читала в полусознанке и процентов 20% (не понимала зачем мне это всё), просто скачала где-то одной кнопкой. Было тяжко продираться, конечно. Когда читала новую версию (прошлого года), страдала от того, как все долго, жидко и повторяется. Recall информации в обоих случаях у себя оцениваю, как одинаковый. Только курс с упражнениями действительно разогнал кашу в голове.

«Только курс с упражнениями действительно разогнал кашу в голове» — это все говорят после выполнения заданий (мы всё-таки предпочитаем говорить не об упражнениях и учебно-тренировочном контексте, а о рабочем контексте в ходе освоения системного мышления, например, контексте стажировки).

За выполнение заданий берётся один человек из десяти, которые начинают освоение системного мышления в режиме «самостажировки». Надеемся, что вы и есть этот один человек, который будет выполнять задания. Если не будете выполнять задания по моделированию, обсуждению рабочих ситуаций и написанию постов в ходе мышления письмом, то каша в голове у вас так и останется, пользы от «чтения текста» (а не «освоения руководства», что подразумевает выполнение заданий) не будет.

Для усвоения материала мы не советуем писать конспекты/изложение материала руководства, не советуем использовать подчёркивание отдельных фраз — это ещё студенческие легенды про то, как надо учиться, городские мифы, не поддерживаемые науками обучения (learning sciences). В многочисленных экспериментах показано, что это бесполезно и никак не улучшает усвоение материала. Мы советуем писать короткие тексты по всем новым мыслям, которые пришли к вам в голову при освоении системного мышления, жанр «сочинения», а не «изложения/шпаргалки». Очень помогает обучению публикация этих заметок,

[15] https://t.me/ailev_blog_discussion/23418

например, в нашем клубе[16]. Аккаунт в какой-нибудь социальной сети тоже подойдёт, хотя там трудней будет получить интересные комментарии, но даже эти комментарии тут не так важны. Ваши собственные мысли, собственные модели, которые будут приходить вам в голову по ходу выполнения заданий — они важны, не теряйте их. Ищите для них слова, записывайте, и даже не столько, чтобы поделиться этими мыслями с миром, сколько для того, чтобы удержать в размышлениях над этим материалом вашу нейронную сетку ещё на некоторое время. Для обучения нейронной сети (неважно, это нейронная сеть человека или реализована кремниевым или даже квантовым компьютером) какому-то набору понятий нужно время работы этой сети с включающим эти понятия материалом. Эта образовательная стратегия мышления письмом/моделированием[17] очень хорошо себя зарекомендовала. Ещё раз повторим, что вы пишете не для внешней аудитории, а для себя: это просто время удержания вашей нейронной сетки в размышлениях над наставлениями нашего руководства[18].

Вопросы. Как устроен текст руководства?

Отметьте правильные высказывания.

--- Материал руководства по системному мышлению тщательно сгруппирован по темам: в одном разделе и далее подразделах раскрывается одна мысль, мысли не повторяются. ### Нет, в одном разделе может быть изложено (даже неполностью) множество мыслей, и мысли эти повторяются. Это не справочник, это руководство/guide: материал для обучения нейронной сети в мозгу инженеров-менеджеров, а не материал справочника для загрузки в компьютер.

--- Каждая тема руководства по системному мышлению излагается компактно и последовательно, чтобы проще было потом находить нужный фрагмент текста с исчерпывающим объяснением. ### Нет, наставления руководства излагаются «вперемешку» (interleaving), причём ожидается, что материал по одной теме излагается с некоторыми перерывами, растянуто во времени (spacing). Текст оптимизирован для обучения, а не для того, чтобы быть справочником.

--- Ваши заметки по руководству нужно публиковать в блоге, чтобы о них узнали нейросетки читателей вашего блога. ### Нет, ваши заметки нужно публиковать в блоге не для того, чтобы кто-то внешний прочёл ваши мысли. Ваши заметки нужно публиковать в блоге, чтобы ваша собственная нейросетка получила дополнительный шанс размышлять над содержанием руководства, а факт публикации нужен для того, чтобы вы не соблазнялись не прописывать полностью термины, опускать важные объяснения и другими способами оставлять в голове вместо слов и фраз неясные ощущения, которые в словах выражаются просто как «ну, понятно». Надо уметь разговаривать на темы системного мышления, для этого надо писать, писать, писать!

--- Автор не знает, что писать термины через слеш прямо не рекомендуется в учебниках по стилю письма. ### Нет, автор знает о том, что написано в учебниках по стилистике текстов. Но текст учебника — это текст совсем не того стиля, для которого написаны руководства по стилям, включая руководства по так

---

[16] https://systemsworld.club/

[17] https://ailev.livejournal.com/1513051.html — в курсе-пререквизите «Рациональная работа» этому учат, как отдельному важному мастерству.

[18] https://www.youtube.com/watch?v=MnCV3sOkVVw

называемому «инфостилю». Наставления руководства — это не реклама, не маркетинг, не научный текст. Это руководство, его задача — научить системно мыслить, причём научить надёжно.

+++ Использование многочисленных скобок и сложносочинённых предложений улучшает текст. Не надо писать короткими удобочитаемыми фразами! ### Да, дополнительные элементы текста для демонстрации его структуры (скобки, показ зависимостей в сложных предложениях) улучшают учебные свойства текста. Были проведены эксперименты, которые показали, что текст с простыми отдельными фразами сложней «собирается» в головах людей. После какого-то количества экспериментов было принято решение оставить стиль текста с обилием скобок и сложносочинёнными (сложное предложение, состоящие из равноправных частей) и сложноподчинёнными (сложное предложение, состоящее из главной и зависимой частей) предложениями.

Разные мышления

Есть два основных цивилизационных пути, условно называемых «восточным» и «западным». **Условная «восточность»** состоит в признании непостижимой сложности мира, невыразимости и непередаваемости человеческого опыта в постижении этого мира. **Условная «западность»** состоит в опоре на рациональность. Рациональность — происходит от латинского ratio, означающего «причину», «объяснение», но также и «отношение», т.е. ассоциируется с делением на части, анализом. Конечно, рациональное (рассудочное, неинтуитивное, не «восточного» типа) мышление в равной мере помогает и синтезу, объединению в целое аналитически разъятого на части. Но в западной культуре исторически придаётся большое значение основанной на логике «аналитике», т.е. формализации и моделированию. Рациональность — один из методов мышления в составе интеллект-стека, подробней о рациональности вы узнаете в руководстве по интеллект-стеку. Формализация и моделирование в рациональности делаются на основе высказанных догадок-объяснений о происходящем в предметной области, а не выводятся «эмпирически» из каких-то наблюдений, поэтому рациональность не только противопоставляется восточной традиции «непостижимости», но и западной традиции «опоры на опыт, на наблюдения, выводимости объяснений из наблюдений». Эти догадки могут затем быть прокритикованы, а по выжившим критику догадкам об устройстве мира принимаются решения о действиях по улучшению мира, «спасению». Теория решений как раз изучается как часть теорий рациональности. Предложение понятия «система» как важнейшего для описания мира — это как раз ход на рациональное познание/«добычу знаний»/learning/cognition.

Можно наблюдать результаты этого «западного» пути развития цивилизации. Именно западная цивилизация дала современные науку и инженерию, опирающийся на компьютеры менеджмент, рынок ценных бумаг как инфраструктуру для перераспределения инвестиций в поддержку новых методов работы[19], то есть инвестиции в обучение агентов новому мастерству, производство нового инструментария.

---

[19] Подробней про преимущества рациональности перед восточным упованием на интуицию и «непосредственное знание» см. в текстах А.Левенчука «Об членораздельное и голографическое в социологии» http://ailev.livejournal.com/1281819.html и «Об интуицию и чуйку» http://ailev.livejournal.com/1295595.html.

Увы, рациональному и логическому мышлению, равно как и многим другим методам мышления, применимым ко многим ситуациям решения самых разных проблем, в школе и вузе сейчас прямо не учат.

Сегодня среди школьных и вузовских педагогов преобладает мнение, что какому-то «хорошему» или «сильному» мышлению (и не спрашивайте, что это такое! Ответа у педагогов не будет!) можно научиться на основе углублённого знакомства с предметами так называемого **STEM**[20]: наука, технология, инженерия, математика. К сожалению, предположения педагогов о косвенном обучении мышлению через обучение предметам STEM не оправдываются, каждому методу мышления нужно учить прямо, а не косвенно[21]. Если вас научили решать физические задачи, то дальше вы не умеете мыслить обо всём, а только умеете решать эти физические задачи. Если научили решать математические задачи, то это умение мыслить не переносится на другие предметные области, рассуждать лучше о танцах или запуске ракет вы не сможете. Так что интеллект-стеку, включая рациональность, нужно учить непосредственно, а не «исподволь» через обучение другим методам работы.

Например, если нужно учить логике, то нужно учить прямо ей, а не через информатику и геометрию. В школьных курсах логика осталась только в рамках изучения логических выражений при обучении программированию и в курсе геометрии, где только и остались доказательства теорем.

Наше руководство по системному мышлению как раз призвано заполнить этот пробел обучения мышлению по интеллект-стеку методов фундаментального мышления, хотя и частично, ибо затрагивается небольшая область объяснений небольшого числа дисциплин, связанных с понятием «система», а не полный набор методов в составе интеллект-стека (скажем, мы практически тут не затрагиваем риторику, не затрагиваем эстетику — а они входят в состав интеллект-стека). Наше руководство наставляет по системному мышлению как использованию понятий системного подхода в самых разных методах мышления, хотя и не касается при этом многих других понятий из этих методов мышления.

Скажем, из входящей в интеллект-стек онтологии **в руководстве по системному мышлению более-менее подробно разбирается иерархия по отношению композиции/часть-целое**, но вот другие виды отношений только упоминаются, подробно не разбираются. А в руководстве по рациональной работе, где подробно изучается рациональное онтологическое мышление, изучаются мыслительные операции с иерархиями как по отношению композиции, так и по самым разным другим отношениям, прежде всего отношениям классификации (присвоение типа), специализации (присвоение подтипа), а также изучается использование инструментария моделеров. В руководстве по системному мышлению мы не наставляем по этим операциям присвоения типа, а просто используем знание о них, считаем их известными из предыдущих руководств. При этом в нашем руководстве достаточно повторений использования этих операций присвоения типов и

---

[20]Определение STEM: https://en.wikipedia.org/wiki/Science,_technology,_engineering,_and_mathematics

[21] Лей Бао и др. показали, что умение рассуждать и тренинг в мышлении на базе какого-то набора концептов это не одно и то же, http://arxiv.org/ftp/arxiv/papers/0807/0807.2061.pdf. Изучение физики оказывается не таким уж "выправляющим мозги" — A historically held belief among educators and researchers is that training in physics, which has a beautiful structure of logical and mathematical relations, would in general improve students' abilities in conducting reasoning that is intellectually challenging. However, the result from this study suggests that training in physics content knowledge in the traditional format alone is not enough to improve students' general reasoning abilities).

выстраивания иерархии отношений, чтобы вы допоняли и поупражнялись в беглости мастерства этих операций, полученного в ходе освоения наставлений руководства по рациональной работе.

Итак, ещё раз (это уже говорилось в первом подразделе, а также подробнее это изложено в руководстве по интеллект-стеку):

- Интеллект — это мыслительное мастерство решения проблем, которые не встречались ранее ни самим людям, ни их наставникам/учителям. Это мыслительное мастерство познания, бесконечного решения всё более и более сложных проблем, в конечном итоге ведущих к выживанию в ходе эволюции (то есть выживанию как отдельного организма прямо сейчас, так и выживанию генома в эволюционном будущем). Решение проблемы — это предложение метода сведения проблемы к задачам, которые имеют метод решения, так что интеллект по факту создаёт новые методы работы.
- Функция::поведение/behavior интеллекта — мышление. Компьютер вычисляет, а интеллект мыслит, мышление — это класс вычислений. Если мы знаем, как решать какую-то задачу (не проблему! Проблема — это когда мы не знаем, как её решить!), то мы не думаем, а просто рассуждаем по известным нам правилам/алгоритмам/дисциплинам какого-то метода (кроме знаний/алгоритмов в методе обычно предусмотрено ещё задействование какого-то инструментария, например, моделера для мышления или даже использование AI для проведения части рассуждений). Без задействования интеллекта будет прикладное рассуждение/вычисление/вывод/inference по прикладному методу.
- Интеллект/мыслительное мастерство представляет собой суммарный набор разных видов мыслительного мастерства, как деятельностных/практических/практичных вычислителей, следующих алгоритмам/знаниям/теориям из методов мышления. Каждый отдельный вид мастерства — это специализированный вычислитель, реализуемый сегодня в его самых сильных/универсальных проявлениях обученной мокрой или кремниевой нейронной сетью человека или компьютера. Мышление тем самым — это вычисление по какому-то методу, в значительной мере нейросетевое, поэтому трудно обсуждаемое как «алгоритмическое», оно происходит чаще всего не в локальных/символьных представлениях, а распределённых. Но трудность представления в знаках не означает, что это «получение информации из космоса/вакуума»! Никакой эзотерики!
- Каждый метод/практика/культура/стиль/деятельность (practice/activity) мышления интеллекта — фундаментальный/безмасштабный. Он включает теорию/знание как трансдисциплину (трансдисциплина/transdiscipline — это теория/знания/объяснения/алгоритмы, использующиеся в самых разных других прикладных методах). Методы мышления интеллекта составляют интеллект-стек (ибо мы условно считаем их упорядоченными в той мере, в которой объяснения каких-то методов позволяет проще объяснять другие методы). Кроме трансдисциплины/«фундаментальных знаний» метод/практика мышления интеллекта подразумевает использование какого-то инструментария в поддержку этой трансдисциплины, ибо мышление выходит за пределы какого-то вычислителя (extended cognition, познание выходит за пределы познающего агента). Раньше инструментарий усиления мышления для фундаментальных методов мышления был ручкой-бумажкой, но теперь чаще всего это компьютер с какими-то программами

моделирования/моделерами. А в случае прикладных методов кроме моделеров на базе компьютеров будут ещё и задействованы инструменты, меняющие физический мир. Это может быть экскаватор или станок с ЧПУ, но в простейших случаях хватает человеческих рук. Даже «наблюдение» — это отдельное действие, для этого может использоваться телескоп на спутнике, но может хватить глаз, поворачиваемых мышцами. Мышление::вычисление тем самым — это функция/метод/практика работы мастерства::вычислитель, включающая задействование объяснений/теорий/алгоритмов (алгоритмы необязательно пошаговые/императивные) из трансдисциплин интеллект-стека, а также подразумевающая поддержку инструментарием моделирования и даже для добычи данных могущая включать изменения мира в ходе измерений или создания новых инструментов. Конечная цель мышления как функции интеллекта — создание прикладных методов, по которым возможны работы по изменению окружающего физического мира к лучшему, например, создание удобных городов, сверхскоростных квантовых компьютеров, здоровых бессмертных тел (включая своё собственное), изобилия вкусной и полезной дешёвой еды. **Интеллект-стек позволяет рассуждать о прикладных методах и создавать их по потребности, а затем по знаниям этих вновь созданных прикладных методов вновь выученное прикладное мастерство задействует инструментарий (иногда уже известный, иногда и специально для поддержки нового метода разработанный) — и изменяет мир к лучшему**.

- Логичности, этичности, алгоритмичности, рациональности и т.д. в мышлении нужно учить через учебники и задания (например, задания по моделированию) этих методов мышления (с опорой на дисциплины/теории/знания/объяснения этих методов и инструментарий этих методов, прежде всего моделеры), а не «исподволь» через учебники и выполнение заданий по каким-то другим учебным предметам. **Обучение физике не даёт прямых знаний по семантике, логике, онтологии — а без этого умнее не станешь! Физики не более умны, они просто больше знают именно физику! Умны те, кто владеет всеми методами мышления из интеллект-стека, к которым относится и физика. При этом физика в интеллект-стек входит не в части прикладных методов мышления каких-то разделов физики (оптики, механики), а в части владения физическим мышлением как таковым: как мыслить об изменениях физического мира в самых разных ситуациях. В том числе само понятие «система» в системном мышлении когда-то пришло из такой физики (а не из какого-то «раздела физики»): система — это часть мира, отделённая как-то (граница системы) от остального мира (окружающей среды).**
- Методологический/трудовой/практический/инженерный/культурный/стилевой кругозор нужно тоже учить (для нейронной сетки это «насмотренность», достаточное число отсмотренных примеров), а не только получать «из опыта жизни на предприятиях», то есть «исподволь», за долгое время. Кругозор менеджмента, обучения людей и AI — это всё надо учить, и учить быстро, а не просто «жить долго, и всё узнаешь».
- Сам же **системный подход (systems approach) — это рациональная (удачная догадка, выдержавшая критику и взятая всерьёз, то есть как основа для принятия решений по поводу изменений мира) идея**

о том, что весь мир состоит из вложенных на много уровней и взаимодействующих между собой систем как физических объектов. Система/system — это взятый вниманием в физическом мире в каких-то границах фрагмент этого мира, взаимодействующий какими-то частями (subsystems) внутри себя, а также взаимодействующий с остающимся по ту сторону границы (boundary) системы остальным миром, называемым окружением/средой/environment.

Вопросы. Типы интеллекта и мышления.

Отметьте верные высказывания:

--- Интеллект — это свойство агентов, которые умеют планировать. ### Нет, интеллект не типа «свойство», это вычислитель (у которого самого есть свойства, например, свойство универсальности).

--- Интеллект — это проявляемое интеллектуальными агентами поведение. ### Нет, ошибка типа: интеллект — это не поведение, это функциональная часть самого агента, чьё поведение — мышление.

--- Мышление — это мастерство агента. ### Нет, ошибка типа: мышление — это не мастерство, это — поведение интеллекта, как мыслительного мастерства.

--- Системный подход — это идея о том, что нужен сильный интеллект. ### Нет, системный подход рассказывает не про интеллект, а про то, что весь мир состоит из вложенных на много уровней и взаимодействующих систем.

--- Системное мышление — это мастерство, проявляемое интеллектом. ### Нет, ошибки типов: 1. Системное мышление — это поведение интеллекта, а не мастерство. 2. Интеллект сам является мыслительным мастерством, а не проявляемым поведением или свойством.

--- Функция интеллекта — мастерство. ### Нет, функция/поведение интеллекта — мышление.

+++ Интеллект — это суммарный набор разных видов мыслительного мастерства. ### Да, интеллект/мыслительное мастерство представляет собой суммарный набор разных видов мыслительного мастерства.

+++ Мастерство — это вычислитель. ### Да, мастерство — это деятельностный/практический/практичный вычислитель, следующий методу/практике. В случае мастерства изменения физического мира, мы обобщаем понятие вычисления по знаниям/теориям/алгоритмам в обработке информации на понятие изменений физического мира по знаниям/алгоритмам, осуществляемое создателем/constructor, в котором есть мастерство и который использует инструментарий/оборудование/аппаратуру для усиления этого мастерства.

--- В руководстве по системному мышлению разбирается главным образом отношение классификации. ### Нет, отношение классификации разбирается как одно из главных в руководстве по рациональной работе, в части наставлений по онтологиям.

+++ В руководстве по системному мышлению разбирается главным образом отношение композиции. ### Да, в руководстве по системному мышлению отношение композиции/«часть-целое» лежит в основе представлений о системах,

как частей во взаимодействии с целым в её окружении, и так на многих системных уровнях, то есть уровнях иерархии по отношению часть-целое.

Критерии сильного мышления

В этом мире интеллект решает проблемы по поводу неодушевлённых предметов, живых существ, разумных существ (людей), организаций этих людей, сообществ и обществ в этом мире, а также моделей мира в людях и компьютерах. Мышление тут вполне деятельностно и инициативно: оно включает действия людей и роботов и их организаций и сообществ с миром и моделями мира. Чтобы о чём-то подумать, нужно на это что-то посмотреть, а для этого повернуть голову. Вот эти действия «повернуть голову и посмотреть», а иногда ещё и «полететь на Луну, и посмотреть» или даже «послать на Луну робота и поглядеть его приборами» мы тоже включаем в мышление. Мышление как проактивное познание включает и работу тела, а также экзотела.

Мышление — это поведение интеллекта в тот момент, когда интеллект пытается найти способы решения проблем, которые раньше ему не встречались[22]. Мышление — это практическое/деятельное вычисление/рассуждение, которое может выходить и в изменение физического мира. Интеллект — это мыслительное мастерство, часть личности. Интеллект реализован мозгом и телом (embodied), или даже совместно работающими мозгом и компьютером (экзокортекс), телом и инструментами (экзотело) обобщённый вычислитель-как-устройство, «мозг с глазками, ушками, ножками и ручками, компьютерами и инструментами». Мы такой вычислитель «с глазками, ушками, ножками и ручками, компьютерами и инструментами» будем называть дальше создатель/constructor. Вычислитель главным образом преобразует информацию по каким-то алгоритмам/знаниям из методов/способов вычислений, создатель — по каким-то алгоритмам/знаниям из методов и с задействованием инструментария/аппаратуры/оборудования (tools and instruments) в конечном итоге не только «рассуждает», но и «действует», то есть преобразует физический мир. Если речь идёт о какой-то части создателя, реализующей именно вычислительную часть алгоритма, мы называем это **мастерством** в методе работы. Интеллект иногда считают только мастерством мышления, работающим с информацией об окружающем мире, а иногда — мастерством изменения окружающего физического мира (то есть в него включают не только функции создания прикладных методов, но и функции самих этих прикладных методов). Мы будем чаще всего говорить об интеллекте как мастерстве, работающем с информацией/описаниями мира с целью создания других мастерств, а когда речь будет идти об изменении мира, то будем говорить о целом агенте (включающем кроме личности, состоящей из многочисленных мастерств, в том числе интеллекта в составе этих мастерств, ещё и организм/тело с инструментами). Тем не менее — вычислитель/computer тоже материален, обработка информации происходит в физическом мире.

Интеллект::вычислитель мы считаем состоящим из двух частей:

- **Врождённый/hardware**, обусловленный биологическими особенностями человеческого мозга и тела или конструктивными особенностями аппаратуры компьютера, реализующего искусственный интеллект (AI). Мозг и тело дают интеллекту огромные возможности (сравните интеллект человека и

[22] François Chollet, On the Measure of Intelligence, https://arxiv.org/abs/1911.01547

шимпанзе, настольного калькулятора и датацентра с программой AI), но и существенно ограничивают эти возможности (сравните интеллект человека на таком классе задач как умножение и деление многозначных чисел и абсолютно неинтеллектуальный электронный калькулятор, при этом хорошо обученная искусственная нейросеть тоже довольно неплохо разговаривает, но плохо умножает и делит многозначные числа, если не обращается при этом к отдельной программе калькулятора).

- **Выученный/приобретённый/software/«виртуальный вычислитель»** в ходе приобщения к человеческой культуре. К AI это тоже относится, ибо AI учится на огромных массивах знаний, накопленных человеческой цивилизацией, в более-менее интеллектуальных реализациях AI речь идёт о подборках текстов с общим объёмом от десятка триллионов знаков. Эта «научаемая» часть интеллекта включает в себя беглое владение ограниченным кругом методов мышления, своими предметами имеющих окружающий физический мир.

Мы не делаем предположений о том, как устроен интеллект в его физической реализации, из каких частей мозга и тела он составляется и как именно они связаны, хотя и высказываем тут догадки по составу интеллект-стека. Подробней об этом говорится в первом же разделе руководства по интеллект-стеку.

Умение мыслить с использованием приёмов/методов/способов мышления из состава интеллект-стека отличает интеллект современного высокообразованного человека (и не только человека, AI тоже) от интеллекта человека-дикаря или интеллекта простейших AI-систем прошлых поколений. «Аппаратура» интеллекта в организмах профессора престижного американского вуза и дикаря из джунглей Амазонки одинакова. Но по мере накопления объяснений о том, как устроен мир, люди (и заодно AI) получили возможность усиления интеллекта в ходе обучения: поменялись и знания/объяснения/алгоритмы мышления, и инструментарий — думают теперь не голым мозгом (как и работают не голыми руками), а с задействованием моделеров и других средств, расширяющих аппаратные возможности мозга.

Интеллект необразованного дикаря существенно ниже интеллекта образованного человека именно из-за отсутствия обучения интеллект-стеку. Если дикаря образовать, то он тоже будет умным! Без образования дикарь не сможет **быстро** решить и сотой части тех задач, которые сможет решить образованный человек. При этом образованный человек ещё и успеет подключить средства ускорения решения проблем: лабораторное оборудование для добычи экспериментальных данных, Гугл или ChatGPT для «подкачки» недостающих знаний.

Ключевое слово предыдущего абзаца: **«быстро»**, ибо всегда во время решения задачи можно для дикаря включить время образования, которое получил образованный человек. Если образованный человек решит задачу за 10 минут, то дикарь сможет решить задачу за 10 лет обучения плюс те самые 10 минут. **Интеллект в значительной своей части выучиваем и может быть усилен инструментально, только небольшая часть его врождённая!** Это в существенной мере объясняет, почему высокий IQ как мера способностей биологического мозга к вычислениям/рассуждениям не так сильно влияет на

результативность в бизнесе, инженерии и науке[23].

И всегда нужно помнить, что как дикарь становится удивительно грозным, когда у него в руках граната, так и усилить интеллект он может, если у него есть подчиняющийся ему компьютерный искусственный интеллект и другие инструменты. Но дикарь не сможет сработать со сложным компьютером и сложным инструментом, так что учиться для усиления интеллекта все равно придётся — просто так умным за счёт компьютера не станешь, просто не сможешь им воспользоваться. Для усиления человеческого интеллекта нужно и образование человека, и мощный компьютер с AI!

В ходе развития человеческой цивилизации выяснилось, каких полезных свойств мы требуем от мышления, которое производит чей-то интеллект: **мышление должно быть системно, абстрактно, адекватно, осознанно, рационально, и это ещё не весь список!** Мы уже упоминали, что есть довольно большой список методов мышления, можно из этих методов выбирать какие-то приёмы, дающие желаемые полезные результаты мышления и избегать приёмов, которые не дают этих полезных результатов.

Системность мышления означает, что весь мир мыслится с использованием понятий системного подхода: как состоящий из взаимодействующих систем, вложенных друг в друга по отношению «часть-целое», причём можно предложить несколько разных способов выделения частей из целого (функциональные части, конструктивные части, пространственные части, стоимостные части, и т.д.). Если мы договоримся, как именно мир разбит на какие именно системы, то мы сможем менять мир коллективно, а если не договоримся, то велик шанс того, что каждый будет менять понравившуюся ему часть мира, а вместе эти изменения не дадут желаемого результата. Понятие системы рассматривается в интеллект-стеке в физике, отношение часть-целое в онтологии, способы деления на нужные для деятельности части — в методологии, приёмы инженерной работы — в безмасштабной системной инженерии. Эта системная инженерия дальше специализируется для систем самых разных системных уровней/уровней организации/эволюционных уровней: менеджмент как инженерия организации, инженерия личности, социальная инженерия как инженерия сообществ и даже обществ. И все эти разные виды инженерии нуждаются в представлении своих целевых систем, а также создающих их систем-создателей именно как систем, то есть требуют системности мышления.

Абстрактность — это главный критерий мышления. Нам в мышлении нужно абстрагироваться от неважного и сосредоточиться на важном. Мышление моделирует мир, а не отражает его в полноте всех ненужных деталей. Мышление должно отделять зёрна от плевел и оперировать зёрнами. Мышление должно уметь

---

[23] https://www.quora.com/What-was-John-von-Neumanns-IQ-Was-he-smarter-than-Einstein — сравнение Альберта Эйнштейна и Джона фон Неймана, они оба были гениями. Нейман признавался всеми как ну очень умный и скоростной, явно умнее и быстрее Эйнштейна, с IQ явно побольше. И научных достижений у него было неисчислимое количество. Но вот Эйнштейн признавался всеми как более медленный, но более глубокий учёный. В зачёт бралась глубина и новизна идей, а не скорость и количество спродуцированных идей, которые потом отнюдь не все пройдут проверку критикой и экспериментом. IQ у ещё одного гения физики, Ричарда Фейнмана, был 125, https://www.quora.com/Was-Richard-Feynmans-IQ-really-125. Весь вопрос, оказывается, не в большом IQ, а в том, как употребить мозг! Если мозг употребить правильно, то и небольшого IQ хватит, чтобы оставить след в истории. Если неправильно, то и большого IQ не хватит, чтобы хоть что-то изменить в мире к лучшему. И уж тем более IQ не определяет, насколько вы будете хорошим человеком.

отвязываться от индивидов и мыслить типами, прототипами, абстрактными понятиями: мы требуем какого-то обобщения с опусканием ненужных для предмета мышления деталей, это резко упрощает вычисления мышления и упрощает обучение мышлению. Нам нужна абстрактность в сложных ситуациях, мы хотим уметь планировать и проектировать впрок, мы хотим работать с целыми классами и типами ситуаций. Без абстрагирования мы не сможем переносить опыт одних ситуаций на другие, мы не сможем эффективно учиться, мы не сможем создавать языки, обслуживающие коллективное мышление — языки позволяют обмениваться самым важным по поводу обдумываемых ситуаций, они очищают общение от неважных подробностей. Абстрактность рассматривается в понятизации, математике, онтологии, логике.

Адекватность — это возможность проверить, связано ли наше абстрактное мышление и порождаемые им описания ситуаций с реальным миром, или оно оказалось отвязанным от вещного/физического мира и у нас нет способов проверить его результаты, соотнести его результаты с реальностью. Адекватны ли наши мыслительные представления о ситуациях реальному (т.е. существующему независимо от нас, материальному/физическому) миру? Или мышление нас обманывает и предлагает какие-то неадекватные представления? Нам нужно практичное, применимое для действия мышление, мы хотим быть адекватными и не отрываться от реальности. Адекватность рассматривается в семантике, физике, онтологии.

Осознанность — это возможность понять, как мы мыслим, как мы рассуждаем. Если мы просто «имеем интуицию», это нас не удовлетворит. Мы не сможем научить других мыслить, научить их повторять наши рассуждения. Мы не сможем заметить ошибку в нашем мышлении, не сможем его улучшить или изменить, не сможем выучить другой способ мыслить, ибо мы его не будем замечать, не будем его осознавать. Мы не сможем удерживать внимание в мышлении, ибо нельзя удерживать внимание на том, чего не осознаёшь. Мы не сможем предъявить неосознаваемое нами мышление для проверки со стороны логики и рациональности, не сможем сознательно принять решение о том, что в той или иной ситуации нам достаточно от мышления интуитивной догадки, а не строгого рационального рассуждения. Мы хотим знать, о чём мы размышляем, как мы это делаем, мы хотим иметь возможность выбирать — мыслить нам о чём-то или не мыслить, мы не хотим быть бессознательными мыслящими автоматами. Мы хотим быть осознанными в мышлении, мы должны учитывать не только мышление, но и наличие самого мыслителя как интеллекта, производящего мышление::вычисление. Осознанность рассматривается в понятизации, собранности.

Рациональность — это возможность провести принятие наименее плохого решения о способе действия на основе рассуждения по теориям принятия решений, учитывающих неопределённости. Это возможность отстраниться от своей биологической и социальной природы, не делать связанных с этим ошибок. Рациональность — это возможность проверить результаты быстрого интуитивного мышления, выдающего догадки, на отсутствие ошибок, нарушений правил, возможность задействовать опыт человечества в мышлении. Это возможность явно (хотя бы в диалоге с самим собой, то есть осознанно) обсудить эти выработанные цивилизацией правила хорошего мышления, обсудить логические основания мышления, обсудить допустимость или недопустимость использования каких-то отдельных приёмов мышления. Мы не хотим ошибок мышления, поэтому мы должны

быть рациональными, мы должны уметь распознавать ошибки мышления у себя и других, мы должны уметь выразить результаты мышления так, чтобы уменьшить число ошибок при восприятии наших результатов другими людьми. Мы хотим быть рациональными, нам нужно уметь делить задачи на части (рацио — это ведь «деление»), мы не хотим чистой интуитивности или чистой эмоциональности-спонтанности, хотя мы не отрицаем их необходимости, но нам прежде всего нужна цивилизованность в мышлении, использование лучших достижений цивилизации в том, как мыслить. Рациональность рассматривается в логике, а ещё и в методе мышления, так и называемом — «рациональность». Рациональность как метод мышления занимается изучением генерирования догадок-опций при принятии решений на основании каких-то моделей, а также включает теорию принятия решений, занимающуюся изучением принятия наилучших для деятельности решений. Все решения принимаются по поводу изменений модели себя, модели мира, изменений степени уверенности в точности модели себя и модели мира, изменении себя или мира, что изучает подход active inference[24].

Все остальные **критерии сильного мышления** — это чаще всего частные варианты или сочетания представленных. Так, **«сильное мышление»** обычно сводится к высоким характеристикам по всем критериям (мышление сильного интеллекта), **«мудрость»** — это просто другие слова для адекватности и системности, **«творческое мышление»** — это задействование правильного абстрагирования и высказывание труднокритикуемых догадок по поводу важных объектов, **«рефлексия»** — это осознанность, но только не на текущую ситуацию, а уже прошедшую.

Легко придумать и какую-то другую выборку «мышлений» как мыслительных частных методов из полного набора методов интеллект-стека — по образу и подобию того, как выбрали «системное мышление». Помним: метод/практика/культура/стиль — это теория/знания/алгоритм/«учебная/научная дисциплина» как приёмы мышления о каких-то важных объектах, причём эти приёмы мышления поддерживаются кроме вычислений мастерства ещё и инструментарием, например моделерами для записи представлений мыслительных объектов и операций с ними.

Легко представить себе такую выборку приёмов мышления из полного набора методов интеллект-стека, как, «деятельное мышление». **«Деятельное мышление»** будет включать в себя главным образом рациональное мышление с акцентом не столько на познание, сколько на выход в (инженерные) проекты по изменению мира, задействование прикладных методов системной инженерии в её приложении к каким-то определённым видам систем. Его же можно назвать и **«предпринимательским мышлением»**, в бытовом смысле слова «предприниматель». В деятельном/предпринимательском мышлении одна система принимает решение о том, как изменить мир, в том числе и себя — «что-то предпринять», создать или изменить какую-то систему, предпринять инженерный проект, или «исследовать возможность выгодно изменить мир», что можно описать и как **«исследовательское мышление»**. Деятельное мышление можно назвать и **«трудовым мышлением»**, оно же будет **«практическим мышлением»**, **«проактивным/enactive мышлением»** и даже просто **«инженерным мышлением»**, ибо каждый инженер делает шаг в неведомое, планируя полезное

[24] https://ailev.livejournal.com/1611838.html, https://ailev.livejournal.com/1612513.html

в будущем изменение мира (предпринимательский/проактивный/творческий/исследовательский/деятельный/культурный шаг) и выполняя потом актуальное изменение («рабочая» часть инженерии, «изготовление»). Почему всё это более-менее синонимично, станет понятно по мере освоения мышления по нашему руководству, это всё отсылки к дисциплине методологии, учения о деятельности.

Мышление не «пассивно», не «аналитично», результат мышления — это не «компьютерная выдача» в виде «аналитического отчёта о раздумьях». Нет, мышление проактивно/деятельно/практично/трудово/инженерно, и к системному мышлению это тоже относится. **У нас не «системный анализ», а «системное мышление», мышление меняет мир, а не только «понимает мир»! Если ваше системное мышление не нацелено на изменение мира к лучшему, то это вряд ли системное мышление!**

(картинка Dan Reynolds[25])

Мы вовсе не имеем в виду, что агент (человек или AI, или человеко-компьютерный киборг, или даже организация) с сильным интеллектом, умеющий абстрактно, адекватно, осознанно, рационально, системно мыслить, сможет решить любую задачу. Нет, для этого ему нужно обладать кроме мыслительного мастерства интеллект-стека ещё и мастерством **предметных/прикладных** (domain-specific) **рассуждений** — по прикладным (то есть используемым в рабочих проектах) методам/«видам (инженерного) труда». Агенту, чтобы выжить и ему как «организму», и ему как «виду», нужно быть практичным/деятельным/предприимчивым/творческим/proactive/enactive. Дальше

[25] https://en.wikipedia.org/wiki/Dan_Reynolds_(cartoonist)

это же рассуждение может быть применено ко всей эко-системе, ибо все виды эволюционируют взаимозависимо, это относится и к дарвиновской биологической эволюции, и к техно-эволюции.

Каждый вид труда/деятельности имеет какие-то свои специфические предметные/прикладные рассуждения и действия, исполняемые той функциональной частью мозга и тела, которые мы назвали бы прикладным мастерством по отношению к мыслительному мастерству. Прикладное мастерство важно, оно позволяет рассуждать быстро и без типичных для новичков в этих видах труда ошибок.

Вообще, системное мышление даёт небольшой набор типов, который позволяет быстро ориентироваться в ситуации. Это такая высокоуровневая (высокий уровень абстракции, мета-мета-модель) модель ситуации, «карта». Карта не говорит, куда идти, но знание карты позволяет не заблудиться, не пройти в болото. Карта, конечно, может использоваться, чтобы рассматривать её перед сном, многим людям такое нравится, они коллекционируют атласы. В плане мета-мета-модели это аналитики, они не выходят в действие, не меняют мир. Они именно что «диванные аналитики». Нет, системное мышление используется по-другому:

- Вы выделяете вниманием какие-то объекты в мире, типы этих объектов задаются мета-мета-моделью. По сути дела, у вас есть карта, и вы «ориентируетесь на местности»: отождествляете объекты местности с объектами на карте. Ура, вы теперь знаете, где находитесь (а если какие-то объекты не найдены, но на карте они есть — вы знаете, где их искать).
- Но теперь вы должны знать, куда вам, собственно, надо идти. И вы можете сгенерировать десяток маршрутов, покритиковать их, рационально выбрать лучший.
- И главное: пойти!

Заменяет ли системное мышление прикладное мышление?

Одна из неправильных идей состоит в том, что можно иметь сильный общий интеллект, в том числе развить у себя системное мышление — и иметь огромное преимущество перед профи в своих предметных областях. Грубо говоря, вы станете очень умным, а затем будете иметь преимущество перед, например, более глупым пианистом или даже сварщиком. Увы, это не сработает. Человек с системным мышлением будет иметь перед профи преимущество в том, с какой скоростью он разберётся в проекте в целом, как быстро договорится с остальными участниками проекта (и даже больше: договорит всех вокруг себя), насколько сможет удерживать внимание на главных задачах проекта и не увлекаться чем-то не слишком важным в ходе работы. А ещё он сможет быстро чему-то научиться. Но у него не будет немедленного преимущества в решении прикладных задач! Никакое системное мышление не поможет вам сразу начать играть на скрипке, или пользоваться сварочным аппаратом, или даже ставить медицинские диагнозы, если вы этим никогда не занимались. Не сразу — да, обучитесь всему этому быстрее, но вот просто стать умным и считать, что это даёт преимущество в какой-то прикладной дисциплине перед профи в этой дисциплине — это ошибка. Если вы станете профи в этой прикладной дисциплине, то будете больше профи, чем ваш менее умный коллега. Но если не станете профи — будете «вообще умным», «вечно перспективным».

Более того, ошибки в системном мышлении могут появиться из-за игнорирования полного набора фундаментальных методов мышления в составе интеллект-стека. Если вы хорошо разобрались с тем, как строить иерархию по отношению композиции, но плохо понимаете отношение классификации, вам системное мышление не поможет, ошибки в мышлении будут по другим причинам, вам нужно будет доразобраться с онтологией (например, повторно разобраться с наставлениями руководства по рациональной работе, ибо мастерство рациональной работы — пререквизит к мастерству системного мышления, сначала надо освоить мышление и работу по одному руководству, а затем по другому.

Системное мышление не заменяет прикладных/предметных рассуждений, равно как не гарантирует хорошего мышления по всем фундаментальным методам мышления из их интеллект-стека. Например, системное мышление не гарантирует рациональности: не факт, что если вы мыслите системно, то вы обязательно будете принимать хорошие решения, пользоваться современной теорией решений!

Для того, чтобы видеть ошибку в «2*2=5», нужно по-прежнему знать арифметику, никакое системное мышление тут не поможет. Если вы не умеете ремонтировать унитазы, а вам это потребовалось, то вам поможет не руководство по системному мышлению, вам поможет учебник сантехники. Но системное мышление усиливает, направляет и дополняет прикладные рассуждения, а также рассуждения в рамках фундаментальных методов мышления. Например, системное мышление поможет выбрать современный/лучший учебник сантехники из многих имеющихся, разобраться в ситуации ремонта унитаза в целом. Вдруг унитаз этот вообще не нужно ремонтировать, а проблема в чём-то другом: унитаз тут «симптом», а не «болезнь»! Но фундаментальное системное мышление не заменит прикладного сантехнического мышления. Обучение по инженерии систем-унитазов придётся-таки получить, освоить прикладной инженерный метод/культуру/практику этой работы, то есть получить знания сантехники как учебной дисциплины/объяснений, в том числе знания по использованию тамошнего инструментария (например, знание о том, как использовать разводной ключ, трос с ершом для прочистки труб, пробивные штанги с разнообразными насадками для разных типов засора).

Если вы собираетесь решать задачи какой-то прикладной предметной области без знания SoTA (state-of-the-art, лучшее на сегодня известное знание) методов работы в этой предметной области, опираясь только на смекалку и сообразительность, то мы назовём это кулибинством[26]. Кулибинство — это народное изобретательство, без опоры на современные методы работы с их современными знаниями и задействованием современного инструментария. Кулибинство — это что-то типа знахарства, только в инженерии.

Иногда кулибинство срабатывает и даёт «работоспособную систему», но в серию на рынок такую систему не выпустить. «Работоспособная система» по итогам кулибинства — это необязательно лучшая в своём классе по характеристикам, надёжная, дешёвая в эксплуатации и готовая к массовому выпуску. Нельзя игнорировать достижения человеческой культуры, считать, что «творчество» — это когда ты всё сам придумываешь. Вопрос в том, что качественное творчество должно быть чем-то заведомо лучше, чем уже имеющиеся варианты. Поэтому о имеющихся вариантах инженерных идей нужно как минимум знать, чтобы иметь возможность

[26] Вот примеры кулибинства: https://vk.com/club45696675

сравнить свои придумки с уже имеющимися в культуре.

Нужно как минимум гуглить прикладное знание/объяснения, спрашивать его у интеллектуальных ассистентов и со-пилотов. Ещё лучше — освоить прикладной метод работы из учебника, ещё надёжней — закончить учебные курсы по прикладному методу. Профессионал — это тот, кто не делает новичковых ошибок (знает о них!), а не тот, кто вообще много знает[27].

Плохо действовать всегда методом проб и ошибок, уповая на «свободу творчества» и приговаривая «некогда исследовать вопрос, некогда учиться, работать надо». «Изобрести что-то на коленке» — это ж и есть «попробовать, вдруг получится», такой метод техно-эволюции в конечном итоге очень дорог, разве только у вас миллионы лет в запасе, как у природной эволюции. Конечно, метод проб и ошибок в инженерии используется, в инженерии он признан, но он не главный. Если бы метод кулибинства был главным, то инженерам не нужно было бы образования.

Системный мыслитель — это не тот, который игнорирует учебники, стандарты, регламенты по прикладным методам работы. Совсем наоборот: это тот, кто может быстро выбрать необходимый учебник или найти современный стандарт работы, разобраться в их содержании, учесть особенности текущей ситуации с задействованием всех других прикладных методов работы в сложном командном проекте. Системное мышление помогает прикладному мышлению, а не заменяет его.

Место системного мышления среди других мышлений: интеллект-стек

Есть разные мнения о том, можно ли называть прикладное мышление (например, мышление инженера-разработчика ракетной техники, или ведение обучения людей с использованием педагогического метода blended learning, или ремонт унитазов на космических кораблях) мышлением. По одному мнению, конечно, агенты всегда мыслят, но согласно другому мнению, мышление-— это только функция порождения нового прикладного метода с его знаниями/дисциплинами как потенциальными алгоритмами изменения мира для ситуаций с впервые встреченной проблемой и инструментарием для этих изменений, а вот для вычислений/рассуждений прикладными методами собственно мышления уже не требуется, эти рассуждения идут «автоматом». Работу калькулятора не называют «мышлением», он считает что-то «аппаратно», и всё. По другому мнению (которого придерживаемся и мы) в ситуациях прикладного метода (даже работы калькулятора, например, ребёнка, мучительно умножающего трёхзначные числа в столбик на бумаге) мы всё-таки будем говорить о прикладном мышлении. В какой-то заданной прикладной предметной области можно хорошо понимать, что делать в типовой ситуации — и делать «на автомате», не задумываясь о методе работы, «не думать». Но в этой прикладной предметной области может быть очень много объектов, так что надо будет:

- составить самостоятельно какой-то метод решения (объяснения/знания/алгоритмы и инструменты в их поддержку) конкретной проблемной ситуации, конкретного затруднения, даже не выходя за пределы конкретной предметной области (иногда составление метода для какой-то проблемной ситуации называют стратегированием, а найденный метод — стратегией),

[27] Это определение приписывают Нильсу Бору.

- затем спланировать работы (метод — это только способ выполнения работы, стратегия не предусматривает плана с проставленными в нём сроками работ и ресурсами), исходя из наличных или ожидаемых ресурсов (а если ресурсов не хватает, то откорректировать метод, заменив таким, для которого ресурсов хватит, или предусмотреть работы по каким-то методам добычи нужных ресурсов), а
- затем только выполнять работы по этому методу-стратегии, и ещё
- отслеживать, удовлетворяют ли результаты выполняемой работы, или надо срочно адаптировать метод, поскольку или ситуация с исходными данными или требуемыми результатами изменилась, или были ошибки в предыдущих шагах.

То есть для элементарных каких-то операций/действий в мире мышления вроде не надо, а мышление нужно для выбора какой-то объяснимого способа/метода работы, в составе которого множеством самых разных простых операций, когда требуется учитывать много дополнительных обстоятельств, удерживать внимание на огромном числе объектов, меняющихся на каждом шаге следования методу работы-— и вот тут мы склонны говорить о прикладном мышлении, а не просто об автоматическом выполнении отдельных операций. Ударить молотком, то есть просто двинуть мышцами молоток — большого ума не надо, но вот чтобы ударить безопасно, в нужное время, в нужное место — вот для этого мышление уже надо, надо как-то обсудить метод ударов молотком, знания/дисциплины и инструментарий этого метода.

Сразу освоить прикладное мышление для решения проблем/затруднений в рабочей предметной области, да ещё потом и сочетать эти разные прикладные мышления для разных методов работы в сложных проектах, в которых задействованы сотни людей, не удаётся. Но и после освоения узкого прикладного мышления по одному методу надо признать, что без опоры прикладного знания на фундаментальное/трансдисциплинарное знание хорошо действовать в реальном мире не получится. На стыках работ по любым прикладным методам будут встречаться ситуации, не описанные ни в одном учебнике, ни в одном регламенте или стандарте работы! Люди просто обязаны использовать фундаментальные знания/объяснения человеческой цивилизации, ибо только они позволяют связать в мышлении между собой как знания разных прикладных методов, так и знания о работе на цивилизационном фронтире: решении проблем, методы типовых решений для которых ещё никто на Земле не выработал, поэтому эти решения нельзя вот так просто взять и нагуглить, понять — и затем применить не думая. Хотя можно спросить метод решения проблемы у AI, но не факт, что этот AI будет достаточно умным, чтобы выдать что-то толковое (спросите AI сегодня: как стать бессмертным? Не факт, что получите в ответ описание какого-то подходящего метода решения этой проблемы).

Если мы хотя бы частично что-то знаем о структуре мира, это на несколько порядков уменьшает количество вычислений интеллекта/объем мышления в неполностью известных нам предметных областях. Это много? Скажем, какую-то проблему P мы можем решить человеческим мозгом за десять тысяч лет. Это побольше, чем время существования человеческой цивилизации, хотя вы можете сократить это время до десяти лет, если будет работать тысяча мозгов, а у вас есть ресурсы для поддержания жизни тысячи человек, и ещё вы знаете, как организовать эффективно разделение работ на тысячу человек. Так что лучше сделать какие-то

не случайные, а уже известные цивилизации предположения о структуре задачи и её предметной области. В нашем примере проблемы P они позволят снизить объем вычислений одного мозга в десять тысяч раз, задача будет решена одним человеком за год. На кону примерно такая разница между скоростями работы необразованных людей и образованных: **необразованные люди (дикари) знают мало общих объяснений об устройстве мира, а образованные — много. Надо учиться, чтобы быстро решать задачи собственным мозгом. Вариант: надо учиться, чтобы сдвинуть этот объём вычислений на мозги других людей и на компьютерные инструменты (включая AI).**

Цивилизация при помощи науки с её опорой на письменное накопление объяснений/теорий/знаний/моделей даёт нам в готовом к изучению виде догадки об устройстве мира, а также учит формулировать проблемы (которые не знаешь как решить, предмет работы интеллекта) и переводить их в задачи (которые известно как решать, предмет работы прикладного мастерства). Эти догадки и лежат в основе образования. **Образование — это усиление возможностей интеллекта путём обучения интеллект-стеку. Образование тем самым — это специализация обучения (образование::обучение).** Напомним: обучение — это метод создания мастерства выполнения работ по целевому методу, которому учат. Образование даёт возможность быстрее находить прикладные методы по преобразованию проблемы в задачи, т.е. переводить ситуацию «не знаю как к такому подступиться» в ситуацию «знаю, по какому методу надо работать, чтобы получить результат — какие использовать знания и инструменты».

Приобретённый в ходе образования интеллект::мастерство позволяет решать проблемы в десятки тысяч раз быстрее, чем это могло бы быть сделано необученным структуре окружающего мира естественным врождённым интеллектом homo sapiens. Цивилизованный мозг — это не «дикий», это обученный мозг, он быстр в мышлении, а современный мозг ещё и использует мозги других людей (коллективная мыслительная работа) и компьютеры (классические и с программами AI) для усиления скорости своего мышления. При этом компьютеры могут быть использованы в минимальных вариантах даже не за счёт компьютерных вычислений, а просто за счёт помощи в организации памяти и удержании внимания. Компьютер как «ручка-бумажка» тоже крайне эффективен для мышления! Умный и ленивый образованный человек с ноутбуком может сделать много больше, чем толпа деятельных, но необразованных дураков-дикарей!

Освоение нового мастерства идёт у человека не через «природную смекалку», а через «облагороженную образованием смекалку», через знания/модели/объяснения/теории/дисциплины о структуре мира, структуре проблем и задач, а также знания/объяснения/дисциплины о том, какие доступны инструменты (например, компьютеры как универсальные моделеры для системного моделирования самых разных объектов).

Всё то же самое относится и к AI. Изготовленные на заводе компьютеры для AI тупы, они могут выполнять только простейшие операции типа перемножения матриц. А вот после обучения на огромном объёме знаний, уже накопленных в письменном виде цивилизацией, в этих компьютерах появляется мастерство рассуждений на основе этих знаний, «большие языковые модели/large language models/LLMs», которые иначе называют «фундаментальными моделями/foundation models», имея в виду как раз их трансдисциплинарный характер. Это аналог «образования»: обучение мышлению и каким-то инженерным кругозорам. А потом

такие фундаментальные модели легко или дообучать прикладным знаниям (finetune) или подключать к таким моделям прикладные знания в виде каких-то инструментов (скажем, подключать Wolfram Mathematica для решения математических задач).

Умение и навык, скилл — это отсылки к владению агентом мастерством исполнения работы по какому-то методу, опирающемуся на теорию/знание/объяснения/алгоритм/дисциплину, причём выполнение этого метода/способа работы поддержано каким-то инструментарием. Интеллект — это мастерство владения набором фундаментальных методов мышления, нужных для самого обсуждения методов в условиях, когда непонятно, какой метод применить (возможно, такого метода ещё нет — или нет знаний, или нет инструментария, их нужно ещё создать).

Интеллект::мастерство работает с прикладными методами (и тем самым их знаниями/дисциплинами/алгоритмами/теориями) как объектами своей работы. Можно сказать, что интеллект как мастерство фундаментального мышления как раз создаёт и дальше развивает прикладные методы, он как раз нужен для познания, для бесконечного роста знаний (эволюции знаний) и инструментария поддержки работы с этими бесконечно растущими знаниями. Больше знаний и поддерживающего эти знания инструментария — больше перевода всё самых разных проблем в задачи. Инфекционные болезни были проблемой, но вот знание о микробах и мыло в качестве инструментария с методом гигиенического мытья рук в существенной мере решили эти проблемы, борьба с инфекциями стала набором задач, а не проблемой: известно, что делать, надо только найти ресурсы, и дальше просто делать.

Интеллект в его врождённой части позволяет людям быть умней кошек и обезьян, а вот в полученной образованием/выученной его части — это полученная образованием машинка по получению прикладных дисциплин. Интеллект — это эволюционно полученный людьми инструмент познания, машинка по разработке способов решения проблем — превращения проблем, которые не решаются никаким известными методами в выполняемые/решаемые известными прикладными методами задачи. Это относится и к естественному интеллекту, и к искусственному, и к гибридному, и к коллективному.

Трансдисциплины/«фундаментальные дисциплины» — это и есть объяснения/теории/знания/модели/алгоритмы по поводу устройства мира. Они удобны для скоростного мышления о мире, удержания внимания на вычислениях/рассуждениях/мышлении о важном, сохранении ресурса мозга или компьютера от разбазаривания на мышление о неважном. А само это мышление по знаниям/алгоритмам трансдисциплин затем нужно для создания методов изменения мира к лучшему.

То, что занимает у очень смекалистого дикаря полжизни, у обученного фундаментальным дисциплинам человека может занять несколько часов, или даже несколько секунд — особенно, если учитывать, что знания/алгоритмы методов мышления включают в себя и знания по задействованию инструментария мышления (чаще всего это моделеры). И то же относится к компьютерам с AI, только у этих компьютеров нет биологических ограничений по скорости и объёму вычислений для одного агента, почему их и боятся примерно так же, как в голливудских фильмах боятся гениальных учёных-злодеев.

Интеллект — это мастерство беглого задействования целого стека/stack/стопки/слоёного пирога поддерживающих друг друга фундаментальных методов мышления. Этот набор методов мышления (опирающихся на задействование фундаментальных дисциплин и использование инструментария их поддержки) мы называем **интеллект-стеком.**

В фундаментальных **методах** мышления дисциплины/теории/знания будут только «алгоритмической» частью. Мы эти дисциплины/теории/знания смело будем считать ещё и «алгоритмами» (описаниями задействования метода в самых разных обстоятельствах/ситуациях — ровно как алгоритмы могут быть использованы для вычислений с самыми разными входными данными):

- Есть множество указаний на то, что конструктивная математика — это по факту переход от декларативных (объекты и отношения) описаний к описаниям через операции построения объектов. Это можно распространить на всю работу с понятиями (ментальными/математическими объектами).
- В компьютерной науке давно получены результаты, которые позволяют рассматривать самые разные виды представления алгоритмов, а не только «пошаговое выполнение императивных программ» (в том числе соответствие Curry-Howard[28] между императивным алгоритмом и набором логических высказываний). Мы достаточно широко трактуем этот результат.
- Надо рассматривать не знания сами по себе, а то, что с ними делает вычислитель — в данном случае это мастерство выполнения метода, использующего знания для вычислений (мышление) или даже изменения мира (мышление и задействование инструментов). В теории создателей (constructor theory)[29] проводится обобщение понятия «алгоритм» на описание преобразований не только информации и сверхинформации (superinformation, в квантовых компьютерах, представленной не в битах, а кубитах), но и теоретически любых физических преобразований.

Методы мышления, как и любые другие методы, используют не только понятия из теорий/знаний/объяснений/алгоритмов/дисциплин (в том числе трансдисциплин), но и **инструментарий,** понимаемый как набор расширяющих возможности тела агента инструментов/аппаратуры/оборудования. В случае трансдисциплин мышления инструментарием обычно будет **моделер** (простейший из ручки-бумаги, или программа какого-то моделирования для компьютера), а расходным материалом к моделеру идёт кофе для человека-модельера и электроэнергия для компьютера, воплощающего модель. Другие инструменты в интеллект-стеке редки, хотя бывают. Например, в понятизации используется тело, там ищутся какие-то ощущения, которые потом надо будет перевести в мысль, роль исполнителя метода понятизации — «поэт».

Несмотря на практичный характер мышления, интеллекту больше нужно моделировать мир, то есть заниматься познанием/cognition/learning, созданием моделей, нежели непосредственно его менять в действии — но помним, что это модели, как раз нужные для изменения себя и мира к лучшему, причём интеллект принимает в случае затруднений решения о том, менять ли модель мира, модель себя, себя или мир — причём взаимозависимо.

В целом мышление как познание происходит в конечном итоге методом деятельных

---

[28] https://en.wikipedia.org/wiki/Curry%E2%80%93Howard_correspondence

[29] http://www.constructortheory.org/

проб и ошибок, то есть не только высказыванием «умственных» догадок-объяснений и их умственной же критикой, а «активным зондированием» физического мира, деланием догадок и наблюдением результатов — получилось или нет. Это происходит даже по поводу того, где граница между самим агентом и окружением — чем можно просто командовать, на что можно существенно влиять, на что можно влиять несущественно, на что не удаётся пока влиять. Агент непрерывно что-то делает с окружающей средой, чтобы понимать границы своего влияния — и менять среду к лучшему (для себя ли, для своих генов, для популяции — это отдельный вопрос).

В случае перехода к прикладной инженерии (как изменению мира) методом «проб и ошибок» в старой и известной для этого агента или даже новой малоизвестной агенту предметной области, для изменения мира агент задействует изобилие самого разного инструментария и применяет самые разные исходные материалы: станки, химические реагенты, дрессированных животных, солнечный свет, воду в пруду, часы, балетный станок, квантовый компьютер, и т.д.

Есть некое лукавство в том, что мышление по фундаментальным методам — это чисто «ментальный акт». Ввод-вывод в вычислитель (например, мозг у людей) вполне материальны и требуют инструментария/оборудования (книгопечатание, электронные онлайн-курсы, мессенджеры для получения проблем и отправки решений), да и сам вычислитель вполне физический объект. Как любит напоминать Дэвид Дойч, математик и астрофизик — вполне себе физические объекты, «умственный труд» требует физичности трудящегося! В наших руководствах мы используем понятие «создатель», который является обобщением компьютера, способного выполнять алгоритм вычисления на систему-создателя/constructor из constructor theory, который способен выполнять знания/алгоритм метод как «алгоритм изменения физического окружения»[30].

Метод работы, выполняемый создателем включает знаниевую часть (которую программно-аппаратно реализует в создателе мастерство выполнения алгоритмов/объяснений/теорий метода) и аппаратную часть (мастерство в его поддержке телом агента можно тоже отнести к этой аппаратной части, а дальше идёт аппаратура инструментов как датчиков и актуаторов, а также «экзотела» как платформы для всех этих датчиков и актуаторов, то есть оборудование/аппаратура, которые помогают мастерству делать дополнительные вычисления и действия по измерениям в физическом мире и изменениям физического мира). Для универсальных создателей (интеллектуальных агентов) можно говорить не просто о вычислениях, а сразу о мышлении, а также рассматривать ситуации, когда в ходе выполнения метода создатель сначала строит дополнительную аппаратуру — инструментарий (раскрутка/bootstrapping).

Constructor theory даёт обобщение для понятий

- алгоритма (термин остаётся тем же) как описаний/теории/дисциплины,
- измерения (ввод, физическое взаимодействие для получения данных)
- изменения (обработка/processing в случае информации, в случае вещества — transformation)
- вывода (в случае создателей отдельно эта операция не рассматривается, включается в transformation)

[30] https://www.constructortheory.org/, https://www.youtube.com/watch?v=40CB12cj_aM

Так что можно дальше обсуждать преобразования не только информации, но и физических объектов, а также понятие **вычислителя/computer**, реализующего «обработку информации по какому-то алгоритму»::метод расширять до понятия создателя/constructor, реализующего «преобразование физического мира по какому-то алгоритму»::метод.

Универсальный компьютер при наличии достаточных вычислительных ресурсов (памяти и времени) может принципиально выполнить любое вычисление, которое может выполнить машина Тьюринга (известная нам теория компьютинга), а вот универсальный создатель с учётом раскрутки/bootstrapping (например, начиная с выплавки металла из руды, получение чистого кремния для полупроводника из песка) принципиально при наличии достаточных ресурсов (памяти, времени, начального набора инструментов) может выполнить любое преобразование вещества — и «любое» понимается как в математике, речь тут идёт о теоретической возможности. Практическая же возможность будет ограничена ресурсами и рисками какой-нибудь катастрофы (скажем, прилетает астероид и уничтожает создателя, который имеет все необходимые ресурсы — и он не успевает закончить свою работу.

При этом особо подчёркивается, что по одному методу (алгоритм/теория/объяснения/знания и аппаратура/инструменты) создатель способен выполнить множество вычислений, оставаясь при этом неизменным — примерно так же, как молекула катализатора (простейший создатель) может выполнить множество актов катализа, оставаясь при этом неизменной.

Вопросы. Системный анализ, необходимость прикладного мышления.

Отметьте правильные высказывания.

--- Руководство по системному мышлению основным своим содержанием имеет системный анализ. ### Нет, руководство по системному мышлению учит менять мир. У нас не только наставления по системному анализу! Нам недостаточно только понимать и находить ошибки! Инженерам-менеджерам нужен системный синтез, умение решать проблемы, а не только их анализировать!

+++ Образование включает в себя обучение системному мышлению. ### Да, образование состоит в обучении интеллект-стеку, а методы в составе интеллект-стека включают в себя и методы системного мышления.

+++ Системное мышление в качестве инструментария использует моделер. ### Да, системное мышление — это набор приёмов мышления из интеллект-стека. Методы включают в себя дисциплину/знания/теорию и инструментарий — и моделер как раз такой инструмент для усиления мышления. Так что системное мышление использует моделер как инструмент для системного моделирования.

--- Интеллект-стек экономит мышление за счёт тщательно выбранной терминологии. ### Нет, терминология в сильном мышлении не слишком важна. Мышление экономится за счёт тщательно выбранных понятий, которые дают компактное объяснение устройству мира.

+++ Системное мышление экономит вычислительные ресурсы за счёт удержания внимания вычислителя на важных объектах-системах. ### Да, системное мышление заставляет в мире выделять вниманием важные объекты-системы, и дальше вести мышление об этих важных объектах. Это экономит вычислительные

ресурсы агента, освобождая их от менее важных вычислений.

--- Системное мышление — это ментальный метод, чтобы системно мыслить, надо не забыть с собой только голову. ### Нет, системное мышление (как и любое сильное мышление, идущее по методам из интеллект-стека) минимально требует моделер в помощь мозгу, но ещё системное мышление может задействовать и другие доступные вычислительные ресурсы, например, электронные компьютеры для вычислений по моделям самых разных систем.

Состав интеллект-стека

Приведём краткое описание интеллект-стека в обратном порядке трансдисциплин (сами методы называются обычно по названию их дисциплин/теорий, редко какие методы называются по их инструментарию), снизу-вверх, чтобы было понятней, как объяснения/теории одних дисциплин пользуются понятиями, уже введёнными другими дисциплинами:

- **Понятизация** учит выделять фигуры из фона и делать их предметами рассмотрения. Роль агента, занимающегося понятизацией::метод — **поэт**, «дающий имена». Какие-то из этих объектов потом окажутся системами.
- **Собранность** учит удерживать во внимании «объекты», которые уже обсуждены в понятизации и даёт понятие о сознании. Роль тут **«собранный»**, причём в случае человека — киборг, ибо вниманию голого мозга мы не верим, мы поддерживаем его хотя бы ручкой-бумагой, но лучше — компьютером. Не пишешь — не мыслишь.
- **Семантика** учит отделять физические объекты (ими занимается физика) от математических/абстрактных/ментальных/идеальных объектов (ими занимается математика), тем самым разделяя объекты в мире и объекты в их более и менее формальных описаниях. Но эти объекты уже могут быть удержаны во внимании и для их обозначения при мышлении (собранность уже показала, что всё надо записывать) вводятся знаки/символы (ими занимается семиотика, учение о знаках). То есть по факту современная семантика — это семиотика, учение о значении и смысле знаков, только дополненная тем, что отсылает к математике и физике, а не просто сосредотачивается только на знаках, как это делается в семиотике. Роль — **семантик**.
- **Математика** учит тому, какие бывают абстрактные объекты и их отношения, какое поведение абстрактных объектов. Но семантика уже сказала про их существование. Лучшие системные описания, конечно, основаны на математических представлениях. Математика непрерывно развивается, сегодня она предлагает интересные виды математических объектов, которые раньше не использовались. Роль задействующего математику как метод математического мышления — **математик**. Дальше эти новые хорошо изученные ментальные объекты окажутся в своём ментальном поведении похожи на реальное поведение физических объектов, и математические объекты будут использованы физиками.
- **Физика** учит поведению физических объектов, которые представлены математическими объектами. Семантика уже рассказала о том, что физические объекты представляются в мышлении ментальными/математическим объектами. Но в том числе в физике

затрагиваются вопросы физико-математической теории информации: как именно математические объекты представляются в физическом мире («математик и астрофизик — физические объекты»). Именно **в физике впервые вводятся понятия системы и многие другие понятия системного подхода.** Роль — **физик**. Виталий Ванчурин дальше различает философов и физиков: физик даёт такие объяснения (ибо опирается на математические объекты), в которых можно затем что-то измерить и что-то посчитать-предсказать, тем самым проверив. Философ от этого свободен, поэтому физик сочинять ерунду особо не может, а вот философы — нет проблем, их рассуждения могут быть вообще никак не связаны с реальностью, сказочники среди философов могут быть, а среди физиков — нет. Привязка рассуждений к реальности — это для физиков главное.

- **Теория понятий** учит, что все объекты в каком-то смысле подобны друг другу, и это описывается типами/классами или прототипами. И о том, что об объектах мы можем судить по их отношениям друг с другом. Примеры часто встречающихся типов отношений — это классификация, специализация, композиция. Физика (и в ней теория информации) при этом уже сказала, как все эти описания представлены в физическом мире на носителях информации. Обычное мышление людей — это «образное» мышление, основанное на метафорах. По большому счёту, моделирование — это тоже метафора, метафоры нужны и для построения убедительной речи, и для этого нужно понимать, как работать с прототипами для обсуждаемых объектов. Но критика требует представлений об объектах и отношениях, чтобы потом с ними сработала онтология, алгоритмика, логика. Для этого нужно инсталлировать в мозг (или в программу AI) «машинку типов», чтобы она могла выполнять операции присвоения типа. Роль — **типолог**. Конечно, тип — это математический объект, а удерживать внимание на присвоенном типе нужно будет собранностью, для чего всё записывать.
- **Онтология** учит отвечать на вопрос, каким способом мы многоуровнево описываем/моделируем мир: как мы определяем важное и неважное (моделирование как создание «графов знаний», соответствующих понятиям и их отношениям из теории понятий), как мы используем графы знаний для рассуждений и объяснений. Мы разбираемся с многоуровневым мета-моделированием (описания как абстракции получаются не произвольно, но абстрагирование управляется абстракцией более высокого уровня). Формальные выраженные признанными математическими объектами (например, логическими предикатами) онтологии/графы знаний и выраженные текстами на естественном языке менее формальные онтики/frameworks задействуют понятия (используется теория понятий) и выражают свойства физического мира. Роль — **онтолог**.
- **Алгоритмика** — это естественная/экспериментальная наука/science, которая обсуждает способы проведения рассуждений с информационными моделями (то есть способы вычислений), которые нам уже известны из онтологии. Эти рассуждения/вычисления идут с объектами в разных по физической природе универсальных (все они соответствуют машине Тьюринга, это важнейший теоретический результат алгоритмики) вычислителях (мозг, электронный компьютер, квантовый компьютер, оптический компьютер). Роль — **алгоритмист**. Есть универсальные

алгоритмы, которые могут быть обучены бесконечно близко аппроксимировать какие-то прикладные алгоритмы. Это и есть алгоритмы искусственного интеллекта, то есть алгоритмы, занимающиеся мышлением/познанием/learning/cognition!

- **Логика** говорит, какие есть способы рассуждений над моделями, чтобы результаты рассуждений (тоже модели) при правильных посылках и правильных правилах рассуждений как-то соответствовали реальному миру. Онтология для этого уже рассказала про то, как мы нарезали мир на объекты, описав эту нарезку какими-то моделями, так что рассуждения работают с моделями, а «работают» — это идут вычисления, мы об этом знаем из алгоритмики. Роль — **логик**. В принципе, современная «математическая логика» вроде бы часть математики, но мы используем более традиционное понимание логики, в существенной степени пересекающееся с включением вопросов семантики в логику и выходом в прагматизм, «принятию всерьёз» — нам важно, что непротиворечивые рассуждения должны быть положены в основу для действия, а противоречивые должны быть призывом задействовать интеллект и думать дальше, а не действовать, исходя из заведомо нелогичных рассуждений. Поэтому главный вопрос, который мы тут обсуждаем — «а чо такова?», то есть игнорирование ошибок типа «2*2=5». Если знаешь, что в рассуждениях заведомо есть ошибка, такое рассуждение нельзя класть основанием для действия! Нельзя быть нелогичным!
- **Рациональность** как фундаментальный метод говорит о том, что логичные рассуждения по моделям/теориям/объяснениям/знаниям нужны для действий, улучшающих мир. Поэтому нужны рассуждения по связи причин и следствий в конкретной ситуации, а для этого нужно с одной стороны добыть информацию о мире, для чего нужно определить, на что смотреть, потом посмотреть, потом принять решение о действии, в том числе о таком действии, посмотреть ли на что ещё, или деятелю/актёру уже можно принимать решение о действии по изменению мира в условиях неопределённости («на вас напал тигр: собирать дополнительную информацию и наблюдать, бежать или нападать, или есть какие-то другие опции — придумать и реализовать их?! У вас примерно три секунды на все размышления»). Теория решений будет ядром рациональности. Роль — **разум.**
- **Исследования/познание** как отдельный метод. Этот метод по факту — часть методов, опирающихся на «теорию познания»/эпистемологию, рациональность как «научное мышление» — это вторая часть эпистемологии. Эпистемология — это теория не любого метода познания, а такого, результаты которого отчуждаются (письменная культура науки! Изложение словами и формулами на формальных языках!) и коллективно проверяются. Она отличается от «теории познания»/гносеологии, включающей художественное и религиозное иррациональное «познание» мира в форме неотчуждаемых собственных ощущений. «Рациональные исследования»/«научное познание» говорит о том, каким образом рациональные агенты получают полезные теории/дисциплины/объяснения, которые они потом **принимают всерьёз**, то есть начинают действовать по ним в надежде «спастись» от неприятных сюрпризов, которых полна Вселенная. Мы делаем догадки о хорошей объяснительной (причинной)

предсказывающей/порождающей модели/теории, а затем критикуем эту догадку на предмет непротиворечивых результатов рассуждений по этой модели и на предмет лучшего соответствия предсказаний этой модели с результатами эксперимента. Все нужные понятия для описания исследований уже известны из понятизации, собранности, математики, физики, семантики, онтологии, алгоритмики, логики, рациональности. Исследования как дисциплина/знание/теория/модель объясняет, как все они задействуются в ходе бесконечного познания. Роль — **исследователь/учёный**.

- **Эстетика** даёт критерии красоты (в исследованиях принято говорить об **элегантности**) в результатах мышления и прикладного труда. Эстетика даёт объяснения влиянию создаваемого агентами (в том числе художниками, артистами и прочими «людьми творческих профессий», а сейчас и AI) объекта на других агентов, чтобы вызвать у них какие-то изменения их моделей мира и моделей себя, а также уверенности/beliefs в правдоподобности этих моделей. Не факт, что современная эстетика обсуждает, например, эмоциональное воздействие каких-то продуктов труда и описаний только на агентов-людей. Нет, современная эстетика рассматривает и людей, и агентов с искусственным интеллектом, и искусственную жизнь. Роль — **эстет**.
- **Этика** говорит нам о том, чего нужно добиваться в жизни: какие цели приемлемо ставить агенту и какими средствами добиваться реализации этих целей. Должны ли люди умирать, или лучше бы их сделать бессмертными? Нормально, если люди меняют своё мировоззрение и убегают в другие общества, а их исходное общество тем самым умирает? Что лучше: убить и сжечь группу из заражённых смертельным вирусом людей и тем самым спасти человечество, или не убивать, потому как «не убий» это заповедь — и чёрт с ним, с человечеством, оно само как-то выживет? Для этических рассуждений об этом мы уже владеем пониманием, что такое рациональность и как устроены исследования. Современная этика многоуровнева, а для этого мы уже можем в более-менее полной мере задействовать понятия системного подхода из интеллект-стека, чтобы рассуждать об агентах разных системных уровней, конфликтующих между собой в целях «спасения» объектов этих разных системных уровней, возможно, в ущерб «спасению» объектов других уровней. Для одних чёрт с ним с многомиллионной страной, зато все люди живы, а страны могут и исчезнуть, а вот для других — страны должны выжить «любой ценой», даже если в них останется после тотального смертоубийства парочка человек в каждой выжившей из многих миллионов. Роль ищущего многоуровневую оптимизацию выживания объектов разных системных уровней при неминуемом их конфликте — **совесть**. А дальше, когда в агенте «говорит совесть» — это или «этическое кулибинство», когда агент ничего не знает о современной этике как фундаментальном методе мышления (со своей трансдисциплиной), или всё-таки это мышление цивилизованного и образованного агента (человека или компьютера с AI), который что-то знает о современной многоуровневой этике.
- **Риторика** говорит о том, как убедить какого-то человека совершить какие-то действия, или наоборот — убедить его не действовать. Начинаем с того, что вы должны иметь какую-то рациональную модель ситуации (полученную вами в ходе исследований). Далее вы объясняете вашу модель ситуации

какому-то другому агенту, пытаясь его уговорить использовать эту модель для достижения каких-то ваших целей, а преследование его собственных целей пока отложить. Риторика учитывает этику, чтобы сладкими речами не подбивать агентов (животных, людей, роботов) на что-то плохое. Роль — **ритор**. Но ещё вы используете теорию понятий: сладкие речи задействуют метафоры и образы для работы с прототипами. Сладкие убедительные речи обращаются к вашему быстрому мышлению (S1 по Канеману) для интуитивного принятия решений (с возможными ошибками!) и привлечения внимания в коммуникации. А вот передача рациональных моделей/знаний требует, наоборот, отказа от быстрого мышления в пользу медленного мышления со строгими типами объектов и отношений, и дальше решения будут приниматься рациональные, на основе логических рассуждений и теории решений, а не на основе метафор и лежащей за метафорами интуиции. **Ритор** занимается как раз вот этим: переводом содержания между разными способами выражения знания, а также побуждением к действию на основе переданного знания.

- **Методология** рассказывает о **методах/способах/практиках** работы (труде/деятельности/культуре/инженерии), в которых агенты (люди, AI-агенты, предприятия) организовываются в команду (в случае предприятий — это «расширенное предприятие»/extended enterprise), занимают в ней какие-то роли в рамках **разделения труда** по «методам работы»/«видам труда»/практикам/культурам, а затем выполняют коллективные работы по изменению состояний каких-то объектов, удерживая свои роли, то есть каждый агент профессионализируется в каком-то методе работы. Главное, что изменение состояний объектов проходит не просто в ходе работы, но в ходе работы по методам/способам работы. Методы непрерывно эволюционируют, из них выбираются лучшие на сегодняшний момент. Риторика позволяет понять, как агенты договариваются. Роль, занимающегося методологией::метод — **методолог**. Как описать прикладной метод? Методология как раз даёт ответ на этот вопрос. Вы описываете методы создания ракеты или выращивания овса — и дальше уже это будет основой для принятия организационных решений по тому, как организовать такой инженерный проект, изменять и заменять методы, которыми ведутся работы в организации. Методология занимается методами работы, которые называются самыми разными терминами из длинного синонимического ряда: «вид труда»/«метод работы»/«способ работы/way of working»/практика/деятельность/культура/стиль/«вид инженерии» и даже «стратегия» оказывается именно методом работы, которого будет придерживаться какой-то агент в достижении своих целей.

- **Инженерия** (универсальная/трансдисциплинарная/фундаментальная, часто называемая **системная инженерия**) — это самый общий способ/метод создания новых и изменения старых систем так, чтобы мир изменился к лучшему. В фундаментальные методы мышления входит только самая общая инженерия как нормативный метод работы, используемый на всех уровнях организации систем — от неодушевлённых молекул и наночастиц до человечества в целом вместе со всей его материальной культурой, миллионы тонн вещества. А дальше уже в прикладных методах инженерия будет конкретизироваться для систем разных масштабов и разных их видов на каждом масштабе. Для рассуждений об инженерии задействуются все

предыдущие слои интеллект-стека, особенно если учесть, что в качестве агентов действуют люди, люди и компьютеры, иногда люди с другими живыми существами (например, слепые с собаками-поводырями), и даже уже сами компьютеры без людей (смотри материалы по современной робототехнике). Роль выполняющего методы инженерии — **инженер**.

Каждый метод мышления интеллект-стека, основанный на фундаментальной/безмасштабной дисциплине/трансдисциплине/теории/знании помогает разобраться со следующим методом/культурой/способом/практикой мышления в стеке. Хотя это утверждение про стек методов мышления довольно условно: все эти фундаментальные методы мышления тесно переплетены в своих объяснениях/теории/знаниях друг с другом, да и инструментарий эти методы используют одни и те же (моделеры): что там что поддерживает не очень понятно.

Мы выбрали такой состав методов интеллект-стека и порядок методов в нём главным образом в методических целях: для облегчения понимания в ходе обучения сильному мышлению. Конечно, в ходе эволюции знаний этот набор методов мышления будет существенным образом меняться, а также будет предложено множество альтернативных вариантов интеллект-стека, которые будут конкурировать между собой. Тем не менее, наши руководства будут опираться на такой состав фундаментальных методов мышления и такой порядок этих методов в интеллект-стеке.

Интеллект-стек и системное мышление

На вершине стека фундаментальных методов мышления происходит тот самый «транс» переход: интеллект-стек помогает в мышлении и действии прикладным инженерным методам изменения мира.

Системное мышление можно понимать как мыслительные приёмы нескольких методов из состава интеллект-стека как набора фундаментальных дисциплин/теорий/объяснений/знаний, поддержанных инструментарием, прежде всего инструментарием моделирования. Эти мыслительные приёмы системного мышления задействуют понятия «система», «системный уровень», «эмерджентность», «неустроенность» и другие понятия **системного подхода**.

Можно было бы сказать не только «системное мышление», но и «онтологическое системное мышление», ибо системное мышление основано на трансдисциплине онтологии, и «методологическое системное мышление», ибо существенно задействуются положения трансдисциплины методологии, и «трудовое системное мышление», и «деятельностное системное мышление» и «инженерное системное мышление», «рациональное системное мышление», и так далее.

Системная инженерия — это инженерия, описания методов которой основаны на системном мышлении. Но можно было бы и просто сказать «инженерия», системности современного инженерного мышления это бы не убавило. Можно было бы добавить и в это название онтологию: «онтологическая системная инженерия», так иногда и говорят — ontology based systems engineering[31]. Можно добавить и этику, «этическая системная инженерия». Можно рациональность, «рациональная системная инженерия».

---

[31] https://www.sciencedirect.com/science/article/abs/pii/S0166361518307887

Системное мышление как часть мышления интеллекта в целом (мышления по интеллект-стеку во полноте его методов) не выключается в ходе работ прикладного метода/«вида труда»/деятельности/инженерии/практики/стратегии. Системное мышление — часть фундаментального мышления, которое будет задействовано всегда, когда в жизни будет встречаться что-то, что не описано в учебнике/регламенте/инструкции/стандарте прикладного метода и его дисциплины/теории/модели/знаний/объяснений, например, в учебнике менеджмента, учебнике медицины, учебнике правоприменения.

Само сопоставление содержания учебника (регламента, стандарта) по прикладному методу мышления с жизнью уже требует задействования фундаментального мышления. Скажем, в регламенте («учебнике для сотрудников») будет сказано «несовершеннолетним клиентам кредиты не выдаются». Чтобы разобраться с этой фразой, требуется крепко поразмыслить:

- Что такое «несовершеннолетний», если пришёл иностранец и в его стране принят другой возраст совершеннолетия — вы ущемляете его права?
- Что такое «клиент», если человек в середине оформления своего статуса клиента?
- Что такое «кредит» из десятка очень похоже выглядящих банковских продуктов, и часть из них «займы»?
- Что такое «выдача кредита» и в какой момент можно считать, что она произошла или не произошла?
- Это всё было переводом начальной фразы с тамильского: как убедиться, что перевод был точным?

А теперь попробуйте это объяснить даже не себе, а тупому компьютеру, которому нужно реализовать эту организационную норму/business rule[32]. Поручить программисту это объяснение не удастся, ибо это всё не вопросы информационных технологий, это вопросы не его специальности «программная инженерия»/software engineering.

И онтологическое мышление никуда не девается, когда идёт системное мышление, в самом системном мышлении есть и куски онтологии. Скажем, каким образом мы считаем, что по-разному нарезанный на части объект — это и впрямь один и тот же объект — это традиционный вопрос онтологии. Скажем, ножницы из двух половинок и винтика на заводе и ножницы из ножевого блока и ручки в ходе их эксплуатации — это одни и те же ножницы, ибо эти два разных объекта занимают одно и то же место в пространстве-времени. Это мы явно упомянули принцип 4D экстенсионализма, вы должны его знать из руководства по рациональной работе. Напомним, что для освоения системного мышления качественное освоение рациональной работы — это обязательный пререквизит.

Собранность как управление вниманием не выключается, когда идёт онтологическое мышление, вы ведь всё записываете, а не просто «ментально собраны, всё запоминаете»! Эти все мышления (интеллект-стека и даже прикладные мышления прикладных видов мастерства) идут практически одновременно в тесном переплетении, и только наше (часто поддержанное компьютерами и хорошо натренированное) внимание может выделить в этом связном и непрерывном процессе мышления какого-то интеллекта те или иные моменты, связанные с разными частными видами мышления, проводимыми

[32] https://ailev.livejournal.com/693597.html

частными видами мастерства.

Этот одновременный учёт самых разных методов мышления как рассуждений с самыми разными объектами внимания довольно легко понять: когда вы смотрите на дерево, раскачивающееся под ветром, обсуждаете углы раскачки в зависимости от силы ветра, но это абсолютно не исключает того, что в этом дереве прямо в этот же момент по ходу рассуждения продолжает идти фотосинтез, в дупле этого дерева высиживает птенцов неведомая птичка. Нет, это всё присутствует в реальном мире. Просто когда ваше внимание сосредоточено на образе раскачивающегося под ветром дерева в целом, то можно подробно рассмотреть именно объект «раскачивающееся дерево в целом», но объекты других размеров, входящие в физический объект «раскачивающееся дерево в целом» и находящиеся в окружении этого физического объекта, не будут рассмотрены. Но они из жизни никуда не исчезают, просто будут временно (а иногда и навсегда) от вас скрыты. Временно скрыты, пока вы не обратите внимание именно на них. Навсегда скрыты, если ваше внимание так к ним и не обратится.

Но вот вы обратили внимание на фотосинтез, при этом и раскачивание дерева сильным ветром, и кто там у этого дерева в дупле — всё это вмиг потеряло значение, внимание переключено на фотосинтез и дальше мы удерживаем внимание на фотосинтезе, задействуем собранность. Но «исчезновение объектов в рассмотрении» исключительно работа нашего внимания, с деревом ничего не происходит! В жизни дерево существует во всей его полноте, никуда не девается ни фотосинтез, ни птичка, ни раскачивание под ветром. Просто нам удобно рассматривать и обсуждать всё это по очереди, а не всё сразу. И мы при этом деятельны: принимаем решение вообще рассмотреть это дерево (а не велосипед), может быть даже подойти к дереву, или просто воспроизвести его модель прямо «в голове», и ничего телом в этот момент не делать (не забывая при этом, что «думать/мыслить/рассуждать/вычислять» — это тоже работа тела, хоть человеческого, хоть робота).

Вот это рассуждение про выделение вниманием разных **целых объектов одной камерой внимания, а другими камерами внимания отслеживание частей этих целых объектов** для удобства рассмотрения сложных ситуаций, а потом надёжное удержание некоторое время во внимании именно этих **целых объектов и их частей** без блуждания камер внимания по другим объектам (использование собранности для работы с частями и целыми для этих частей) — оно типично для системного мышления, это самая суть подхода, для этого и было разработано системное мышление, хотя вы смело можете считать, что это некий синтез из понятизации (выделение объектов, их именование, чтобы потом различать и говорить о них) и собранности (удержание во внимании), физики (выделение частей в их изоляции от окружающего мира), теории понятий (отношение часть-целое), онтологии (понимание иерархии в отношениях часть-целое как основы для выделения системных уровней).

Вы выбираете правильный для решения ваших задач **уровень рассмотрения частей-целых (масштаб крупности в длинах, масштаб времени в продолжительностях, «уровень организации», «эволюционный уровень», «техно-эволюционный уровень», «системный уровень» — для живых систем это, если грубо, молекулы, клетки, органы, организмы, популяции и примерно так же «по размерам» это будет нарезано вниманием для техно-систем).** Дальше вы решаете на выбранном уровне

рассмотрения частей-целых свои проблемы, направляя внимание выше или ниже по этим уровням в зависимости от того, над чем размышляете.

Если что-то обсуждено недостаточно подробно, то можно всегда вернуться и дообсудить (а для того, чтобы не терять внимания, всё записываем, а не размышляем только «внутри головы»). **Обсуждаем части, не теряя из виду целого. Обсуждаем целое, не забывая о частях.** Обсуждение задействует наше внимание к какому-то уровню частей целого объекта, и дальше уже частей этих частей. Сам объект остаётся в его натуральной целостности, а части и целые объекты на многих уровнях выделяются только вниманием. Внимание может работать и в обратном направлении: для каких-то частей целого объекта оно может быть направлено дальше «вверх» по системным уровням, то есть на целый объект. А затем и этот целый объект может быть объявлен частью, а внимание найдёт объемлющее целое ещё более высокого системного уровня.

Интеллект-стек как набор методов мышления нельзя считать классической системой из физики (методы — это поведение, системой можно было бы пробовать назвать сам мыслящий объект, то есть интеллект, хотя тут тоже есть нюансы), но работа внимания к отдельным методам мышления, а в самом интеллект-стеке внимание к понятиям отдельных трансдисциплин и поддерживающих их отдельным инструментам поддержки рассуждений по мыслительным приёмам этих трансдисциплин (главным образом моделеров) устроены похоже: из плотного неразрывного клубка понятий и их отношений мы вниманием выделяем, а затем удерживаем во внимании и подробно обсуждаем и отдельные методы мышления, а в них дисциплины и инструментарий, и дробим дальше вниманием методы на отдельные приёмы мышления, и даже отдельные составляющие этих приёмов (при всей неопределённости разговора про «части операций/приёмов/процедур», хотя мы дальше в нашем руководстве расскажем, как пробовать разобраться в поведении/activity/behavior так, чтобы точнее проводить границы между разными поведениями).

Мышление неразрывно в его фундаментальных и прикладных методах, оно происходит в целом и прикладное, и общетрудовое/инженерное, и интуитивно-понятийное (понятизация занята этим), и алгоритмическое, и онтологическое, и методологическое, в нём присутствуюn все прикладные методы мышления и интеллект-стек.

Если мы размышляем о самом мышлении (интеллект-стек как раз для этого!), то мы выделяем силой нашего внимания в методах мышления какие-то части и какую-то часть времени думаем только о них — чтобы преодолеть сложность мышления, чтобы лучше понять, как устроено мышление, как ему научить. **Какие-то выделяемые вниманием составляющие полного мышления по знаниям самых разных методов мышления интеллект-стека, имеющие дело с понятиями «система», «системный уровень», «эмерджентность» и другими понятиями системного подхода — это и будет системное мышление. Для наставления в этих составляющих полного фундаментального мышления всего интеллект-стека и было создано руководство по системному мышлению, которое вы сейчас проходите.**

Аналогично тому как рациональное, семантическое, логическое и т.д. мышления используют в своём составе приёмы системного мышления на базе понятий системного подхода, так и рациональное, семантическое, логическое,

онтологическое и многие другие фундаментальные методы мышления задействованы в системном мышлении: понятия теорий/дисциплин фундаментальных методов мышления сильно переплетены друг с другом. Без освоения всего интеллект-стека хорошо системно мыслить не станешь.

Наставлений только руководства по системному мышлению для того, чтобы системно мыслить, не хватит. Руководство по системному мышлению даёт только один срез, одну выборку из всех знаний, нужных для полноценного усиления интеллекта. Студенты вузов получают двойки по курсу системного мышления на основе нашего руководства часто не из-за незнания понятий системного подхода и неумения их использовать в мышлении о своих рабочих проектах, а из-за плохой подготовки в области семантики, теории понятий, онтологии, логики[33].

Например, рассмотрим такую дисциплину/теорию как семантика: типичная ошибка при её незнании — это неразличение понятия и термина для этого понятия. Стоит расслабиться — и в системном мышлении уже перепутано понятие «потребности» как психологической потребности пить-кушать-размножаться и «потребности» как интереса внешних проектных ролей к своим системам (надсистемам для целевой или системам создания в их графах). Слово-термин одно, словарные гнёзда разные, понятия разные — но это часто не отслеживается, термины и понятия не различаются, ибо знаний по семантике нет. В руководстве по системному мышлению нет возможности подробно наставлять по семантике, тут уже пользуются этими знаниями, полученными из руководств-пререквизитов (руководство по рациональной работе)! Хоть какое-то владение семантическим мышлением нужно пройти до освоения системного мышления!

Руководство по рациональной работе устроено примерно так же, как руководство по системному мышлению: взяты какие-то знания/теории/правила/алгоритмы самых разных методов мышления интеллект-стека, но акценты проставлены другие: на собранности, семантике, теории понятий, онтологии, логике. И ещё добавлено немного прикладного знания по операционному менеджменту. Понятия системного мышления в руководстве по рациональной работе затрагиваются, но на них нет акцента, нет подробности изложения. Поэтому только руководства рациональной работы тоже не хватит, чтобы существенно усилить свой интеллект, «стать другим человеком», равно как и только руководства по системному мышлению или только руководства по методологии. Конечно, вы получите существенный результат после освоения работы по каждому из этих руководств! Это не так, чтобы вы целый год занимались освоением всех руководств программы рабочего развития, и потом все результаты появились скачком по итогам года! Конечно, нет! Результаты будут заметны сразу же по освоении методов мышления и действия по каждому из этих руководств. Но после освоения всей программы вы вдруг осознаете, что «стал другим человеком» (это нам сообщали многие инженеры-менеджеры, которые прошли этот путь). Для получения такого результата на однократное прохождение стажировки по программе рабочего развития надо затратить порядка 480 часов, это примерно год на пятидневке по 2 часа стажировки в день, желательно в рабочее время (ибо это мышление согласно нашим руководствам про ваши рабочие проекты, а не личные или учебные проекты).

Ещё один пример: неумение агента работать с типами в мышлении. Способы этой

[33] https://ailev.livejournal.com/1465753.html

работы с типами описываются знаниями из теории понятий, одной из трансдисциплин методов в составе интеллект-стека. Если вы пишете, что «система X — это система отношений между покупателями и продавцами», то нельзя через три строчки писать, что «система X — это софтверная платформа», а ещё через три строчки писать, что «система X — это проект по предоставлению сервиса». Это ошибка того же вида, что написать «X — это огурец», через три строчки написать «X — это пушной зверёк», а ещё через три строчки написать «X — это система ценностей». Тип ошибки тут один и тот же, но для огурца и пушного зверька здравого смысла хватает заметить ошибку, а вот что «система отношений» это ни разу не «софтверная платформа», а «платформа» — это не «проект» — вот это уже для плохо тренированного в работе с типами человека не берущаяся задача. Если такое видишь в тексте, то чётко понимаешь, что понятия целевой системы как физического объекта в голове автора текста нет, а есть эдакое «облачко смыслов», не доведённое в мыслях до какой-то физической реализации.

Работа с типами касается ещё и возможности провести рассуждение об объектах из учебника на объектах из жизни. Если ты понимаешь, что у тебя тип «физическое тело» в учебнике физики летит по параболе, если его кинуть при наличии гравитации, то это рассуждение о типах. В жизни надо ещё будет сообразить, что кинутый мячик — это то самое физическое тело из учебника! Далее надо перейти к материалу учебника физики и взять оттуда, что полёт будет по параболе. Потом вернуться к мячику и выдать суждение, что «поскольку мячик — физическое тело, то он будет лететь по параболе», и даже прикинуть, какое место абстрактных объектов «парабола» и «траектория» вокруг мячика в реальном мире. Это рассуждение абсолютно такое же, как про какие-нибудь целевые системы. Если вы производите «системы бухгалтерского учёта», то надо ещё сообразить, что слово «система» тут — тип из теории/знаний системного подхода, обратиться к материалам нашего руководства, затем найти эту систему в окружающем мире (что это за «система»? Это тип для набора правил учёта? Но вы их явно не производите, и они не похожи на системы из руководства. Более того, в руководстве у нас явно сказано, что наборы правил и системы уравнений — это не те «системы», которые в системном подходе. Далее вопрос: это софт? Это отдел «бухгалтерия», который использует софт бухучёта? Это отдел, который использует софт и все те люди, которые вынуждены поставлять данные для отдела, который кормит этими данными софт? В какой момент времени можно сказать, что «мы выпустили систему бухгалтерского учёта»? Почему разные люди в предприятии не договорились, что же мы выпускаем?). **Это всё про операции с типами — если с этим не разобрались, то невозможно найти важные объекты внимания в жизни, о которых много чего будет известно сразу, как только вы отождествите их с объектами из нашего руководства. Руководство не даёт наставлений о конкретных системах в конкретных проектах. Но руководство даёт наставления про все системы во всех проектах. Надо только уметь присвоить типы из руководства объектам из реальной жизни, из вашего реального рабочего проекта.** Скажем, ваша организация выпускает сепульки, а вы делаете для сепульки винтики. Надо уметь присвоить тип «целевая система» сепульке, выявив именно сепульку как целевую систему, а винтику надо присвоить типы «наша система» и «подсистема целевой системы» (да, сразу два типа).

Особо тут можно упомянуть плохое обращение с отглагольными существительными (мышление тут тоже может выступать примером). Мышление один раз может быть процессом (и в нашем руководстве именно так), который реализуется

вычислителем-интеллектом/«мыслительным мастерством», а второй раз у тех же людей иногда и в той же фразе — это часть мозга, синоним интеллекта, а не поведение этой части (в нашем руководстве это не так)! И то, что мышление как «глагол» и мышление как «существительное» могут путаться — это люди с плохим тренингом в теории понятий не замечают. У них будет всё очень, очень плохо с концепцией системы, ибо функции в концепции системы задаются как раз отглагольными существительными, и это поведения, а не вещи, и путаться в этом нельзя. Это всё вопросы, которые не затрагиваются руководством по системному мышлению, но рассматриваются в других руководствах программы рабочего развития, прежде всего руководством по рациональной работе.

Сюда же можно отнести суперобобщения/overgeneralizations, это тоже к работе с типами как онтологической уже работе. Вместо какого-то объекта очень плохо указывать его супер-супер-супер-тип/класс, очень высоко стоящий в классификаторе, и считать, что дальше всё берётся операцией наследования свойств типа. Например, вместо «тигра» везде говорить про «зверя» — а потом удивляться, почему другие люди подставляют в разговоре вместо зверя свою «мышь» (это же тоже зверь!), после чего фраза «мышь опасна для человека» для них является неожиданной и им невдомёк, откуда она берётся. А появляется такая фраза из-за того, что «тигра» нужно называть тигром, а не более общим классом «зверь». Не нужно обобщать чрезмерно, в лишних обобщениях часты ошибки! Если вы указали дрель как «оборудование» и дали характеристику «частота вращения», то это кажется нормальным. Но потом кто-то добавит такое «оборудование», как люк, и у люка появится характеристика «частота вращения»! Это ведь простое логическое следствие того, что «люк — это оборудование». Ибо вы думали о «дрели», а писали «оборудование»!

Ещё одно логическое препятствие для системного мышления как использования понятий системного подхода — это проблемы в отслеживании отношений «часть-целое» на нескольких уровнях. Эти уровни по отношению «часть-целое» называются системными, но иногда их же могут назвать эволюционными, структурными, сложности, или даже организационными уровнями. Отношения часть-целое (композиции, речь идёт о физических частях) как между мной-телом в целом и моей рукой как частью тела вдруг заменяется отношением классификации (моя рука — одна из четырёх конечностей человеческого тела, конечность тут — это уже класс! Рука классифицируется как конечность, она не часть конечности!), после чего палец на руке оказывается пальцем на какой-то конечности, причём уже не обязательно именно на руке — переход от «руки» к «конечности» изменил ситуацию! Система в результате представляется как состоящая из не-пойми-чего, а не физических частей.

Недоучившиеся инженеры-менеджеры легко складывают колбасу в штуках с яблоками в тоннах, не считая это ошибкой. Да, это не ошибки в системном мышлении, это ошибки в онтологии и логике как определении того, с какими объектами ведётся рассуждение и по каким правилам оно ведётся, системы ли эти объекты, или какие другие. Без онтологичности и логичности никакого системного мышления не будет. Если в решении дифференциального уравнения вы в арифметике посчитали 2*2=5, то ответ для всего решения уравнения будет неправильный, даже если нет ошибок в высшей математике в части дифференцирования!

Как семантика, теория понятий, онтология и логика лежат в основе системного

мышления и поддерживают его, так и само системное мышление лежит в основе трудового/практического/деятельного/инженерного мышления и прикладных инженерных методов, например, мышление для классической «железной» инженерии, программной инженерии, агропромышленного производства, включая генную инженерию, образование и коучинг как «инженерию личности», менеджмент как инженерию предприятия и так далее, вплоть до общественной деятельности как прикладной культуры/практики/метода изменения общества.

Менеджер (инженер организации) без системного мышления — это плохой менеджер. Быстро меняющиеся прикладные деятельности все основаны на крепких навыках мышления по интеллект-стеку фундаментальных методов мышления: инженерии, методологии, риторике, этике и так далее до собранности и понятизации. Под работой каждым методом есть какие-то умения, какое-то мастерство. Так, собранность — мастерство/умение обратить на что-то внимание и удерживать это внимание, в том числе осознанность в том, на что именно обращено внимание и насколько хорошо оно удерживается — например, понимаете ли вы, сколько минут подряд вы читаете наставления нашего руководства, а сколько минут подряд вы смотрите ленту в соцсетях. Но это самое простое понимание собранности. Более уместное: умеете ли вы выявить и удержать тип объекта в ходе рассуждения, достаточно ли для этого собранны? А если это рассуждение происходит в форме чата с коллегами и идёт пару недель? Понятизация — это мастерство/умение облачить неясные ощущения о понятиях в словесную форму. Семантика — это мастерство/умение сопоставить знаки, объекты, понятия.

В сильном мышлении задействован весь интеллект-стек, а не только какая-то его часть. Это означает, что в сильном/универсальном/общем мышлении задействованы методы мышления всего интеллект-стека, а не только мыслительные приёмы, набранные из теории понятий, онтологии, логики, рациональности. В сильном мышлении задействованы не только мыслительные приёмы, связанные с системным подходом и известные вместе как «системное мышление». При этом интеллект-стек быстро эволюционирует. Например, существенно изменилась часть методов, связанных с семантикой: в связи с быстрым развитием современных систем AI появилось обучение представлениям (representations learning, главным образом векторным/распределённым/distributed[34], отвязанным/disentangled[35], непрерывным/continuous[36]) как альтернатива семиотике (изучающей знаки как локальные/атомистические/дискретные представления).

Моделирование: понятия.

Заполните таблицу, основываясь на наставлениях нашего руководства:

Термин	Английский вариант	Синонимы (если есть)
Система		
Граница (системы)		
Окружение		
Подсистема		
Метод		
Трансдисциплина		

[34] https://deepai.org/machine-learning-glossary-and-terms/distributed-representation
[35] https://deepai.org/machine-learning-glossary-and-terms/disentangled-representation-learning
[36] https://www.sciencedirect.com/science/article/pii/S2666651021000206

Поведение		
Системный подход		
Системное мышление		
Стек		
Безмасштабный		
Проактивный		
Лучшее в мире (на сейчас)		
Познание		

Мыслительный минимум современного человека: учим один раз, используем во всех проектах

Освоение мыслительных методов наверху интеллект-стека обычно требует определённого уровня владения теми методами, которые находятся ниже. Если вы не знаете, что такое теория понятий, не знакомы с логикой, с этикой, то изучение риторики вам не будет доступно. Напомним последовательность слоёв сегодняшнего интеллект-стека (набор методов может меняться! Развитие не стоит на месте! Более того, могут быть и альтернативные варианты интеллект-стека уже сегодня, эволюция предполагает существование множества видов, в том числе множество видов теорий/объяснений/дисциплин):

- Системная инженерия
- Методология
- Риторика
- Этика
- Эстетика
- Исследования/познание
- Рациональность
- Логика
- Алгоритмика
- Онтология
- Теория понятий
- Физика
- Математика
- Семантика
- Собранность
- Понятизация

Едва ползающему человеку прыжки и танцы не будут доступны: нужно сначала накачать мышцы (инструментарий), а мозгу освоить приёмы управления мышцами (дисциплина). Нужна подготовка к действию, освоение метода, получение мастерства. Сразу действие не получится. Только после получения готовности тела к действию можно учить какие-то паттерны сложных спортивных и танцевальных движений.

Образование в частности и обучение в целом (**образование** — это специализация обучения, направленная на усиление фундаментального/универсального/общего интеллекта, то есть обучение мышлению по интеллект-стеку фундаментальных методов мышления) устроено так же: предусматривается некоторая последовательность обучения (curriculum learning), знания прирастают по кусочкам,

а не «одним большим куском за раз».

Арифметика изучается перед интегралами, без знания таблицы умножения высшей математики не освоишь — арифметика тут пререквизит для высшей математики. Сначала готовность и автоматизмы/беглость в мышлении для более базовых мыслительных навыков, а затем готовность и автоматизмы/беглость на более прикладных уровнях мышления — и так на нескольких уровнях.

Сначала нужно уметь обращать мысли в слова хоть как-то (понятизация), потом удерживать внимание на мыслях (собранность), потом различать сами мысли, объекты мира и слова о них (семантика), потом разобраться с физикой и математикой — как они отличаются и почему без них нельзя, потом разбираться с типами объектов и отношений (теория понятий), удерживать многоуровневое задание объекта при выделении его вниманием из фона (онтология), и так далее.

Конечно, возможно огромное число образовательных маршрутов (учебных программ/куррикулумов/curriculums), в ходе которых будет усилен интеллект, то есть нужна какая-то последовательность освоения мыслительных приёмов интеллект-стека. Так, для освоения системного мышления, нужно быть хоть как-то знакомыми не с одним, а со всеми методами мышления интеллект-стека. Например,

- В понятизации вы узнаете, что есть объекты, которым мы можем давать имена,
- В собранности вы научитесь записывать результаты ваших рассуждений,
- В теории понятий вы узнаете, что мир надо моделировать через типизированные объекты и их отношения,
- В физике будет дан тип «система»,
- В онтологии иерархия по отношению композиции даст системные уровни,
- в методологии рассказано о том, что одна система создаёт другую систему каким-то методом/трудом/практикой/культурой/деятельностью/стилем,
- в инженерии будет понятно, какими способами мы создаём эти системы.

Это мы ещё раз повторили маленький кусочек того, как из знаний/теорий/трансдисциплин самых разных методов мышления, входящих в интеллект-стек, получается знание системного мышления.

Системное мышление дано «в типах» высокого уровня абстракции, оно безмасштабно, универсально в плане прикладных предметных областей и не антропоцентрично. Поэтому системное мышление применимо для проектов по созданию и развитию таких разных систем, как деталь ракетоплана, авиалайнер, породистая овца и гектар леса, шеф-повар и робототехник, фирма по производству подгузников и фирма интернет-провайдер, сообщество любителей Толкиена и «незримый колледж» в науке. Иногда создаётся и общество как «отдельная страна» или же это общество модифицируется, не меняя страны, но становясь совсем другим («революция»), а иногда создаётся супер-интеллект как гибридный из компьютерных интеллектов и интеллектов людей.

В самых разных проектах одновременно происходят работы самых разных людей по самым разным методам/практикам с их самыми разными дисциплинами/теориями/знаниями/алгоритмами, разными инструментами, разными материалами для этих работ. Вам требуется немного думать о многих из них, чтобы лучше разобраться в собственном проекте: каждый проект и сам включает в себя много прикладных работ по прикладным методам, а при получении проблем ещё и

требует работы методов мышления интеллекта, но ещё не бывает «сферических проектов в вакууме», проект встречается с огромным количеством методов работы других проектов. Каждый приходящий со стороны в проект занимается чем-то своим: внешние контрагенты, поставщики материалов, покупатели, менеджеры, посредники, организации стандартизации, каждый день в проекте появляются новые люди, и со всеми ними надо содержательно разговаривать. Надо уметь как-то разобраться, что они все делают, чтобы как-то выстроить с ними содержательный разговор. **Мышление и деятельность во всём этом разнообразии деятельностей самого проекта и окружающих его проектов устроены примерно одинаково, и можно это компактное мышление выучить один раз, а потом применять в разных работах одного проекта, или даже в разных проектах. Усиление интеллекта, в том числе овладение системным мышлением посвящены как раз этой компактификации мышления: один раз учишься думать, затем применяешь во всех проектах. Включаешь сильное мышление один раз, а потом просто никогда не выключаешь.**

Есть легенда, что талант к мышлению (какого бы вида оно ни было) врождённый. Да, генетическая предрасположенность к какому-то виду мышления бывает, как у спортсменов к какому-то виду спорта. Но там не так всё однозначно: связанных с интеллектом генов сотни, за счёт генетики далеко в мышлении не убежишь.

Так что мы рекомендуем полагаться не на генетику, а на обучение мышлению: сами приёмы мышления не заложены в мозге, они должны быть усвоены и натренированы. Это означает, что натренированный «не талант» легко обойдёт в том или ином виде мышления нетренированного «самородка», который так и останется «вечно подающим надежды», он просто не будет знать, как мыслить правильно. Выученный волками потенциально гениальный Маугли не будет уметь даже разговаривать, до сильного мышления дело даже не дойдёт. Врождённый IQ не имеет особо большого значения (уже приводили примеры), хорошее образование (то есть обучение интеллект-стеку) в жизни значит много больше!

Интеллект-стек — это набор **лучших на сегодняшний момент в нашей цивилизации** фундаментальных/трансдисциплинарных методов мышления, основанных на лучших объяснительных теориях. Лучших (state-of-the-art) в цивилизации по состоянию на нынешний год, а не какой-нибудь 2011 (новая весна искусственного интеллекта с использованием глубокого обучения на нейросетях началась в 2012 году, в 2011 году компьютеры ещё не разговаривали и не могли хорошо видеть!) или уж совсем древний 1980 год (год появления первого персонального компьютера IBM PC).

Эти решения о выборе тех или иных приёмов мышления изо всего известного человечеству множества вариантов как раз и направлены на то, чтобы думать абстрактно, адекватно, осознанно, рационально, системно, практично/проактивно/деятельно, а не «дикарски», с игнорированием всего накопленного цивилизацией мыслительного опыта. И эти решения по выбору безмасштабных (универсальных для разных размеров систем и разных масштабов времени существования систем) и неантропоцентричных (неспецифических именно для человека) приёмов мышления предполагают письменное оформление используемых моделей мышления, начальных данных, промежуточных и конечных результатов мышления, да ещё и выход в реальный мир. Условно, компьютеры тут тоже «письменно». А «выход в реальный мир» — это выход в изменение

физического мира, выход в действия: от «подойти посмотреть» до «пойти поговорить» и «изменить, чтобы не мешало».

Решения по выбору методов/приёмов мышления делаются отнюдь не только приёмами мышления «внутри головы», чисто информационной/вычислительной работой без тела. Это вполне себе проактивный и деятельный метод мыслительной работы, выходящий в мир (в чужие головы, в чужие датацентры) и изменяющий затем к лучшему как мир, так и самого принимающего решения мыслящего агента.

Насколько окультуренный цивилизацией интеллект, то есть проводящий мышление с использованием современного интеллект-стека, сдерживает или наоборот, стимулирует творчество по сравнению с живым «дикарским» мышлением? Опыт цивилизации показывает, что образованные и мыслительно тренированные люди обычно выигрывают в массе своей у неучей, несмотря на их якобы «шаблонное мышление». Гениальные самоучки-дикари-кулибины чрезвычайно редки. При этом на поверку «самоучки-дикари» оказываются часто более чем начитаны и образованы, разве что их образование не было связано с каким-то официальным учебным заведением, а паттерны своего «гениального самородного мышления» они тоже брали из литературы и подхватывали у своих вполне образованных учителей, а не изобретали по ходу дела. Это народная легенда, что самоучки нигде не учились. Они очень даже учились, только сами, а не «официально».

Интеллект в порядке самообразования нужно «накачать» и «разработать» так же, как мышцы и суставы для готовности тела к движению — мозг ведь тоже тренируем, он пластичен, то есть физически изменяется в ходе тренировки! И именно поэтому тренировки мышления не быстры. Как и с обычными мышцами, быстрых результатов за одну-две тренировки мышления не получишь, нужны месяцы и годы, ибо при этом задействуются медленные биологические процессы в мозге.

В ходе человеческого мышления с использованием фундаментальных знаний отращиваются синапсы нейронов, улучшается кровоснабжение мозга. Интеллект как физически реализованный на мозге вычислитель/«машина рассуждений» для методов мышления развивается медленно, это месяцы и годы. Хорошим сравнением тут будет спорт: за три месяца тренировок чемпионом не станешь, а вот за три года — чемпионом ещё нет, но от окружения «людей с улицы» будешь отличаться драматически.

Но есть и трюк: человек не полагается только на биологическую природу своего мышления, а задействует и компьютер. Даже если речь идёт не о полноценном компьютере, а просто о бумаге и ручке, то биологическому мозгу становится проще управлять вниманием, проще задействовать большую память, проще обмениваться результатами мышления. А ещё человеческий интеллект задействует тело, и речь идёт тут не только о том, что при письме шевелятся пальцы рук, а при чтении работают глазные мышцы. В руководстве по рациональной работе, где даются и основы понятизации, довольно много рассказывается о связи собранности ума и тела (например, онтологический дребезг можно распознавать по ощущениям в теле).

Мышление проактивно, оно выходит в физический мир, тело в нём тоже имеет значение, включая продолженность тела в форме инструментов — телескопов, микроскопов, автомобилей, ракет, человекоподобных роботов.

Человеческое мышление имеет внешний характер, оно проходит не только в мозгу,

но и во всём теле, и выходит за его пределы (тезис 4E[37]), поэтому системное мышление тренируется не только как умственное упражнение, но и с задействованием компьютерных средств моделирования — от просто письма в редакторе текстов или редакторе сложных табличек (вроде notion.so или coda.io) до изощрённого математического многомасштабного (multiscale, на разных уровнях структуры/организации моделируемой системы как физического объекта, с разными видами моделей для разных масштабов) имитационного моделирования.

Мы подчёркиваем, что нельзя говорить только о фундаментальных дисциплинах, то есть о чистой теории/знании/объяснениях. Надо говорить о методах/способах работы, где дисциплина/знания/алгоритмы поддержана инструментами, усиливающими мастерство «голого мыслящего мозга» (или «голого мыслящего компьютера», если речь идёт об AI). Люди (а сейчас уже и компьютерные нейросети, и роботы) давно не работают «просто руками», они используют инструменты. В мышлении люди давно не думают «просто мозгами», они используют компьютеры (а компьютеры используют в помощь себе другие компьютеры, или даже людей, или даже давно умерших людей — например, читают результаты мышления в книгах давно уже умерших авторов).

Если вы обнаружили себя в ходе глубокого размышления, в котором вы не ведёте никаких записей, не делаете никаких компьютерных моделей — вы явно что-то делаете не так. Мышление происходит сегодня письмом и моделированием, на чисто человеческую биологическую память одного человека надежды нет. Системное мышление тут не исключение.

Продолжительное фундаментальное образование нужно, чтобы не просто цивилизованно мыслить, но и мыслить бегло, причём с использованием инструментов для мышления (собеседников типа AI и других умных людей, записей в моделерах, заметок на естественных языках). Тут как в спорте: за три месяца чемпионом не станешь ни в одном виде спорта, но за три года вы будете драматически отличаться от «людей с улицы». А за десять лет можно уже думать и о чемпионстве: большинство ваших потенциальных конкурентов тут просто не дойдут до сравнения вашего и их мастерства, если вы тренируетесь/обучаетесь все десять лет. И обучение-развлечение деток в спортшколах сильно отличается от жёсткого обучения олимпийского резерва. В образовании всё то же самое: если у вас обучение-развлечение на три месяца — ваш интеллект вряд ли будет сильным. Если впахиваете много лет, то драматически будете отличаться от всех людей «с улицы», а если речь идёт о десятке лет — можно поговорить о том, что вы будете общаться с умнейшими людьми планеты, и не только вам будет с ними интересно, но им с вами будет интересно!

Натренированные до беглого применения паттерны мышления дают возможность как по проложенным в мозгу рельсам быстро проводить типовые абстрактные, рациональные, адекватные, осознанные, системные, практичные рассуждения, не затрачивая на это мыслительных усилий, то есть интуитивно, «на автомате» — включая рассудочное «пошаговое» мышление по образцам «из учебника», которое не кажется после тренировки чем-то запредельно трудным.

Только если эти «рельсы мышления» оказываются вдруг где-то не проложены, только при столкновении с чем-то действительно новым,

---

[37] Обзор подхода к мышлению как выходящему за границы тела, включая критику таких подходов, см. в https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7250653/

можно переходить на затратное «просто мышление, уж как можем», задействовать какие-то иные, поисковые механизмы мышления, «голый биологический интеллект», как-то иногда методом проб и ошибок задействующий внешние средства типа компьютерного моделирования. Но такие выходы за пределы знакомого в мышлении — исключение, а не правило. Вам очень повезло, если вы оказались на таком фронтире, радуйтесь, что вы в первых рядах человеческой цивилизации.

Но не факт, что вы сможете на этом фронтире что-то придумать, пополнить запас эффективных паттернов мышления человечества: эволюция с методом проб и ошибок очень действенна, но большинство проб оказываются ошибками и на удачные пробы может просто не хватить времени и ресурсов.

Эти ускоряющие мышление взятые из культуры паттерны используются как в самых базовых видах мышления (логические рассуждения общего вида), так и в основанных на них более сложных (инженерное мышление общего вида), так и в быстро меняющихся ещё более специализированных и сложных вариантах рассуждений, включающих смесь познания/мышления и каких-то рутинных прикладных рассуждений по уже давно известному материалу.

Паттерны мышления используются в труде инженера-авиастроителя, менеджера, инвестора, дрессировщика дельфинов, танцора, учителя начальной школы, политика (при этом они могут быть людьми, но могу быть и роботами, и даже организациями из людей и роботов). Мышление должно быть культурно/практично/стильно/деятельно — оно должно следовать методу/способу. Это означает, что интеллект должен как-то повторять одни и те же приёмы мышления, если это лучшие известные приёмы мышления. Слабые приёмы мышления, которые вы придумываете каждый раз «под проблему» — не факт, что они справятся. Чтобы быть умным, надо знать, как быть умным. Паттерны сильного мышления надо выучить. Это делается в форме изучения приёмов/паттернов мышления интеллект-стека.

Беглости мышления нужно добиваться в любом мышлении, мышление надо тренировать, как любое другое поведение, как любой другой метод/практику/культуру/стиль/деятельность/стратегию.

Кроме того, что выученные и далее натренированные до беглости культурные/стильные/практичные/«каким-то методом» паттерны мышления дают выигрыш в скорости по сравнению с мышлением-дичком, они предохраняют от грубых мыслительных ошибок. Один раз выучиваете операцию умножения — всю жизнь затем используете. Один раз выучиваете, что систему нужно рассматривать функционально в момент её работы в составе надсистемы, а ещё рассматривать конструктивно в момент её создания и развития — и тоже используете всю жизнь.

«Платят за согласование и удовлетворение интересов внешних проектных ролей, а не за «реализацию концепции системы» и уж тем более не за «реализацию требований», которых в современной инженерии уже нет, сама идея «требований» устарела» — этот материал излагается в нашем руководстве по системному мышлению, один раз для всех систем. Если вы его не выучили не применили в жизни, то можно и без денег остаться! Лучше бы это выучить один раз, чтобы потом всю жизнь не ошибаться! Знание приёмов системного мышления очень практично, экономит время, спасает от ошибок.

Для «образованного человека» нужно освоить один раз компактное мышление по методам интеллект-стека, и оно пригодится ему для самых разных деятельностей и проектов (но потом не забывать его обновлять, ибо . Ведь человеку придётся в жизни играть много самых разных трудовых/проектных/профессиональных ролей, занимаясь методами самой разной инженерии самых разных систем — выращивать цветы, детей, изготавливать шестерёнки, программировать ассистентов AI, обустраивать гостиничный номер, обустраивать палатку в горах, ловить змей, проектировать фемтосекундные лазеры, договариваться с AI по поводу формата выдачи материалов и т.д. Тут будут и рабочие методы, и методы/способы/культура поведения члена семьи, родителя, избирателя, политика.

Человек (хотя не только человек, но и предприятие, а то и компьютерная система с AI, или даже без AI) в жизни играет много ролей, за каждой из которых стоят какие-то иногда крупные («авиастроение»), а иногда очень дробные и маленькие («производство подшипников») по требуемым для них знаниям/объяснениям/теориям прикладные методы создания каких-то систем определённого уровня организации. Каждый из этих прикладных методов предполагает какие-то свои специфические прикладные паттерны рассуждений, но при столкновении с новым и неожиданным поворотом в проекте для решения проблем нужно будет подключать в мышлении и весь интеллект-стек. Хорошее фундаментальное образование даёт конкурентное преимущество!

Будь вы основателем фирмы на рынке секс-игрушек, или менеджером проекта космического туризма, или инженером квантовых компьютеров — вам придётся для исполнения всех многочисленных прикладных методов для этих занятостей быть собранным, задействовать логику, согласовывать сложные модели систем с вашими коллегами, удерживать внимание на многочисленных ваших и чужих системах, которые затрагивает ваш проект и которые затрагиваются вашим проектом, вы будете использовать компьютеры с универсальными (AI) и не очень универсальными (обычный корпоративный софт) алгоритмами. В следующем проекте всё повторится, но на совершенно другом содержании проекта: весь ваш интеллект потребуется опять, чем бы вы ни занялись: проекта, где всё известно и можно рассуждать только по правилам, не бывает. А если есть какое-то совершенно знакомое действие, его даже «проектом» не назовут!

Мышление как деятельность интеллекта по решению проблем, по познанию сложно устроенного и непрерывно меняющегося мира — оно универсально, оно всегда будет с вами, и **системное мышление входит в состав этого мыслительного минимума цивилизованного человека** и цивилизованного AI. Хотя про системное мышление можно сказать и то, что оно входит в состав мыслительного минимума цивилизованной организации: системное мышление коллективно, оно объединяет интеллекты в организации, усиливает интеллект и организации в целом.

Обязательно нужно учитывать, что речь идёт о лучших на сегодняшний момент (state-of-the-art) приёмах мышления. Базовые приёмы мышления относительно стабильны (время их изменения может исчисляться сотнями лет: сколько веков было аристотелевой логике до момента прекращения её использования?), но в 21 веке и базовые приёмы за время длинной человеческой жизни могут меняться, так что тут нужно быть начеку и вовремя переучиваться. Удивительно, но немногие сегодня знают, что аристотелева логика с её силлогизмами осталась в истории, вместо неё сейчас множество вариантов математической логики. При этом логика

перестала считаться основанием всей математики, поменялась и математика, и математическая логика (подробней об этом в руководстве по интеллект-стеку, в разделах математики и логики).

Уже в 21 веке существенно изменилось понимание самого интеллекта, научного мышления, причинно-следственных отношений, логики как вероятностного вывода, да и самого системного мышления. Если вы будете изучать эти предметы по учебникам более древним, чем 2017 год издания, то вы можете удивиться, насколько они уже не отражают современное состояние методов мышления, основанных на этих дисциплинах. Не учитесь старью! Не учитесь системному мышлению авторов 80-х годов 20 века, проверяйте годы издания ваших учебников! **В нашем руководстве системное мышление приведено на момент середины 2025 года!**

Так, при поиске учебника системного мышления в гугле одним из первых находится учебник тренеров нейролингвистического программирования Джозефа Коннора и Яна МакДермотта в переводе на русский язык. Но этот учебник в английском оригинале был написан аж четверть века назад, в 1997 году[38]! Неудивительно, что он так сильно отличается по содержанию от наставлений нашего руководства, в нём приведены довольно древние представления о системном мышлении. Системное мышление не стояло на месте, оно интенсивно развивалось в 21 веке, ибо интеллект-стек эволюционировал, в дисциплинах/теориях его методов всё активней использовались понятия системного мышления — и системное мышление получало обратную связь в ходе его использования в других методах мышления, поэтому оно тоже активно эволюционировало.

Моделирование мыслительного мастерства

Оцените по десятибалльной шкале, насколько вас научили мыслительному мастерству интеллект-стека в вузе, и сколько вы добавили самообразованием (в сумме 10 баллов на вуз+самостоятельное изучение). Это ничего, что вы не очень понимаете содержание интеллект-стека (оно будет раскрыто в руководстве по интеллект-стека): попробуйте догадаться по краткой характеристике, данной в нашем руководстве. Приведите год, в котором ваши знания являлись бы SoTA. Скажем, если вы изучали физику в последний раз в 1980 году, то этот год не может быть свежее 1980, но вероятнее всего, вы освоили на тот момент ещё более древнюю версию метода мышления, основанному на более древней дисциплине/теории, скажем ньютонову механику 1687 года, когда впервые были сформулированы законы Ньютона, а не квантовую механику или теорию относительности. Оценка вашего текущего мастерства может учитывать кривые забывания: если вы были отличником по какому-то методу мышления в 2010 году и тогда поставили бы себе 10 баллов, но с тех пор как-то ни разу не освежали своё мастерство в применении этих методов мышления — умение на сколько баллов у вас осталось?

[38] https://www.amazon.com/Art-Systems-Thinking-Essential-Creativity/dp/0722534426

Мастерство в работе по методу интеллект-стека	Оценка текущего вашего мастерства по десятибалльной шкале	Из них научили в вузе	Из них научился сам	Год версии метода (год появления дисциплины метода)
Системная инженерия				
Методология				
Риторика				
Этика				
Эстетика				
Исследования/познание				
Рациональность				
Логика				
Алгоритмика				
Онтология				
Теория понятий				
Физика				
Математика				
Семантика				
Собранность				
Понятизация				

Задания: мышление и интеллект-стек.

Поставьте отметки о выполнении дел:

1. Работа по руководству рациональной работы освоена, «машинка типов» у меня работает, я не ленюсь мыслить письмом/моделированием (записывать результаты выполнения заданий), могу удерживать внимание на работе с руководством по 2 часа в день без отвлечений.
2. Моё расписание на ближайший месяц откорректировано: на проект «освоение системного мышления по руководству» выделено 2 часа в день, заниматься буду за компьютером (а не читать урывками с телефона в метро), главным образом за компьютером на работе. Понял, что заниматься системным мышлением — это не столько читать руководство, сколько писать/моделировать рабочие ситуации.
3. Написан и опубликован первый пост по первому разделу руководства по системному мышлению. Что пришло в голову в ходе прохождения наставлений этого раздела? Что было наиболее удивительным? Приведите в посте список терминов, которые вы увидели впервые в этом разделе («список незнакомых мне слов»).
4. Освоение исследовательского руководства по интеллект-стеку входит в отдельную программу — исследовательского развития, её обычно проходят последней (после программ личного развития и рабочего развития). Тем не менее, это руководство было проглянуто вами «по диагонали» (как минимум,

посмотрено оглавление). Напишите пост с описанием того, что содержательного вы ожидали увидеть в этом руководстве, и что увидели при «пролистывании». Пост не формальный, то есть про форму, а не содержание руководства, то есть не пост вроде «ожидал увидеть буквы, и увидел их», «ожидал увидеть набор методов мышления, по одному методу на каждый раздел, и увидел, что так и есть». Легко проверить: есть ли в вашем посте какие-то термины из руководства по интеллект-стеку, или нет таких терминов? Если нет, то вы не коснулись в посте его содержания.

2. Наш вариант системного мышления: третье поколение

Варианты системного подхода

Системное мышление (systems thinking) — это мышление с использованием основных понятий и мыслительных приёмов **системного подхода** (systems approach). Есть много разных вариантов системного подхода, существенно отличающихся друг от друга в степени проработанности, используемой ими терминологии и деталях, но совпадающих в своих основах. Главное во всех вариантах системного подхода — это многоуровневое рассмотрение системы как части какой-то надсистемы **сначала**, чтобы **потом** рассматривать подсистемы как части системы. Этот мыслительный ход (сначала к надсистеме, «наверх» по системным уровням, потом к подсистеме, «вниз» по системным уровням) выполняется на много уровней вверх и вниз.

Речь идёт о выделении систем, надсистем, подсистем исключительно вниманием прямо на работающей/функционирующей системе. Никакой разборки систем на физические отдельные части не производится. В ходе физической сборки/разборки во время создания системы (а не во время использования/работы/функционирования) такое «строительное» рассмотрение конструктивных/материальных частей системы тоже есть, но оно в системном мышлении оказывается не главное. Главное — это в момент использования/работы/функционирования, и вот эти функциональные части системы (функциональные объекты, роли) выделаются вниманием. Чтобы не утерять внимания, используем запись — системное мышление предусматривает системное моделирование, оно не проходит чисто «в уме». Не пишете — не мыслите! Об этом «выделении вниманием» и записях для неутери внимания при отвлечениях мышления подробней наставлялось в руководстве по рациональной работе.

Впервые системный подход появился в физике. Понятие системы в современном системном подходе более развито, чем понятие системы в физике. Система в физике — это просто часть мира/вселенной в рассмотрении. В этом плане есть рассматриваемая часть всего мира как система, граница системы и весь остальной мир за границей системы как окружение/среда/environment. Системы в физике (например, понятие термодинамической системы) долгое время не относили даже к полноценному системному подходу, потому как в физике особо не обсуждалась многоуровневость систем: там хватало обсуждения системы, состоящей из каких-то частей в окружении, и только. В то же время понятие системы в физике используется весьма активно до сих пор[39].

[39] https://en.wikipedia.org/wiki/Physical_system

Системный подход как основа системного мышления именно под названием systems thinking появился сначала на биологическом материале. Биологи пытались описать заливной луг как целое с его сотнями видов растений и животных и круглогодичными изменениями. Живое на части не разрежешь, луг оказался исключительно сложным объектом для описания и понимания. Поэтому **системное мышление появилось как управление движением внимания исследователя по разным уровням деления целой системы на части (или наоборот, сборки вниманием целой системы из отдельно выбираемых вниманием частей).**

Основы системного подхода претерпели существенное развитие с момента предложения в 1937 году биологом Людвигом фон Берталанфи[40] **общей теории систем**. Вообще, **подход** (approach) — это когда разработанные в рамках одного метода на примере одной предметной области понятия, методы мышления, приёмы действия и инструменты их поддержки применяются затем к другим предметным областям. Общая теория систем была разработана главным образом на развитом на биологическом материале понятии системы из физики (это было что-то типа «теория физики для биологов»), а уж затем было предложено применять её положения ко многим и многим другим предметным областям («общая теория систем», оторванная уже и от предметной области биологии).

Сегодня предложение каких-то важных типов объектов и отношений в какой-то предметной области чаще всего называют framework, что переводится тоже как «подход». Этот «подход»/framework представляет собой какую-то онтику (частную онтологию, не согласованную с другими онтиками в других предметных областях, всё-таки под онтологией понимается какое-то общее понимание объектов и отношений, существующих в мире), задаваемую не на формальном математическом языке, а в виде не слишком формального текстового описания. Мы дальше отметим, что чем выше уровень абстракции таких «подходов», тем меньше они будут похожи на «математические формализмы», тем больше они будут выражаться текстами на естественном языке, задающим понятия и отношения между понятиями путём предъявления каких-то примеров употреблений этих понятий в тексте, а не путём математических формальных определений, как в математике. Интеллектуальные агенты сегодня — это не логические компьютеры, а живые люди и AI-агенты, работающие не в локальных/символьных (как математики, «с формулами»), а в распределённых (как в мозгу или компьютерной нейросети) представлениях, несимвольных (можете думать о «голографии» как примере нелокальных представлений, хотя этих нелокальных представлений существует множество самых разных вариантов[41]). Вот на таком низком уровне формальности описывается и набор объектов и отношений системного подхода/approach/framework, хотя начальные работы по системному подходу и пытались унаследовать более формальный язык описаний из физики, откуда и пришло понятие системы.

Онтология нашего варианта системного мышления приведена (с источниками литературы, откуда взяты эти понятия) в кратком виде в последнем разделе нашего руководства, и это тоже сделано в виде текста на естественном языке, ссылающегося на источники литературы, которые тоже являются текстами на естественном языке. Но важно подчеркнуть: **системный подход подразумевает тексты, изложенные не в рамках прототипной теории понятий (много**

---

[40] https://ru.wikipedia.org/wiki/Берталанфи,_Людвиг_фон

[41] https://ailev.livejournal.com/1281819.html

метафор, произвольная типизация), а в рамках теоретической теории понятий (строгая типизация объектов и отношений). Если вы не поняли выделенный предыдущий фрагмент текста, вам обязательно нужно вновь обратиться к руководству по рациональной работе, наставления по онтологическому моделированию — там!

С момента появления общей теории систем в 30-х годах 20 века на базе системного подхода возникали и умирали целые дисциплины/теории. Например, так родилась в 1948 году и затем в семидесятых была предана забвению **кибернетика**. Поэтому до сих пор можно встретить старинные варианты системного подхода, существенно переплетённые с кибернетикой и несущие в себе все её недостатки, прежде всего попытку свести всё понимание мира к работе поддерживающих гомеостаз (т.е. неизменность своего состояния) систем с обратными связями.

Кибернетика активно использовалась в госпланировании и показала там неадекватность: экономика сама по себе неравновесна, никакого «баланса спроса и предложения», возвращающего к равновесию, нет, ибо экономика развивается, меняется, а не балансирует вокруг какого-то «равновесия». Остатки кибернетики существуют теперь только в виде теории автоматического регулирования, где действительно нужно управлять в технической системе каким-то постоянным параметром, следить за «отклонениями». Но в большинстве ситуаций речь идёт не об «отклонениях» от точки равновесия, а как раз о продвижении к каким-то целям и опоре на неравновесные состояния. Современные варианты описания взаимодействия системы и окружающего мира (например, теория active/embodied inference) не предполагают выделения «управляющей системы» в простой «петле обратной связи», и они базируются на других предположениях о принципах взаимодействия системы и окружения, нежели кибернетика.

Самый распространённый вариант кибернетического системного подхода отражён в способе моделирования «системная динамика» (system dynamics[42]) и сводится к нахождению и явному отражению в модели каких-то связей, которые могут замыкаться в циклы, приводя к появлению колебаний вокруг какого-то положения равновесия. Такое «кибернетическое моделирование» сверхупрощено и плохо отражает самые разные виды систем, совсем не похожие на кибернетический «регулятор Уатта».

Развитие, эволюция отлично описываются системными представлениями, но плохо описываются представлениями «управления»/control, то есть представлениями кибернетики. В целом моделирование разных по природе (электрических, химических, экономических и т.д.) связей на одном системном уровне (и даже на разных системных уровнях) выполняется произвольными системами дифференциальных уравнений. Иногда это описание произвольными системами дифференциальных уравнений в инженерии называют «системное моделирование», но оно существенно шире узкого класса уравнений «системной динамики». Но это уже не совсем системный подход, это просто имитационное моделирование систем в «железной» инженерии (и иногда организационных систем). Так что если вы увидели «системный моделер», то это необязательно системный подход и системное мышление — чаще всего речь идёт о «системном моделировании» как варианте акаузального имитационного моделирования киберфизических систем, чаще всего на языке Modelica. Как и в случае системной

[42] https://en.wikipedia.org/wiki/System_dynamics

динамики, для этого есть большое количество разнообразного софта[43]. Так что какой-нибудь System Modeler[44] от Wolfram — это совсем необязательно моделер, поддерживающий системный подход и системное мышление. Это просто инструмент физического моделирования киберфизических систем («железных» систем, в которых есть какие-то управляющие компьютеры, «кибер» часть).

Системный подход уже получил широкое распространение в инженерии и менеджменте. В инженерии в пятидесятые-шестидесятые годы прошлого века превалировало «математическое» понимание системного подхода, которое по факту сводилось просто к активному использованию математического моделирования при решении инженерных проблем («системное моделирование» как имитационное моделирование — отголоски ещё тех времён). «Системность» заключалась в том, что модели при этом набирались из самых разных теорий/дисциплин, как естественнонаучных, так и инженерных (электрика, механика, химия для случая химических реакций, расчёты изменения давления, тепломассопереноса и т.д.) для разного уровня структуры системы (системы в целом и подсистем разного уровня).

Описание тех или иных «систем» проводилось с использованием многочисленных моделей, отражающих разные интересующие инженеров и учёных свойства систем в различных ситуациях.

Чаще всего такое в большой мере завязанное на математические имитационные/simulations модели **системное моделирование** (часто говорили «**системный анализ»,** ни о каком синтезе тогда и речи не было) противопоставлялось так называемому **редукционизму** (сведению к простому).

Для редукционизма, объявлявшегося главным в науке (это же «сведение к простому», деление на всё более и более мелкие части — научность искалась тут, «как более сложное объяснить работой более простого») было характерно выделение одной главной точки зрения, одной теории/дисциплины и метода моделирования для какого-то уровня структуры объекта или предмета исследования. Скажем, человек рассматривался на уровне молекул (т.е. биохимическом уровне), и из этого пытались выводиться все знания о человеческой природе: в том числе и его мышление, и социальное поведение объяснялось как сложное сочетание биохимических процессов.

Системный подход преодолевал очевидную бессмысленность одноуровневого упрощенчества редукционизма, и поэтому стал очень популярен. **Системно мыслить — это прежде всего удерживать во внимании тот уровень дробления системы на части, на котором уместно обсуждать проявляющиеся на этом уровне новые (emergent, эмерджентные) свойства, которых ещё не было на предыдущих уровнях разбиения системы на части, и уже нет на уровнях выше системы, на уровне надсистемы. Эти свойства необходимы для удобства действия в физическом мире.**

Системный подход не подразумевает простого физического или математико-логического понимания отношения «часть-целое». Так, «дом состоит из кирпичей» — это верное утверждение в физике и математике, но неверное в

---

[43] https://modelica.org/tools/

[44] https://www.wolfram.com/system-modeler/

системном подходе. В системном подходе заметят, что пропущен ещё один уровень в системном разбиении целых систем на части («состоит из» — это отсылка к отношению композиции/«часть-целое»): стены! Если вы будете строить дом, то указывать положение каждого кирпича в доме, чтобы более-менее компактно описать дом — это очень неправильно. Много проще описать, как из кирпичей составлены стены, а затем описать, как из стен составлен дом. Системы вводятся тогда, когда их введение позволяет что-то описать более компактно, удобно, проще для действия в физическом мире.

Вкус борща в момент его готовки нужно обсуждать как зависящий от способа приготовления его из кусочков овощей и мяса, неадекватно обсуждать идущие в ходе готовки биохимические процессы на уровне клеток растений-овощей и клеток мяса. Эти процессы никуда не деваются, они вполне себе идут в ходе готовки, но это неправильный уровень структуры вещества для обсуждения вкуса борща! Знание о том, как сворачивается белок мяса в ходе варки борща, конечно, имеет непосредственное отношение к изменению вкуса сырого мяса на варёное, но вряд ли это поможет повару! Вниманием нужно выделять целые овощи и их куски, приёмы готовки и зависимость вкуса от этих приёмов обсуждаются на этом уровне крупности вещества: целые овощи и куски мяса, нарезанные на небольшие кусочки, принятые в той или иной кухне (чуть более крупные в южной готовке, чуть более мелкие в северной). И нельзя обсуждать вкус борща, если обсуждать званый вечер со сменой шести блюд, где борщ будет только одной из смен: обед уже не имеет «вкус борща», хотя борщ там и является его составляющей частью. И главное — это просто выделение вниманием в реальной ситуации готовки борща и реальной ситуации обеда нужных нам для каких-то целей (приготовление обеда с вкусным борщом) частей.

Про редукционизм «борща из свёрнутых в ходе варки белков» будут все те же самые рассуждения, что и про «дом из кирпичей».

Ситуация с домом или борщом кажется простой, но давайте возьмём проект создания авиалайнера, в котором планируется 6 млн индивидуальных деталей. Как вы с огромной командой из пары сотен тысяч человек, занятых его изготовлением на заводах разных стран, будете рассматривать этот авиалайнер, чтобы не упустить ничего важного? На уровне материалов, из которых этот авиалайнер состоит? Это будет правильно, если считать прочность лайнера. Но если считать подъёмную силу его крыльев, то уровень материалов не поможет. Большую и сложную систему из миллионов индивидуальных частей нужно описывать на множестве уровней её сборки в целое, причём описывать для самых разных целей самыми разными способами, не терять ни один из них! **Системное мышление помогает не потерять внимание тысяч людей, не забыть что-то важное при делении на части или при сборке целого из частей, не отвлечься на неважное.**

Управлять вниманием к 6 млн индивидуальных деталей в авиалайнере, чтобы не забыть ни одной детали, и рассмотреть и аэродинамику, и пассажировместимость, и общую стоимость проекта, и безопасность при попадании молнии, и размеры цеха для сборки авиалайнера — вот это всё стало не интеллектуальным подвигом, а обыденной мыслительной работой после появления системного мышления в инженерии. И имена гениев-авиаконструкторов вроде Мессершмитта и Туполева остались в прошлом, для современных более сложных самолётов уже не нужно иметь гениев в составе команды! Системное мышление, поддержанное компьютером (даже без компьютера с AI), вполне справляется. Не нужно иметь

абсолютного гения Королёва, чтобы делать такие сложные запуски космических кораблей, какие делает сейчас SpaceX.

По этой линии борьбы с редукционизмом дошли до того, что системное мышление начали объявлять **холистическим,** то есть говорящем о примате целого над частями. В холизме поведение частей объясняется существованием целого. Но **холизм** оказался такой же ошибкой, что и редукционизм: **системное мышление борется с полным отказом от рассмотрения зависимостей поведения целого как поведения его частей (редукционизм) как и полным отказом от рассмотрения зависимостей поведения частей в зависимости от происходящего с целым (холизм).**

Слово «система» в конце семидесятых годов стало респектабельным, и его стали использовать в том числе и те люди, которые были совсем незнакомы с системным подходом в любой его версии, которые не понимали сути системного подхода, его способа управления вниманием при рассмотрении сложных ситуаций. По факту, слово «система» вдруг стало синонимом слова «объект» — что-то, что попало в сферу нашего внимания. Связь со вниманием осталась, но специфика того, что речь идёт о внимании к определённому уровню крупности нарезки на объекты, и уровней этих множество, и способов нарезки тоже множество — вот это было полностью потеряно. Никакого системного мышления, которое потом бы работало с «объектами-системами», увы, у пользующихся словом «система» не было.

Это «попсовое» понимание слова «система» было унаследовано и современными системами AI (Bard, Claude, ChatGPT и т.д.). Увы, с ними нельзя поддержать разговор про современное системное мышление: в бытовом понимании слово «система» не тип объекта, как в системном подходе, а «почти синоним» объекта. Вместо теоретической теории понятий и её строгой/формальной типизации (медленное мышление S2 по Канеману[45]) в системах искусственного интеллекта используется прототипная теория понятий с её нестрогими аналогиями как основой мышления (быстрое мышление S1 по Канеману), причём не из современного системного подхода, а из обыденной речи. Поэтому осторожней беседуйте с системами AI (а также с обывателями) на тему системного мышления, слово «система» знают все — но для них всех это не тип объекта из системного подхода, и не все могут отследить строгость использования типа!

В восьмидесятых в менеджменте тоже появилось множество учебников системного подхода, математики там уже не было. Акцент делался на том, что в системе «всё со всем связано», при этом существенные связи могут выпасть из традиционных монодисциплинарных рассмотрений (опять борьба с редукционизмом!). Поэтому нужно привлекать самых разных людей, чтобы в их общении получить возможность выявления этих существенных связей.

Менеджерское изложение системного подхода было ценным тем, что в нём обратили внимание на необходимость при обсуждении каких-то целевых систем учитывать их системы-создатели, обычно состоящие из людей, организованных в предприятия. Потом рассмотрение этих людей сделают обязательным, и речь пойдёт не о самих людях, а об их ролях в проекте, которые назовут стейкхолдерами/stakeholders. Иногда стейкхолдерами называли при этом исполнителей ролей, а иногда и сами роли — стейкхолдером в «инженере Васе» некоторые люди считали Васю, а некоторые — инженера. Мы в нашем руководстве

[45] https://ru.wikipedia.org/wiki/Думай_медленно…_решай_быстро

поэтому избегаем говорить «стейкхолдер» и называем инженера ролью, а Васю — агентом (играющим роли, в том числе роль инженера). В любом случае, без систем-создателей обсуждать целевую систему стало неправильно, тем самым в восьмидесятых годах прошлого века появилось второе поколение системного подхода. Хотя исполнителей ролей и сами роли нещадно путают до сих пор, слово stakeholder используют очень по-разному, будьте тут аккуратны.

В книгах с менеджерским изложением «системности» на каждую рекомендацию «учитывать целостность системы», «думать холистически», «смотреть на проблемы с разных сторон» нужно было бы дать ещё десяток: как именно это делать. На каждое «думать холистически» не мешало бы напоминать, что про части системы тоже нужно не забывать, холизм так же полезен и так же вреден, как и редукционизм. В менеджерском системном мышлении (это очень, очень много книг! Если вы начнёте гуглить, с большой вероятностью найдёте именно менеджерские книжки) многоуровневость разбиения системы на части не подчёркивалась, различные способы разбиения на части (например, функциональные части — удобные для обсуждения функционирования/работы системы, конструктивные части — для обсуждения того, из чего система сделана) не рассматривались.

Например, возьмём ножницы: функционально там ножевой блок и ручка (время использования/эксплуатации, то есть объясняем, «как работают»), а конструктивно — два конца, два кольца и посредине гвоздик (время изготовления, «как и из чего сделать»). Оба описания (дальше мы узнаем, что таких описаний надо больше) нужны для того, чтобы спроектировать, изготовить, а затем эксплуатировать ножницы. Теперь найдите подобное различение разных делений на части системы в «попсовых» книжках о системном мышлении в менеджменте. Нет, этого там не будет (в нашем руководстве — будет!), просто говорится, что «систему нужно описывать разными способами». Это понятно и без учебников, что разными способами. А вот какими именно способами?

Ответ на этот вопрос в менеджерских книгах отсутствовал, постулировалось просто, что «надо думать о частях и целом, это крайне полезно», и всё. В инженерных книгах по системному мышлению (вернее, книгах по использованию системного мышления в инженерии — книгах по системной инженерии) ответ на этот вопрос («сколькими способами минимально надо разбивать систему на части?») давался, причём со временем число этих способов росло — сейчас это минимально четыре способа (и уже есть кандидат на пятый способ), и это минимальное число способов разбиения на части! Мы их рассмотрим подробнее в последующих разделах нашего руководства.

Такая же неконкретность в советах по управлению вниманием в сложных ситуациях может быть обнаружена во многих книгах по общей теории систем, восходящих к трудам Людвига фон Берталанфи: прописанные там общие закономерности мало отличаются от философских обобщений, их трудно непосредственно применять в деятельности.

Да, хорошо бы думать о системе в целом — но как вообще увидеть систему в её границах, когда вокруг мельтешит довольно пёстрый и сложный непрерывно меняющийся мир? Вот мы захватили своим вниманием какой-то объект в этом мире. Это будет надсистема или подсистема? А если соседи по проекту увидели систему совсем по-другому, провели границу системы другим способом, нашли в системе другие части, определили функцию системы в надсистеме не так как вы, что в этом

случае делать именно вам?!

Менеджерские книжки по системному подходу выглядят пожеланием «быть здоровым и богатым, а не бедным и больным». Никто не возражает «смотреть на систему с разных сторон»! Но с каких именно сторон? И как смотреть на что-то невидимое, например, на вездесущий в менеджерских книгах «процесс»?

Самых разных школ системной мысли с различающимися терминологиями, выделенными самыми разными основными принципами системного подхода, какими-то наработанными вариантами системного моделирования существуют десятки и сотни. Поэтому говорят о **системном движении**, у которого нет каких-то влиятельных координаторов или ярко выраженного центра, просто отдельные люди в разное время в разных странах чувствуют силу системного подхода и начинают им заниматься самостоятельно, не слишком сообразуясь с другими. А поскольку критериев для отнесения той или иной школы мысли к системному движению нет, то иногда «патриоты» в России и тектологию А.Богданова считают ранним вариантом системного подхода[46]. Более того, это потихоньку транслируется на англоязычную аудиторию[47].

Буквально в последние пять лет появились работы физиков, которые пытаются объяснить сложность биологических систем со множеством уровней организации/эволюционных уровней/системных уровней как вытекающую из физических законов. Раньше эти попытки не удавались, но Giorgio Parisi получил нобелевскую премию по физике 2021 года[48] за открытие явлений неустроенности/frustration (не путайте с психологическими «фрустрациями», это от совсем других, геометрических «неустаканенностей»[49], термин пошёл в физике с 1977 года). Конфликты между объектами разных системных уровней приводят к беспорядку и флуктуациям (те самые неустроенности/неустаканенности/frustrations) в системах от атомарных до планетарных пространственно-временных масштабов.

Физики стали изучать механизмы процессов, в которых участвуют неэргодические системы, то есть системы с памятью. Первым хорошо изученным примером таких систем стали спиновые стёкла. Стекло — это не кристаллическая структура, но и не жидкость. Стекло нельзя сначала нагреть, затем охладить — и сказать, что оно пришло в то же состояние, как это было бы с кристаллической решёткой или жидкостью. Нет, после нагревания и последующего охлаждения состояние стекла (расположение фрагментов кристаллической структуры) будет другое, ибо в стёклах есть память, они не эргодичны[50] в отличие от самых разных других систем в физике. Эти исследования позволили продвинуть понятие «система» так, что системы в биологии получили объяснительные модели с опорой на физику и математику.

Идея неустроенностей/frustration из объяснений поведения стёкол как систем с памятью позволила физику Кацнельсону и биологам Вольфу и Кунину в 2018 году сделать предположение[51], что сложность биологических систем и вся эволюция в

---

[46] Например, это прописано в статье русскоязычной Википедии: https://ru.wikipedia.org/wiki/Тектология
[47] https://bogdanov.systemeconomics.ru/en/
[48] https://www.nobelprize.org/prizes/physics/2021/parisi/facts/
[49] https://en.wikipedia.org/wiki/Geometrical_frustration — и посмотрите на видео, которое очень хорошо объясняет суть этой «неустроенности/неустаканенности».
[50] https://ru.wikipedia.org/wiki/Эргодичность
[51] https://www.pnas.org/doi/10.1073/pnas.1807890115

целом происходят именно от вот этих «неустроенностей», причиной которых становятся конкурирующие/конфликтующие взаимодействия на разных системных уровнях. Скажем, клетки печени хотят неограниченно размножаться, но им это не дают — ибо для организма это же будет рак печени! Или паразит хочет заразить и убить всех хозяев, но тогда вымрет вся его популяция, и выживают только не слишком заразные паразиты.

Именно эти «неустроенности» от конфликта устремлений «спастись» систем на разных системных уровнях (в биологии — уровнях организации: молекулы, клетки, организмы, популяции, экосистемы, все они стремятся сохранить своё существование сейчас и в будущем) порождают все более и более сложные системы всё более и более высоких уровней организации. Это и есть источник жизни в её многообразии. **Жизнь — это физический процесс, порождающий сложность за счёт преодоления неустроенностей, проистекающих из-за конфликтующих взаимодействий на разных системных уровнях.**

Эволюция (как показывает работа 2022 года Ванчурина, Вольфа, Кацнельсона, Кунина «Toward a theory of evolution as multilevel learning»[52]) оказывается многоуровневой оптимизацией вот этих неустроенностей, работа эволюции оказывается очень похожа на работу нейронной сети, многоуровнево оптимизирующей свою структуру на каком-то потоке входных данных. Системное мышление из физики (а именно, термодинамики) вернулось в биологию, и принесло объяснительную теорию на основе математики, включая и объяснение существования всё более и более сложных системных уровней в ходе эволюции (от молекул к клеткам, от клеток к организмам, от организмов к популяциям)[53]. Результаты этих догадок физиков безмасштабны, то есть приложимы не только к существам как биологическим системам, но и к их сообществам, а также к сообществам разумных существ (которые ведь тоже физичны!). Более того, эти результаты применимы и к техно-эволюции, разве что геном заменяется на более общий мемом — и в техно-эволюции мемом находится не в каждой клетке описываемого геномом организма, а отдельно от техно-системы, где-нибудь в конструкторском бюро.

Интересно, что (по нашему опыту) работы по приведённым в предыдущем абзаце ссылкам вы прочтёте не сейчас, а только в ходе освоения методологии или даже системного менеджмента: когда будете изучать теорию стратегирования или даже методы орг-стратегирования. Стратегирование (выбор метода, который потом будет принят всерьёз — по которому потом надо будет работать системе, в том числе вашей организации или даже вам самому) вы делаете в условиях техно-эволюции, поэтому хорошо бы понять теорию происходящего. Вот она как раз и изложена в этих работах. Это основа современной эволюционной системной инженерии, подробно техноэволюция рассмотрена в руководстве по системной инженерии. Можете сразу посмотреть работы по ссылкам из предыдущего абзаца, там даже можно найти объяснение, почему все телефоны или автомобили так похожи друг на друга — и существуют на рынке одновременно в огромном разнообразии моделей. Это не сильно отличается от биологических видов: там ведь тоже сосуществует огромное их количество, и они все существенно похожи друг на друга.

Точно так же инженеры в последние несколько лет выяснили, откуда и в

---

[52] https://www.pnas.org/doi/10.1073/pnas.2120037119

[53] https://www.pnas.org/doi/full/10.1073/pnas.2120042119

технических системах (какая-нибудь система управления авиалайнером), и в биологии (управление велосипедом при спуске по горной дороге) возникают такие сложные обратные связи в поддержании устойчивого управления. Оказалось, что это нужно для достижения точности и скорости одновременно. Обычно элементная база (техническая или биологическая — не имеет значения) или медленна и точна, или быстра и неточна. Если предусмотреть множество обратных связей в самой системе управления, и достаточную разнородность характеристик элементов по шкалам скорости и точности, то можно предложить удивительно хорошо работающие механизмы и даже (по мере развития генной инженерии) организмы.

Скажем, танцоры обладают удивительно точным управлением своим телом, при этом биологи удивляются, насколько медленно и неточно работает wetware («мокрое обеспечение», «мясо») при подобных характеристиках скорости и точности. Грубо говоря, большие мышцы быстро и сильно, но неточно двигают руки-ноги-тело к нужному месту в пространстве, а мышцы поменьше, поточнее и помедленней подруливают, управляясь не столько даже многоуровневыми вычислениями, сколько просто запомненными паттернами управляющих мышцами сигналов (ибо доступ к запомненным значениям быстрее вычислений). Мозг работает тоже не очень быстро, но он берёт на себя только высокоуровневое управление, а основное управление движением идёт на многих уровнях самых разных рефлексов.

Это исследование команды John Doyle[54] показало универсальный характер найденных закономерностей в устройстве многоуровневого управления с множественными обратными связями на необходимо разнообразных элементах. Это исследование выводит заново на идеи кибернетики (хотя слово «кибернетика» в этих исследованиях и не используется) об обратных связях — они должны быть множественными, и внутренними «от контроллеров», а не внешними «от датчиков», как в классической кибернетике! Так что системное мышление обновилось ещё и в части создания надёжных, точных и быстрых систем управления, причём сами закономерности описаны безмасштабно.

Сам John Doyle использовал результаты своих исследований сначала для создания контроллеров в киберфизических устройствах, но затем обнаружил, что это хорошая объяснительная теория для устройства управления в биологических системах. Теперь его интересуют общественные системы и даже человечество в целом, которые он рассматривает при помощи того же математического аппарата и тех же концепций, что и найденные им в ходе изучения киберфизических систем.

И это не единственные новинки системного мышления, которые появились за последние пять-десять лет. В нашем руководстве мы затронем ещё несколько современных трендов в системном подходе. Системный подход активно развивается и прямо сейчас, для него появляются объяснительные физические теории, поддержанные математикой и не противоречащие наблюдениям в технике и биологии.

Моделирование: пропущенные системные уровни

«Дом состоит из кирпичей» — это верное утверждение в физике и математике, но неверное в системном подходе. Приведите ещё три примера таких утверждений.

---

[54] https://ailev.livejournal.com/1622346.html — и там много ссылок на работы Джона Дойля и его учеников.

Номер п/п	Утверждение	Пропущенный системный уровень
Пример	Дом состоит из кирпичей	Стены
1		
2		
3		

Системная инженерия: инженерия на основе системного подхода

Наиболее активно после физики, биологии, кибернетики и после этого менеджмента, но до последних наработок по связке физики и биологии, системный подход в 20 веке разрабатывался в **системной инженерии** (systems engineering).

В русскоязычных переводах инженерной литературы менеджеры часто слово engineering не удосуживаются перевести как «инженерия», так и оставляют «инжинирингом». Можно считать, что «системная инженерия» и «системный инжиниринг» синонимы, но есть маленькая проблема: в России почему-то в тех местах, где занимаются инженерным менеджментом, а не инженерией, называют его тоже «системным инжинирингом» — хотя при этом никаких инженерных (т.е. по изменению конструкции и характеристик системы) решений не принимается, речь идёт только об инженерных решениях по поводу системы-создателя как организационной системы. Эти решения в «системном инжиниринге» делаются тоже с использованием системного подхода, но касаются главным образом организации работ команды проекта по созданию **целевой системы** (system-of-interest). Грубо говоря, в оригинальной системной инженерии при рассмотрении проекта создания супер-дупер-табуретки::«целевая система» рассматривали главным образом особенности этой табуретки и особенности методов её создания, а вот в «системном инжиниринге» в его российской версии про саму табуретку говорилось бы крайне мало, зато мы бы узнали всё о том, как ведёт себя команда создателей этой табуретки, как она организована. Это тоже важно, но это системная инженерия организации::«создатель системы», то есть системный менеджмент.

Мы будем считать «инженерию» и «инжиниринг» синонимами, но в случае «инжиниринга» рекомендуем проверять на всякий случай, не менеджмент ли (инженерия организации) имеется в виду вместо инженерной работы с целевой системой (то есть занимаются ли в ходе «инжиниринга» изменением целевой системы, или это делают в ходе ещё какой-то другой «инженерии» рядом с «инжинирингом»).

Старинная инженерия работала с веществом, в котором не было никаких особых вычислений, кроме простейших каких-то «обратных связей» типа регулятора Уатта на паровой машине. Современная системная инженерия работает главным образом с киберфизическими системами, типичными из которых будут роботы, ракеты, аэролайнеры, автономные/беспилотные автомобили.

Уже в двадцать первом веке системные инженеры заговорили о том, что ограничений на вид систем по их уровням организации/эволюционным уровням/системным уровням для системной инженерии нет. То есть и изменение вещества такое, чтобы на выходе появилась мыльница или спичка, и изменение вещества такое, чтобы на выходе случился авиалайнер, и изменение людей (их ведь тоже нужно лечить и учить), и изменение организаций (их нужно проектировать и развивать), и сообществ, и обществ, и даже человечества — все эти изменения

физического мира к лучшему ведутся системными инженерами. Во всех этих случаях речь идёт о многоуровневом взгляде на устройство объектов: и мыльница, и спичка рассматриваются как части чего-то-то целого (объекты в ситуациях использования мыльницы и спички) и в них самих есть какие-то части, но и общество тоже часть целого человечества, и тоже имеет какие-то свои части. Любой труд по изменению физического мира оказался инженерным, причём речь шла именно о наиболее общей и универсальной, безмасштабной/scale-free инженерии — системной инженерии. Международный совет по системной инженерии (INCOSE) послал инженеров за парту учиться социальным теориям/дисциплинам, чтобы быть готовыми к такому повороту событий, а также призвал именно системных инженеров к ответственности за всю Землю: если не они, владеющие системным мышлением, то никто![55].

Жизнь с тех пор показала, что не столько люди-инженеры побежали за парту, чтобы стать политиками как инженерами общества, или менеджерами как инженерами организации (хотя многие стали в том числе и менеджерами). Нет, наоборот: люди, считающие себя политиками и менеджерами, начали работать инженерными методами.

Даже в пропагандистских кампаниях сначала определяют потребности, затем проектируют, потом осуществляют, получают отклик, корректируют проект и методы его реализации, занимаются этим непрерывно, а не разово, при этом ещё и эксплуатируют результаты прошедшей части кампании. Удивительно, но вот такой простой и понятный способ описания работы по изменению окружающего мира и его необходимость были сформулированы в более-менее чёткой форме именно системными инженерами. Простота и понятность, конечно, только относительные: такому взгляду на мир надо было учиться.

Унификация рассмотрения самых разных проектов по изменению мира к лучшему и тем самым резкое увеличение простоты описаний самых разных целевых систем самых разных проектов основывались на идее системы как выделяемом из окружения куске мира, который и нужно изменить (материал превратить в деталь, стажёра превратить в мастера, неиндоктринированное общество превратить в индоктринированное). Далее постулировалось, что этот кусок мира (**целевая система**) в момент его использования наносит миру непоправимую пользу, изменяя его к лучшему (деталь делает работоспособной какое-то «железное» устройство, мастер выполняет полезные работы, индоктринированное общество поддерживает своё существование). Все эти изменения выполняются более-менее одинаково при общем системном взгляде на самые разные системы, несмотря на разницу в терминологии в каждой предметной области.

Скажем, роли менеджера (бизнесмен, организатор с подролями орг-проектировщика и лидера, орг-архитектор, администратор) по отношению к предприятию как системе оказываются примерно теми же, что роли инженера (визионер, разработчик с подролями проектировщика и технолога производства, архитектор, инженер платформы разработки) по отношению к целевой системе, и это позволяет излагать знание о менеджменте много проще, чем подход «с нуля», без использования знаний системной инженерии. Это можно увидеть и на примерах других систем.

Скажем, если личность понимать как набор самых разных мастерств, реализуемых

[55] https://incose-ru.livejournal.com/49015.html

агентом (человеком или даже роботом с AI), то научение человека или робота чему-то новому (например, усилить врождённый интеллект путём образования: обучения мышлению по интеллект-стеку фундаментальных методов) оказывается инженерией личности. И мы видим там те же роли, что и в инженерии «железа» или «софта», что и в менеджменте: культуртрегер (аналогично бизнесмену, визионеру), автор наставлений (аналогично организатору, разработчику, включая подроли методолога как аналога орг-проектировщика и проектировщика, методиста как аналога технолога производства и лидера), тьютор (как аналог архитектора и орг-архитектора, забота об архитектурных характеристиках личности), служба стажировки или деканат (аналог администратора, инженера платформы разработки).

Так что системная инженерия в современном её понимании — это эволюционное изменение мира к лучшему, во всём разнообразии систем разной природы вэтом мире. **Труд** оказался просто разными изводами инженерии самых разных систем. Труд и понимался обычно не только как «тяжкий труд» со стороны трудящихся, но как изменение мира к лучшему со стороны потребителей продуктов труда. Раньше труд был связан с сельским хозяйством, затем инженерным «заводским» изменением на уровне косного вещества, которое не «оживлялось» компьютерами, вспомните «уроки труда» в школе пятидесятилетней давности, где нужно было изготовить табуретку или выточить деталь на токарном станке.

Теперь ограничение на тип систем, ассоциируемых как продукт труда оказалось снято. Труд, практика, деятельность, инженерия, даже культура и стиль как метод/способ выполнения работы (то есть изменения состояния физического мира) — всё оказалось более-менее синонимами, если использовать эти слова безмасштабно, для всех возможных системных уровней. Хотя у каждого слова и есть какие-то свои оттенки смысла, мы их рассмотрим в руководстве чуть дальше по тексту. Подробному рассмотрению труда как поведения по какому-то паттерну (методу/культуре/практике) интеллектуального агента как системы-создателя какой-то целевой системы-продукта посвящено отдельные руководство, по методологии.

В нашем руководстве по системной инженерии мы главным образом будем рассматривать целевую систему, но не системы-создатели в их графе создателей, где одни создатели создают и развивают других создателей, а затем создают и развивают целевую систему (я основываю предприятие по изготовлению табуреток, предприятие изготавливает при помощи подрядчиков производственную линию по изготовлению табуреток, а затем эта производственная линия изготавливает табуретки — это всё перечислены системы в отношениях создания, причём там ведь ещё происходит и непрерывное развитие меня, предприятия, производственной линии предприятия, конструкции табуреток).

Системы-создатели в графах создателей будут рассмотрены подробно в руководстве по методологии, методы создания и развития систем-продуктов (инженерные процессы) в их самом общем виде и связанное с этими методами разделение труда будет рассмотрено в руководстве по системной инженерии. Методы программирования интеллектуальных создателей на выполнение каких-то методов работы (то есть обучение агентов этим методам) — в руководстве по инженерии личности. Системы-создатели из людей и машин (организации) будут рассмотрены в руководстве по системному менеджменту.

При этом руководства по системной инженерии, инженерии личности, системному менеджменту используют владение системным и методологическим мышлением, что позволяет им быть структурированными схожим образом. Один раз рассказывается, как что-то сделать, это всегда в общих чертах будет одинаково (спроектировать, изготовить, эксплуатировать — и повторять это в бесконечном цикле развития) — а дальше можно рассказывать только некоторые особенности, отличающие инженерию систем разной природы, неживых и живых. Один раз выучить, использовать всегда!

Самое современное из по факту уже устаревших определений системной инженерии дано в Guide to the Systems Engineering Body of Knowledge (руководство по корпусу знаний системной инженерии) в 2023 году, «Системная инженерия — это трансдисциплинарный подход и средства создания воплощения успешных систем. Успешные системы должны удовлетворять нужды их клиентов, пользователей и других проектных ролей» [56]. В глоссарии этого руководства даны и другие определения, например определение от почётных членов (Fellows) INCOSE 2019 года: «трансдисциплинарный и интегративный подход создания успешных воплощения, использования и вывода из эксплуатации инженерной системы, использующий системные принципы и понятия, и научные, технологические и менеджерские методы работы»[57]. В этих определениях можно подчеркнуть:

- **Успешные замысливание, воплощение, использование и вывод из эксплуатации инженерной системы в физическом мире, с учётом множественности версий непрерывно развиваемой системы** — это и есть системная инженерия как особый «метод работы»/«вид труда»/деятельности, способ/культура/стиль/практика работы (а в уникальных вариантах, который был выбран для конкретного проекта — стратегия). Слово «успешные» (successful) тут крайне важно, и имеет не бытовое, а терминологическое специальное значение. Термин **«успешный»** означает, что проект учитывает **ролевые предпочтения** (это тоже термин из системного подхода, мы рассмотрим его подробней в следующих разделах руководства) как затрагивающих систему агентов, так и затрагиваемых системой и её проектом агентов. Абстрагируемся пока от того, люди ли эти агенты, или AI, и организованы ли эти люди и AI в какие-то организации, или даже общества, не будем пока разбираться с многоуровневостью агентов по иерархии отношений частей-целых этих агентов. Если ролевые (то есть диктуемые какими-то прикладными методами работы) предпочтения всех этих агентов в ролях заказчиков, плательщиков, пользователей и других (интересы/предпочтения вредоносных ролей, например, воров, учитываются с обратным знаком) учтены, то это и будет «успех». Тем самым успех тут определяется не бытовым, или финансовым, или экологическим, или ещё каким образом, а именно через приемлемость результата проекта для множества агентов-в-ролях, **успех определяется как «мы в проекте договорили всех, все довольны». «Договорили всех»**, кстати, более сильная формулировка, чем **«договорились со всеми»**. Ибо если эти «все»

[56] https://sebokwiki.org/wiki/Introduction_to_Systems_Engineering, SE is a transdisciplinary approach and means to enable the realization of successful systems. Successful systems must satisfy the needs of their customers, users and other stakeholders.

[57] http://sebokwiki.org/wiki/Systems_Engineering_%28glossary%29, A transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.

не договорились между собой, то наша договорённость со всеми разваливается, поэтому надёжней договорить всех (включая себя), то есть заняться также организационной работой, а не только собственно инженерной работой. Собственно, это и было новацией второго поколения системного подхода: агенты-создатели (тогда — люди и их организации, сейчас включаем и AI) как надсистемы целевой системы (потребности от них), так и целевой системы (инженерная команда проекта) не менее важны, чем создаваемая ими целевая система.

- Термин **«системы»** используется в очень специальном значении: это «системы» из системного подхода, а не бытовое слово для «каких-то сложных объектов». Для системной инженерии слово «система» примерно то же, что «физическое тело» для ньютоновской механики — если вы сказали про компьютер «физическое тело», то это автоматически влечёт за собой разговор про массу, форму, объём, потенциальную энергию, модуль упругости, температуру и т.д., но уж никак не цену и не быстродействие самого компьютера. Если вы сказали «система» про компьютер, то это автоматически влечёт за собой разговор про системные уровни, задействованные в работе над созданием и развитием компьютера (транзисторы, микросхемы, печатные платы, компьютер в сборе — от компьютера идём вниз, к частям) и с готовым компьютером (обустройство рабочего места, установка и использование разнообразного софта, от физической границы компьютера вверх, к надсистеме) роли разных агентов и их ролевые предпочтения в важных характеристиках компьютера (concerns), и дальше разговор идёт про концепцию использования, концепцию системы, архитектуру, непрерывную разработку системы и т.д. Все эти понятия будут подробно рассмотрены в нашем руководстве в последующих разделах.
- **Трансдисциплинарный и интегративный/междисциплинарный подход** — системная инженерия как метод претендует на то, что она работает со всеми остальными прикладными инженерными методами (впрочем, не только инженерными). **Трансдисциплинарность** (transdisciplinary) означает внешнесть/«потустороннесть» по отношению к самым разным другим дисциплинам (а в нашем случае ещё и другим методам, основанным на этих дисциплинах/теориях/знаниях), а не нахождение в одном ряду, «между» другими дисциплинами. Трансдисциплинарность — это очень сильное заявление, оно означает, что системная инженерия входит во множество самых разных других прикладных методов, она не «равнопредставлена» с ними, а используется прямо внутри рассуждений в рамках прикладных методов. В INCOSE, откуда взято это определение системной инженерии, ничего не говорят про интеллект-стек, но имеется в виду именно то, что системная инженерия входит как метод в сам интеллект-стек, а не является прикладной дисциплиной за его пределами. Системная инженерия в силу своей трансдисциплинарности может «в одну упряжку впрячь коня и трепетную лань». Например, системная инженерия сопрягает людей и AI в ролях инженеров-механиков, баллистиков, криогенщиков, психологов, медиков, астрономов, программистов и т.д. в проектах пилотируемой космонавтики, которые с использованием системноинженерных понятий координируют свои работы в этих сложнейших проектах. Междисциплинарность подхода системной инженерии как раз и

означает, что использование метода системной инженерии делает возможным объединение труда множества людей, работающих по самым разным прикладным методам. Сегодня становится понятным, что важнейшей характеристикой трансдисциплинарного знания является его безмасштабность как приложимость к системам любого размера/масштаба/системного уровня, от молекул до человечества с его техносферой в целом, а ещё и выход на бесмасштабность во времени, учёт техно-эволюции. И, конечно, трансдисциплинарность подразумевает неантропоцентричность, агентами могут быть не только люди/homo sapience.

- Слово «**воплощение**» (realization, «перевод в реальность») означает буквально это: создание материальной (физической, занимающей место в пространстве-времени, т.е. вещественной/материальной) успешной системы. Речь идёт об изменении физического мира, дело не ограничивается только проектированием и другой информационной работой, проект выходит в физическую реальность и меняет её. При этом, как любят заметить физики, «человек-математик и человек-астрофизик — это тоже вещественные объекты, физические системы». Это же замечание относится и к человечеству в целом: оно вполне физично. Так что инженерам можно ставить задачи по изменению человечества, вместе с его техносферой. Вопрос только в том, можно ли потом эти поставленные задачи выполнить, но в физике это не запрещено, значит о таком можно думать. Например, поставить задачу невымирания человечества на ближайший период от десяти тысяч лет, а там посмотрим.
- **Использование принципов и понятий системного подхода** говорит само за себя. Не используются эти понятия — не системная инженерия, а какая-то другая!
- **Методы работы** системная инженерия берёт не только **научные**, но и **технологические** (полученные методом проб и ошибок), и даже **менеджерские** (но не, например, религиозные). В руководстве по системной инженерии разбирается вопрос, научна или не научная системная инженерия, и там даётся ровно такой ответ: научна и не научна одновременно. С одной стороны, не отвергаются никакие достижения науки, они сильно облегчают жизнь инженерам. Но если к средневековому инженеру придут с просьбой построить мост через речку, а он ответит, что сопромат изобретут через триста лет, и поэтому он не будет строить мост, «это ненаучно»? Это пример Billy Koen, который чётко говорит, что инженерия не сводится к использованию «научных знаний», но сама она развивается вполне в соответствии с тем, как развивается наука: есть догадки по тому, что представляют собой лучшие методы системной инженерии, а затем эти догадки критикуются, и пережившие эту критику догадки как раз и являются современной версией системной инженерии. Методам мышления о методах работы в инженерии («железа», «софта», живых систем, AI, личностей, сообществ — неважно) посвящено руководство по методологии.

По-английски «системная инженерия» — systems engineering, хотя более ранние написания были как system engineering. Правильная интерпретация (и правильный перевод) — именно «системная» (подразумевающая использование системного подхода) инженерия, а не **инженерия систем** (engineering of systems) — когда любой «объект» обзывается «системой», но не используется системный подход во

всей его полноте. Под инженерией систем[58] (например, control systems engineering, manufacturing systems engineering) понимаются обычные инженерные специальности, там легко выкинуть слово «система», которое лишь обозначает некий «научный лоск». Предметные/прикладные (не системные) инженеры легко любой объект называют «системой», не задумываясь об осознанном использовании при этом системного мышления, то есть не используя системный подход и не согласовывая предпочтения самых разных проектных ролей в важных характеристиках системы и характеристиках проекта её создания. В самом лучшем случае про систему предметные инженеры скажут, что «она состоит из взаимодействующих частей» — на этом обычно разговор про «систему» и «системность» заканчивается, он не длится больше двадцати секунд, понятие «система» тут означает примерно то же самое, что «система» в классической физике. Занимающиеся «инженерией систем» очень полезны и нужны, но они не системные инженеры.

А вот из системной инженерии квалификатор «системный» без изменения смысла понятия выкинуть нельзя. Неформально определяемая **системная инженерия — это инженерия с системным мышлением в голове** (а не любая инженерия, занимающаяся объектами, торжественно поименованными системами просто для добавления указания о сложности этих объектов и научности «как в физике» в их описании).

Справедливо будет сказать, что любая инженерия, которая начинает опираться на владение мышлением по интеллект-стеку, становится системной инженерией. Это верно даже по отношению к инженериям, традиционно таковыми не считаемым (те же менеджмент, или медицина, или образование детей). Все они по факту становятся «системными», если ими занимаются с задействованием всего интеллект-стека, даже если в явном виде слово «системный» к этим инженериям не приписывать и слова «инженерия» не говорить. Вопрос только в том, насколько современная версия как системного подхода, так и остальных положений из фундаментальных теорий/дисциплин методов в составе интеллект-стека используется в каждом конкретном случае. Два разных врача могут использовать очень разные версии интеллект-стека, поэтому один может быть едва системен и рационален в своей работе (по факту это будет не врач, а знахарь), а другой проявлять и рациональность, и системность на уровне не хуже специально обученного системного инженера киберфизических систем.

Интегральность/целостность (полнота охвата всех частей целевой системы согласованным их целым, многоуровневое разбиение на части-целые), трансдисциплинарность (использование самыми разными дисциплинами рассуждений системной инженерии), безмасштабность — это ключевое, что отличает системную инженерию от какой-то прикладной инженерии.

При этом из инженерии уже успела исчезнуть отдельная роль системного инженера: как врача-гинеколога и врача-дантиста сегодня не путают, но признают, что они врачи, так и у системных инженеров не путают разработчиков (с подролями проектировщика и технолога производства) и архитекторов, и даже признают, что у них «продуктивный конфликт ролей». Но принято говорить, что самые разные роли в инженерии (она вся по факту системная сегодня) считают ролями

[58] см. проф. Derek Hitchins, «Systems Engineering vs. Engineering of Systems — Semantics?», http://systems.hitchins.net/profs-blog/systems-engineering-vs.html

«системных инженеров», если агенты в этих ролях занимаются всей системой в целом в разбиении на много уровней вниз и вверх от границы системы, а не только отдельными частями системы или только отдельными прикладными инженерными (теплотехника, электротехника) или прикладными менеджерскими (операционный менеджмент, орг-проектирование и лидерство в отличие от полноценной инженерии организации) методами работы.

Системная инженерия как отдельный метод/способ работы с «системой в целом» поначалу применялась главным образом для борьбы со сложностью аэрокосмических проектов, и она была там крайне эффективна. Для того, чтобы маленький проект уложился в срок и бюджет, нужно было на системную инженерию потратить 5% проекта, что предотвращало возможный рост затрат проекта на 18%. Для средних проектов на системную инженерию оптимально тратить было уже 20% усилий всего проекта, но если не тратить — возможный рост затрат проекта был бы 38%. Для крупных и очень крупных проектов оптимальные затраты на системную инженерию оказались 33% и 37% соответственно, и это для того, чтобы предотвратить возможный рост затрат проекта на всяческие переделки плохо продуманного 63% и 92% соответственно[59]. Сейчас системная инженерия не выделяется отдельно от прикладной инженерии, ибо от всех прикладных инженеров тоже требуют понимания того, что происходит с системой в целом, их тоже учат системной инженерии.

Системная инженерия с её методами/культурой документирования всех решений по координации труда самых разных агентов в простых небольших проектах «на одного человека» почти не даёт эффекта (там всё хорошо продумывается «в уме» и не требует особых методов мышления, не требует многочисленных согласований важных характеристик системы и проекта). Но в сложных и очень крупных проектах системная инженерия оказывается необходима: без системного мышления в таких коллективных больших проектах допускаются ошибки, которые потом оказывается очень дорого переделывать. Без системного мышления согласно упомянутым в предыдущем абзаце исследованиям сталкиваться со сложностью выйдет чуть ли не вдвое дороже за счёт дополнительной работы по переделкам допущенных ошибок.

В результате системным инженерам, которые догадались в инженерных проектах использовать системное мышление ещё в прошлом веке, удалось выполнить сверхсложные проекты — например, они в 1969-1972 году отправили на орбиту вокруг Луны 24 космонавта, а по самой Луне пешком ходили 12 человек[60]. Да что там пешком, рекорд скорости по Луне на луномобиле составил 18.6 км/час, при этом люди уезжали от ракеты на Луне на расстояние больше 7 километров!

Достижения современной космонавтики, думаю, тоже не нужно рекламировать, даже с учётом того, что инженерное развитие в этой области было существенно искажено военными проектами, а инженеры развращены государственным финансированием. Сложность космических проектов не позволяла добиваться успехов «обычной инженерией». Так, советская школа инженерии не смогла повторить достижений лунной программы системных инженеров NASA, не смогла повторить многих и многих достижений планетарных программ, которых достигли в NASA. Конечно, у отечественной космонавтики есть и отдельные достижения (например, удачные ракетные двигатели), но при росте сложности проекта в целом

[59] Цифры из https://yadi.sk/i/jBxY4Ny5w7j09g
[60] https://en.wikipedia.org/wiki/Apollo_program

неудачи начинают резко перевешивать достижения — типа четырёх подряд неудач лунного старта Н-1[61].

Тут нужно отдельно оговорить, что всё это были достижения ещё первого поколения системного мышления, когда не обращали внимания на успешность системы как удовлетворения предпочтений в важных характеристиках системы и проекта её создания для самых разных проектных ролей. Тогда не обращали явного внимания на системы-создатели надсистемы и их потребности, на системы-создатели целевой системы. В учебниках системной инженерии прошлого века это всё начало появляться только в 80х годах, когда были сформулированы основные положения системного подхода второго поколения.

Космические программы имели астрономические бюджеты, и критиковались за то, что вместо помощи больным и голодным людям деньги выкидывались на удовлетворение каких-то политических амбиций (это было верно и для США, и для СССР, поэтому лунные старты и были прекращены на десятки лет!). В нашем руководстве по системному мышлению будет подраздел о том, почему государственные проекты не могут быть успешными по критериям самой системной инженерии.

Тем не менее, технический успех (работоспособность сложных технических систем, если не обращать внимания на цену, заплаченную налогоплательщиками за эту работоспособность) в аэрокосмических программах США был поразительным.

Метод работы западных аэрокосмических инженеров — именно системная инженерия, т.е. инженерия с использованием системного мышления. Системные инженеры (и отчасти программные инженеры) уточняли и развивали положения системного подхода, проверяя их действенность в сложных проектах, а самое важное из этих уточнённых и обновлённых положений попало в международные инженерные стандарты.

По иронии судьбы, стагнация системной инженерии от государственных и военных проектов наблюдается и прямо сейчас. Так, на международном симпозиуме INCOSE в 2021 году собралось много системных инженеров из военных и государственных проектов, и демонстрировались умеренные инженерные достижения. Но не было никаких докладов от SpaceX, хотя фронтир системной инженерии в аэрокосмосе демонстрирует сегодня именно эта фирма. Системная инженерия перестала развиваться в этой профессиональной организации, состоящей по факту из чиновников-инженеров. Развитие системной инженерии происходит в реальных коммерческих проектах, и часто не носит имя «системная инженерия». Системное мышление развивается в таких проектах, как постепенно становящиеся автономными/беспилотными автомобили Tesla и роботы Tesla Optimus, инфраструктура быстрого космического интернета StarLink от SpaceX, суперкомпьютеры для искусственного интеллекта от NVIDIA и Google.

В отличие от многих и многих «академических» (университетских) вариантов системного подхода, «системноинженерный вариант» из «реального сектора экономики» в начале 21 века был проверен тысячами сверхсложных проектов, обсуждён десятками тысяч инженеров, унифицирован и доказал свою эффективность на деле. Он не имеет авторства (ибо в его создании участвовало множество людей), он не является «оригинальным исследованием», он не

[61] https://ru.wikipedia.org/wiki/H-1

изобретает велосипеды в части самого системного подхода. Он просто отражает всё самое важное, что было накоплено системным движением за десятки лет и оказалось практичным и относительно легко применяемым на практике десятками тысяч людей, исполняющих инженерные роли.

Подробней по методам/практикам/стилям/культуре системной инженерии будут наставлять руководство по системной инженерии. Осваивать системную инженерию надо после освоения системного мышления и методологии:

- Наше руководство по системному мышлению наставляет безмасштабной эволюционной версии системного мышления как выборке приёмов мышления, основанных на системном подходе из самых разных фундаментальных мыслительных дисциплин, в том числе системной инженерии.
- Метод/практика/культура/стиль/способ/стратегия работы — это сложное понятие. Руководство по методологии наставляет в том, как вообще описывать какие-то работы в аспекте их метода, отвечая на вопрос «как делать».
- Руководство по системной инженерии наставляет по самым общим методам изменения физического мира (инженерным процессам и связанным с ними разделением труда по различным ролям): методам разработки концепции использования и концепции системы, выработки архитектурных решений, детального проектирования и технологической (то есть инструментальной) подготовки производства, изготовления, инженерных обоснований, эксплуатации, и всё это в непрерывном развитии/эволюции созданной однажды в виде MVP (minimal viable product) системы.

Наше руководство по системному мышлению основано главным образом на версии и терминологии системного мышления, принятой в системной инженерии киберфизических (включающих как «железо», так и «софт») систем и отчасти в программной инженерии (software engineering) и менеджменте как организационной инженерии. Именно этот вариант системного мышления более всего ориентирован на методы работы людей (а теперь уже и на человеко-машинные, включающие AI), на труд по изменению окружающего физического мира к лучшему, а не просто на «понимание», «исследования», «анализ», «науку», то есть изменение моделей::описаний окружающего мира. **Анализ-понимание полезен только в контексте последующего синтеза-созидания или изменения чего-то в нашем физическом мире, в контексте изменяющего физический мир к лучшему труда по созданию новых и модернизации уже имеющихся систем.**

Наше руководство наставляет на трудовое/практическое/деятельностное/«по определённому методу» системное мышление, которое используется прежде всего в инженерном процессе, в системной инженерии. Это неважно, что именно является предметом инженерного проекта: «железная» система, софт, живая система (скажем, марикультура как инженерия, помним, что культура/инженерия/деятельность/метод работы — это синонимы), организация, личность.

Не путайте наше руководство по системному мышлению с руководством по системному анализу! Системный анализ — это маленький кусочек системного мышления, связанный с моделированием систем! **Системное мышление имеет**

направленность на преобразование мира к лучшему, а не просто на описание/моделирование мира! Оно прежде всего синтетично, а не аналитично! Анализ в системном мышлении подчинён синтезу!

Наше руководство представляет тот вариант системного мышления, который изначально ориентирован на создание успешных систем (помним о специальном смысле слова «успешные»!) — будь это «железные» системы (самолёт, атомная электростанция), программные системы, биологические системы (клетки и организмы — ими занимается системная биология, генная инженерия), системы-личности, системы-предприятия (организационные системы), или даже такие нестандартные системы как танцор или соревнование по марафону. Это и есть задача нашего руководства: наставить инженера-менеджера, как более-менее одинаково думать о самых разных по природе системах.

Ещё наше руководство применимо к работе с системами по линии безмасштабности. Так, для систем из людей можно говорить и про сообщества (клубная работа: клиентура какой-то фирмы, ассоциация каких-то профессионалов), общества (в том числе и государственное строительство, и вопросы войны и мира) и человечество в целом (изменение его таким образом, чтобы снизить экзистенциальные риски вроде гибели от мощного извержения вулканов или падения астероидов, а то и просто обеспечение биологического бессмертия, почему бы и нет). На этих самых разных уровнях масштаба (размеров в физическом мире) и эволюционной сложности (число и характер частей по линии молекулы — макромолекулы — организм — разумный организм — организация разумных агентов и их инструментов — сообщество — общество — человечество) мы будем использовать более-менее одинаковые паттерны мышления, учитывая при этом особенности систем каждого уровня и особенности терминологии, принятой в прикладной инженерии каждого из этих уровней. Многочисленные примеры этого будут представлены в руководстве по системной инженерии, там первые несколько разделов как раз посвящены эволюции, в том числе рассматривается вопрос о неизбежном росте эволюционной сложности систем.

Определение системной инженерии в версии 2019 года SEBoK сопровождается фразой **«We use the terms “engineering” and “engineered” in their widest sense: “the action of working artfully to bring something about”. “Engineered systems” may be composed of any or all of people, products, services, information, processes, and natural elements»**, то есть объектом инженерии является something/«что угодно», и авторы определения системной инженерии включают в это «что угодно» даже информацию и процессы, при всей неопределённости этих понятий в их связи с физическим миром. Мы в нашем руководстве наставляем, как следовать этим указаниям на всеохватность системного мышления и системной инженерии, не теряя связи с изменением физического мира к лучшему через создание успешных систем.

Главное — это понимать, что мы тут участвуем в эволюции, которая по факту является процессом оптимизации состояния мира, уменьшающим неустроенности/frustrations, вызванные конфликтами взаимодействий устойчивых объектов разных системных уровней. Системное мышление заключается в том, чтобы понимать этот источник «вечной неустаканенности», «вечных проблем» и предлагать оптимизационные решения (заниматься **системным творчеством**: находить противоречия и предлагать новые механизмы их снятия, об этом будет отдельный раздел руководства). Обычно проблемы не могут быть решены на одном

системном уровне из-за этих межуровневых конфликтов. Мир устроен сложно, поэтому для решения проблем (решения задачи оптимизации в многоуровневых конфликтах, чем занимается дарвиновская эволюция в живых системах и техно-эволюция в инженерном мире) нужно задействовать экономящее вычисления системное мышление, чтобы хоть как-то уметь предсказывать будущие состояния этого сложного мира. Системное мышление даёт возможность думать о наиболее важном, то есть о системах, а не обо всех подряд деталях непрерывного мира, о каждой точке пространства — нет, думать о каких-то частях пространства в целом, то есть о системах в их границах как целых, и отношениях систем. Системное мышление квантует непрерывный мир, бесконечное число потенциальных объектов для мышления сводит к весьма ограниченному числу объектов, то есть системам. Про то, как именно системное мышление борется со сложностью, будет рассказано в следующих разделах нашего руководства.

Вопросы. Успешность системы.

Что означает «система успешна»?

--- За неё заплатили разработчикам, она успешна на рынке ### Нет, термин «успешность» в системном мышлении не означает «рыночную успешность» (хотя рыночная успешность у успешной системы вполне может быть).

+++ Проект создания системы учитывает ролевые предпочтения как всех затрагивающих систему агентов, так и всех затрагиваемых системой и её проектом агентов. ### Да, именно так — для успешности системы нужно учесть как предпочтения всех обнаруженных агентов, кто затрагивает эту систему и её создателей, так и предпочтения всех обнаруженных тех, кто затрагиваем системой и её создателями. И обязательно затрачиваются усилия на обнаружение этих агентов, никто не ждёт, что они как-то сами проявятся!

--- Инженерная работа была закончена вовремя, превышения бюджета не было, акты приёмки-сдачи подписаны. ### Нет, успешность системы понимается не как классическая успешность выполнения менеджерского проекта. Для успешности системы мало выполнить предпочтения менеджеров «в срок, в бюджет, с надлежащим качеством», нужно учесть много больше самых разных предпочтений! Если вы украли землю, и на ней построили здание «в срок, в бюджет, с надлежащим качеством», такой проект нельзя будет назвать успешным — не учтены предпочтения владельца земли!

Системность — это не систематичность

В системной инженерии, которая использует системное мышление как основной для себя метод мышления, подчёркивается, что она одновременно **системна** (systemic) и **систематична** (systematic).

Под **системностью** в науке и инженерии обычно понимается следование системному подходу — **специальному фундаментальному/трансдисциплинарному (внешнему по отношению к любой прикладной предметной дисциплине, но глубоко проникающему в мышление этой прикладной/предметной дисциплины) набору типов/понятий/концептов и их отношений, который направляет внимание инженера на главное, о чём нужно подумать в первую очередь для успешного изменения мира: «система», «системный уровень»,**

«неустроенность», и др.

Систематичность — это совсем другое, это просто честное формальное (не содержательное) последовательное обращение внимания на то, на что внимание нужно было обратить. Если вы «формально» (то есть без учёта содержания) не пропускаете во внимании и действии какие-то объекты, которые должны затронуть, то вы систематичны. Дисциплину (во всех смыслах этого слова — и предметную, и административную, «следование алгоритму выполнения действий с объектами предметной области») любого метода нужно чтить, систематичность об этом. Надо честно/систематично выполнить все трудные мыслительные ходы, систематично/без пропусков выполнить рекомендуемые рассуждения, чтобы была уверенность в получении результата. А если речь идёт ещё и о действиях в физическом мире, то нужно честно выполнить и их тоже.

Если требуется заполнить каждую клеточку таблицы 10*10, то нужно сосредоточиться, собраться — и систематично (то есть полностью без пропусков, аккуратно и без описок, в заданном порядке) заполнить. Это не системно, но это систематично. Это занудно и требует времени. Систематичность вовсе не подразумевает системности. Можно систематично поставить крестик на каждом пункте какой-то абсолютно бессистемной инструкции, и выполнить все её безумные предписания, не пропустив ни одного пункта, причём доведя это дело до конца, но системности никакой в этом не будет, то есть не будет использования понятий системного подхода.

Системность — это про содержание мышления, про понятия/концепты системного мышления. Это содержание чеклиста, который определяется системным подходом, и который нужно прочекать своим мышлением **—** обратить внимание на концепты системного подхода. Системность в том, что надо найти целевую систему среди всех систем, найти надсистему, найти систему создания и даже множество создателей, увязанных в граф по отношению создания и развития (изменения состояний) друг друга. Если вы абсолютно несистематично, но хаотично это делаете — вы всё равно системны!

Систематичность — это про документацию и предметы окружающего мира, т.е. про то, что ни один пункт какого-то (возможно, совсем не системного) чеклиста не будет пропущен, все намеченные работы выполнены, результаты этих работ документированы.

Систематичность сразу подразумевает огромное количество всякой (как содержательно нужной, так и содержательно ненужной, хотя это «содержательно ненужное» инженерам может быть «административно нужным» каким-нибудь бюрократам из соседней оргсистемы) умственной работы, причём документируемой. Систематичность подразумевает вгрызание в какие-то детали с упорством маньяка. Системность повышает вероятность, что это важная работа (вы работаете с важными объектами, внимание на которые направляется системным подходом), а не сумма важной и зряшной (вы работаете со всеми подряд объектами, некоторые из них оказываются важными). Системность и систематичность, тем не менее, живут рука об руку, они в балансе[62].

---

[62] https://incoseuk.org/Normal_Files/WhatIs/Systems_Engineering, https://www.researchgate.net/publication/327073164_Envisioning_Systems_Engineering_as_a_Transdisciplinary_Venture

Важность систематичности относится не только к методу системного мышления. Систематичность нужна для любых деятельностей. Человек, занимающийся системным фитнесом на основе соматомеханики[63] может сколько угодно «знать головой», как налаживать себе роскошное подвижное тело. Но если он систематично не будет делать предписанных методом «пяти подходов в упражнении по 30 секунд каждый», так же систематично не определив перед этим, какое место в теле у него наиболее проблемное, чтобы эти пять подходов наносили непоправимую пользу, то это не будет означать, что он практикует системный фитнес как инженерный метод/практику. Он может даже будет системен (в мышлении использовать понятия системного подхода и соматомеханики), но не систематичен: будет пропускать важные действия в мышлении, важные действия в физическом мире. **Для получения результатов в проектах любого сорта нужна не только системность, но и систематичность. Одновременно. Дисциплина мышления и дисциплина действия. Чекнуть предписанное системным мышлением — сработать с заданными системным мышлением типами объектов (системность), чекнуть всё предписанное методами работы без пропусков (систематичность).**

Опора на стандарты и публичные документы системной инженерии и менеджмента в нашем варианте системного подхода

Вариант системного подхода, которому мы наставляем в нашем руководстве, основан главным образом на материале **стандартов** и **публичных документов** системной инженерии. К инженерным стандартам мы добавили некоторое количество стандартов менеджмента как инженерии предприятий. Публичный документ — это стандарт, для которого не определена процедура проверки соответствия стандарту. А так всё остальное совпадает: спецификация выпускается каким-то коллективом, регулярно пересматривается. Публичные документы служат для целей информирования, распространения каких-то знаний, а не для унификации. Это главным образом материалы второго поколения системного подхода, «целевые системы, которые замышляют, изготавливают и эксплуатируют системы-создатели из команд людей».

Конечно, мы добавили современные идеи системного подхода третьего поколения, которые появились в последние пять лет в физике, биологии, а также инженерии систем управления/control systems engineering, об этих идеях мы писали в предыдущих разделах (безмасштабность, включая не только шкалы/масштабы/scales размера, но и шкалы/масштабы/scales времени, в том числе эволюция дарвиновская/биологическая и техно-эволюция, неантропоморфность в изложении, конфликты между объектами разных системных уровней и неустроенности).

Тем не менее, основные понятия и терминология была взята из инженерной и менеджерской версий системного подхода, а не из академической литературы. Из этих стандартов и публичных документов мы брали основные понятия и их терминологию, и только чуть-чуть адаптировали эти понятия и терминологию так, чтобы была очевидна связь друг с другом понятий, определяемых в различных

---

[63] Цепочка текстов «Системный фитнес», https://ailev.livejournal.com/1429126.html, в 2021 году системный фитнес как инженерный метод был переформулирован на основе понятий соматомеханики, ибо в основе фитнеса как телесной инженерии лежит механика (понимаемая как механика в физике: статика и динамика) сомы (человеческое тело, ощущаемое изнутри самим человеком, а не внешним наблюдателем).

стандартах и публичных документах.

Опора на стандарты важна в том числе и потому, что сами стандарты и публичные документы регулярно, раз в несколько лет, пересматриваются. Это позволяет не отстать от жизни, как на десятки лет уже отстали тексты общей теории систем Берталанфи (ОТС), которые во множестве можно найти в книжных магазинах и в интернете даже сегодня (помним, что эти идеи впервые были предложены в 30-х годах прошлого столетия). Когда-то устареет и излагаемый нами в руководстве вариант системного подхода (и такое уже было! Текущий текст — это результат десятой переписки!), но при опоре на регулярно пересматриваемые стандарты и публичные документы, а также регулярно выходящие научные статьи это можно будет заметить (как заметили мы относительно недавний переход от «междисциплинарности» к «трансдисциплинарности» в публичных документах системной инженерии, недавнее открытие роли неустроенности от конфликтов между системными уровнями как основы роста сложности в биологической, меметической и техно-эволюции, уход от инженерии требований в системной инженерии, уход от понятия жизненного цикла к «непрерывному всему»).

В нашем варианте системного подхода были использованы следующие версии инженерных и менеджерских стандартов и публичных документов (этот список далеко не исчерпывающий, приведены лишь главные источники[64]):

- Стандарт **ISO/IEC/IEEE 15288:2023 Systems and software engineering — System life cycle processes** задаёт само понятие системы и жизненного цикла, различает целевую систему, системы в окружении и системы создания, вводит понятие методов/процессов/практик жизненного цикла. Хотя он существенно отстал от жизни, ибо подразумевает главным образом однократную разработку, включает методы инженерии требований, что уже неактуально в инженерии (неактуальным это стало примерно с 2017 года, но международные стандарты такого уровня меняются медленно). Но мы всё-таки используем многие понятия, которые введены этим стандартом.
- Обобщенный с исключительно рекомендаций по созданию архитектурной документации до полного документирования системы стандарт **ISO/IEC/IEEE 42010:2022 Systems and software engineering — Architecture description** привносит множественность описаний и деятельностный подход. Это «поворот мозгов» от редукционистского подхода одностороннего описания к системному подходу, подразумевающему множественность связанных описаний, находящихся в различных информационных системах.
- Обобщенный от программной до системной инженерии стандарт **OMG Essence 2.0:2024 — Kernel and Language for Software Engineering Methods** задаёт язык для описания метода создания и развития системы в

[64] Для многих из этих стандартов есть русскоязычные их официальные варианты в виде ГОСТ, но мы на них не опирались. Во-первых, нас больше интересуют международные, а не национальные стандарты. Мы надеемся, что наш учебник будет использоваться не только в России, и полученные из него знания будут универсальными для разных стран. Во-вторых, переводы международных стандартов для целей их «гостирования» выполняются в порядке хозяйственных договоров без особого внимания к их качеству и гармонизации использованной в разных международных стандартах терминологии. Поэтому мы не используем термины, определяемые переводными ГОСТами. В-третьих, международные стандарты непрерывно пересматриваются, и переводы обычно отстают от текущего содержания стандартов, они доступны только для прошлых неактуальных версий.

терминах методов/практик работы. Этот стандарт также вводит чеклисты/контрольные вопросы. Мы используем из этого стандарта не столько его Kernel (набор основных альф), сколько его Language (само понятие альфы как важного объекта, для которого в проекте создания и развития системы отслеживается состояние).

- Стандарт **IEC 81346-1:2022 Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 1: Basic rules** используется для минимально необходимого описания структуры сложных инженерных объектов, задавая принципы обозначения систем и их частей. Это фундамент для управления конфигурацией в ходе жизненного цикла. Кроме того, этот стандарт различает три вида описаний: функциональное (functional), продуктовое (product) и мест (location), хотя и не затрагивает необходимость стоимостного (cost) описания, уже обязательного для сегодняшнего системного мышления.
- Стандарт **ISO 15926-2:2003 Industrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model** служит для моделирования данных развёрнутых (полных) описаний инженерных объектов. Поддерживает интеграцию данных различных информационных систем жизненного цикла инженерных объектов. Нам он важен для введения понятий 4D экстенсионализма, что экономит мышление за счёт преобразования большого числа отношений между объектами в отношения часть-целое, если объекты эти представлять не как объекты в пространстве (3D), но как объекты в пространстве-времени (4D).
- Публичный документ **NIST PWG Cyber-Physical Systems (CPS) Framework Release 1.0 (2016)** уточняет способы описания для киберфизических систем, вводит классификацию аспектов для важных характеристик систем/предметов интереса проектных ролей.
- Публичный документ **Guide to the Systems Engineering Body of Knowledge (SEBoK):2023** даёт нам определение успешной системы и множество других определений системного подхода.

Повторим: большинство этих стандартов и публичных документов ещё не учитывают изменений последних пяти лет в методах современной системной инженерии и других методов в составе интеллект-стека и тяготеют к использованию денег налогоплательщиков в госпроектах, так что наше руководство не во всём следует этим стандартам. Так, в стандартах повсеместно даётся отсылка к жизненному циклу системы как однократному прохождению работ по методам/«рабочим процессам» жизненного цикла. В современном системном мышлении вместо «жизненного цикла» используется концепция **«непрерывного всего»/continuous everything и эволюционного инженерного процесса**: методы создания и непрерывного развития системы как постоянное создание новых версий системы, а не однократное создание системы, проводящее её «от рождения до смерти», как в биологии. Наше руководство учитывает этот переход к методам эволюционной инженерии, «непрерывного всего» в инженерии.

Мы гарантировали универсальность нашего варианта системного мышления тем, что на деле использовали его в самых разных проектах — инженерных, менеджерских, старта новой деятельности, педагогических, культурных,

искусственного интеллекта, исследований, и т.д. (речь идёт не только о проектах, которые были выполнены системными инженерами мира, но и о проектах авторов руководства, и о проектах инженеров-менеджеров, освоивших системное мышление по многочисленным предыдущим версиям руководства). Руководство с его современным вариантом системного мышления — это не результат только академической работы с текстами, это результат многочисленных опробований предлагаемого в нём варианта системного мышления в реальных проектах.

Основные понятия системного подхода

Вот основные концепты/понятия варианта системного подхода, описываемого в нашем руководстве. Именно эти понятия будут подробно описаны в последующих разделах руководства, и к понятиям собственно системного подхода добавлены тут и некоторые другие понятия из интеллект-стека:

Привязка к физическому миру (эти понятия в основном взяты из семантики, теории понятий, онтологии — но часть тут просто специализация для системного мышления):

- физический объект, занимающий место в пространстве-времени
- воплощение (физические объекты) против их описаний и документов
- изменения (процессы, проекты, кейсы) как физические объекты
- события как физические объекты
- ролевой/функциональный объект (роль) и конструктивный/материальный объект (конструктив) как его аффорданс/подходяшка, их физичность
- софт как физический объект (исходный код как описание софта), игровая сессия как физический объект, личность и мастерство как физические объекты, и прочие «неклассические системы», включая предприятие/оргзвено как физический объект
- части во времени
- методологическое время против времени жизни системы
- при совпадении места/объема двух объектов в пространстве-времени — это один объект (экстенсионализм)
- отношение состава (composition, «часть-целое», разбиение) физических объектов

Методологическая/функциональная/деятельностная субъективность описания системы (эти понятия взяты главным образом из методологии как учения о методах работы агентов, в том числе актёров, играющих «роли в методе работы» функциональные/профессиональные/деятельностные/трудовые/организационн ые/практические/культурные/стилевые роли, синонимия тут огромна):

- метод/способ работы, метафора ролевой игры
- функциональная/трудовая/проектная/деятельностная/культурная/орг роль (stakeholder role): внешняя, внутренняя/командная
- ролевые: «предмет интереса»/«важная характеристика», интерес/предпочтение
- агент, его намерение и отыгрывание метода работы роли как актёра
- успешная система

Системное разбиение:

- системы системного подхода против систематики («система Линнея») и норм/правил («система Станиславского»)

- система, системный уровень, системное разбиение
- эмерджентность/системный эффект
- виды систем: целевая, наша, подсистема, надсистема, окружение (системы в окружении), создатели (системы создания), граф создателей
- имя системы (ролевое, по функции)
- чёрный и прозрачный ящики
- концепция использования, концепция системы
- инженерные обоснования

Описание и документация системы:

- описание (definition) системы
- документация системы (description)
- ролевое/частное описание (view)
- ролевой метод описания (viewpoint)
- модель, мета-модель, мета-мета-модель, мульти-модель, мега-модель
- прожекторный и синтетический подходы к описанию систем

Функциональные и конструктивные части системы, размещения и ресурсы:

- разбиения: функциональные, конструктивные/материальные/модульные/продуктовые/аффордансные, размещения/пространственные, стоимостные (кандидат: работы)
- конфликты между системными уровнями и неустроенности
- функциональная часть (роль), её порты, потоки/связи
- конструктивная/материальная часть (конструктив), его интерфейс, платформа
- пространственная часть (размещение)
- ресурсная часть (совокупная стоимость владения)
- архитектура
- функциональный анализ и конструктивный/модульный синтез (изобретение)

Этот набор понятий/концептов системного подхода, развивавшийся много лет в рамках самых разных фундаментальных методов мышления, удивительно компактен: сложнейший мир самых разных ситуаций представляется относительно небольшим числом понятий, а сам набор этих понятий выбран так, чтобы мир представлялся менее сложным, чтобы о мире было проще мыслить, лучше понимать причинно-следственные связи и источники проблем.

Руководство в последующих разделах подробно описывает эти понятия и связи между ними как объекты и отношения предметных областей методологии и системного мышления, последний раздел руководства даёт компактное (без обоснований и подробных объяснений) описание этих понятий в рамках третьего поколения системного подхода со ссылками на литературу (есть и опубликованный перевод этого описания на английский[65]). Именно на эти понятия опираются прикладные инженерные (включая инженерию уровней организации, сообщества, общества) рассуждения, когда говорят об их опоре на системный подход и рассмотрение методов ведения проектов по созданию успешных систем.

[65] https://arxiv.org/abs/2310.11524

Терминология

Терминология[66] — это учение/знание/дисциплина о словах, которыми обозначают **понятия.** Concepts, «понятия» и «концепты» для нашего руководства вполне могут выступать терминами-синонимами, хотя в некоторых научных школах русские слова «концепт» и «понятие» означают разное, да и по-английски широкое понятие иногда называют notion[67]. «Терминология» тоже имеет два терминологических значения:

1. набор слов-терминов для выражения типов и отношений онтологии какой-то предметной области и
2. Учение/дисциплина не о самих понятиях, а именно об обозначениях понятий словами, о **терминах**.

Это так же, как с геометрией и логикой — и конкретные версии геометрии и логики, и учение о том, какие вообще бывают геометрии и логики.

В каждом языке сформировались (или продолжают формироваться) наборы терминов (терминологии в первом значении) для понятий из разных областей человеческой культуры/деятельности/методов работы. И в этих областях термины приобретают значения, т.е. обозначают (они ведь «знаки», поэтому «обозначают») какие-то находящиеся в реальном мире, а не мире знаков, объекты и их отношения. Терминология тем самым может считаться частью семантики как общего учения о соотношении объектов, их обозначений и понятий/концептов этих объектов. Это уже затрагивалось в руководстве по рациональной работе, а также будет дополнительно разъяснено в руководстве по интеллект-стеку.

Термин — это всегда только слово. Слова нам нужны только для того, чтобы договориться о концептах/понятиях, сами слова не важны. Так, совершенно неважно, используете ли вы термин «концепт» или термин «понятие», если имеете в виду одно и то же: значение того, что обозначается термином. Диалектов много, споры о том, какой из диалектов и есть «настоящий язык» (переводить ли viewpoint в системном мышлении как «точка зрения», «метод описания», так и говорить «вьюпойнт», или всё-таки переводить «аспект», как будет «**единственно правильно»**?!) — они бесплодны, в этих спорах не договориться.

Споры между людьми по самым важным вопросам жизни и смерти оказываются часто всего-навсего «спорами о терминах»: один и тот же объект называется по-разному в разных диалектах и люди считают, что речь идёт о разных объектах (в философской литературе приводится пример Венеры: в одних странах её называют «утренняя звезда», а в других — «вечерняя звезда»), или наоборот — одинаковые слова означают совсем разные объекты («косил косой с косой косой косой на косе»).

В таких «спорах о терминах» важно уметь распознать, что это не содержательный разговор, а всего лишь спор о терминах. Чтобы прекратить такие бесплодные споры, надо формулировать в речи свои представления о мире как ожидаемые наблюдения, а не через терминологию из вашего любимого диалекта, даже если она берётся из вашего любимого стандарта или словаря. У вашего собеседника может оказаться любимым другой стандарт или словарь, он может говорить на другом неведомом вам диалекте — и вы не договоритесь по существу вопроса, просто застрянете на обсуждении слов. Так, если вы попытаетесь поговорить с кем-

[66] https://ru.wikipedia.org/wiki/Терминология
[67] https://www.merriam-webster.com/dictionary/notion

то на тему интеллекта или танца, то вряд ли договоритесь: каждый человек толкует «интеллект» каким-то своим способом, равно как и «танец». Примерно будет понятно, что речь идёт о какой-то «умности» или «движениях под музыку», но подробно ничего обсудить не удастся: застрянете на долгом обсуждении того, что вы понимаете под «интеллектом» (словом!) и что понимает под «интеллектом» (словом!) ваш собеседник, равно как то же самое будет со словом «танец». Люди признают, что в словаре у каждого слова есть множество словарных значений, но потом настаивают, что вам использовать нужно только то значение, которое предпочитают использовать они сами — а от остальных значений слова вам лучше бы воздержаться, или придумать для них какое-то новое слово. Использование самих терминов, если об их словарных значениях ещё не договорились, будет в разговоре бесполезным, сведётся к вечному бесплодному спору о терминах. Но если спора не будет, а значение слова будет воспринято по-другому, то вы получите «ложного друга переводчика» с профессионального сленга на бытовой — и дальше неустранимые и труднонаходимые ошибки в мышлении.

Если вы чувствуете, что не можете договориться по какому-то простому термину, попробуйте табуировать его употребление, то есть продолжайте разговор без использования этого термина, просто чтобы не путаться. Ибо вы не сможете переубедить собеседника в том, что используете для какого-то слова другое словарное значение. В нашем руководстве мы поступили так со словом «стейкхолдер»/stakeholder, словом «танец», словом «обеспечивающий»/enabling. Использование этих слов часто вызывало ошибки. Например, «обеспечивающая система» воспринималась не как система-создатель, обеспечивающая воплощение системы в физическом мире, но как что-то типа «службы снабжения» — система электропитания, кейтеринга, заправочная станция, поставка запчастей. Это невозможно было перешибить никакими разъяснениями, ошибки были чуть ли не у каждого инженера-менеджера, который пытался рассуждать согласно наставлениям руководства. Решение: все эти слова-термины, провоцирующие ошибки, были табуированы. В руководстве их больше нет.

Когда разберётесь с содержанием разговора в условиях табуирования термина, просто назовите разные объекты, которые вы называли с вашим собеседником одним термином, разными словами-терминами, и продолжайте разговор — «споров о терминах» больше не будет.

Использование каких-то определённых терминов для определённых понятий (например, «car», и никогда «автомобиль») означает принадлежность к какому-то определённому сообществу, предпочитающему использовать именно эти термины, говорящем на своём диалекте. Но есть и другие сообщества, для этого же концепта использующие другие термины. Важно договориться, а термины не важны. В системном мышлении, которое трансдисциплинарно/фундаментально, этому уделяется огромное внимание: **мышление идёт в понятиях (типах и отношениях), а не в словах**. Слова могут вызвать какие-то ассоциации, они могут быть полезны для этого, но всё-таки мышление идёт в понятиях.

Никогда не видевшие автомобиль люди племени мумба-юмба вообще не знают понятия, обозначаемого словом «автомобиль». Но знающие про существование автомобилей и не знающие терминов друг друга люди не смогут договориться, если один будет требовать «car», а второй — предлагать не понимаемый собеседником «автомобиль». Но тут хотя бы будет подозрение, что проблема в разных языках. Но вот когда один требует тачку, а ему дают садовую тачку, а не ожидаемый им

автомобиль — и настаивают, то вот тогда в реальных проектах появляется проблема.

Терминология преходяща, концепты выживают дольше. Во времена СССР компьютер назывался ЭВМ (электронно-вычислительная машина), а теперь уже «компьютер». Значение не поменялось, поменялась речь — то есть поменялся термин, слово-обозначение. И речь-слова меняются много быстрей, чем означаемые ими предметы-значения: слово «тачка» уже выходит из моды для значения «автомобиль» как «недостаточно сленговое» в определённых кругах и постепенно заменяется там словом «тачило». Компьютер уже необязательно электронно-вычислительная машина, это может быть квантовый или оптический компьютер/computer/вычислитель. ЭВМ оказалось просто отсылкой к конкретной аппаратной реализации не через гидравлику, пневматику, механику, а «электронику» — задействование в режиме **электронных ключей**[68] сначала электронных радиоламп, а затем дискретных транзисторов и потом транзисторов на полупроводниках (ключ переводит какую-то «дверь» из состояния «открыто» в «закрыто», и обратно). Но попробуйте погуглить «электронный ключ», и вы найдёте, что основное значение этого термина тоже поменялось! Это и «ключ электронной подписи», и просто ключ от двери с «электронной начинкой».

Терминология может существенно отличаться не только для разных профессий, но и для разных подпрофессий внутри одной профессии. То, что называется «программным средством» для системных аналитиков, работающих по ГОСТам, будет «приложением» для продавцов иностранного софта, или «софтиной» для разработчиков.

Если невозможность договориться о терминах становится реальной проблемой, мешающей реализации проекта — к её решению есть разные подходы:

- **Терминологический фашизм**, когда только один термин объявляется кем-то правильным, а все остальные — неправильными (сравните с «Grammar nazi»[69]). У этого подхода есть много вариантов — безусловно требовать единственности используемого термина (отсутствия синонимов для термина), требовать ещё и соответствия принятым стандартам (определённым ГОСТам, например, а не учебникам или другим ГОСТам), требовать использования отечественного корня в слове («мокроступы» вместо «галоши»), настаивать на соблюдении традиций («калоши», но никак не «галоши»), игнорировать современные нормы («кофе» только мужского рода, хотя уже давно даже по словарям можно и среднего).
- **Терминологический пофигизм**, когда на слова и их значения вообще не обращают внимания. Никаких «заведомо правильных вариантов» или ссылок на авторитетные источники. При этом, если значение слова меняется по ходу разговора, это часто вообще не отслеживается, речь оказывается «не строга».
- **Строгость значений (удержание типа) с разрешением самых разных терминов-синонимов, обозначающих одно понятие.** При таком подходе обычно очень долго договариваются, какое именно понятие имеется в виду, а затем уже используются любые слова-термины для указания на

[68] https://studbooks.net/2347860/tehnika/elektronnye_klyuchi
[69] https://neolurk.org/wiki/Grammar_nazi

оговорённое понятие. При этом вполне допускается использование терминов, предпочитаемых разными профессиональными речевыми сообществами. Более того, можно и не пользоваться точными терминами, если будет понятно значение. Так, при обсуждении автомобиля вполне можно обозвать его «самобеглой тележкой», и это не будет преступлением, если адресат сообщения поймёт, о чём речь.

В нашем руководстве будет использоваться **подход, добивающийся строгости понимания значений (понятий), при возможном использовании разных обозначений-синонимов, то есть разных терминов для одного и того же понятия.** Назови хоть горшком, хоть используй пять терминов из пяти разных стандартов на трёх языках — но договорись о том, какое именно понятие/concept ты имеешь в виду: собеседники должны понять не термин, а что ты под этим термином подразумеваешь: понять значение-понятие, а не знак-термин. Это обсуждается семантикой, учением о значениях. Системное мышление основано не только на теории понятий и онтологии, оно основано и на семантике. Для беглого владения системным мышлением нужно как-то быть знакомым и со всем интеллект-стеком.

Когда указываются несколько терминов-синонимов, они в нашем руководстве пишутся через слеш, «паровозиками»: программное средство/software/application/приложение/софтина. А на то, что у каждого из этих синонимов немного разные оттенки значения, мы не будем обращать внимания — или специально обсуждать эту разницу, если она вдруг важна и речь идёт не о терминологическом отличии, а содержательном отличии, разнице понятий.

Критика такого подхода тоже не редкость: «Как вы можете учить людей, когда одно и то же обозначаете разными словами? Вы должны выбрать один термин, и затем использовать в руководстве для обозначения какого-то понятия только его! Так всегда делают в учебниках!». Ответ на эту критику прост: в жизни вы имеете все шансы встретить людей, которые обозначают понятия не теми терминами, которые введены в монографиях, учебниках, статьях, материалах видеокурсов, стандартах, регламентах, в том числе введены даже в нашем руководстве. Например, разговор может идти вообще не на русском языке, поэтому выбор термина будет ещё и зависеть от варианта перевода. Так что наше руководство будет вас тренировать на использование самых разных терминов для одних и тех же понятий: обращайте внимание — вас не просто учат новым словам, не просто заставляют зубрить терминологию. Вас стараются научить пользоваться понятиями системного мышления и их связями, под какими бы словами-терминами эти понятия ни скрывались в разных ситуациях.

Про то, что указание многих синонимов замедляет чтение текста, мы уже писали: медленное чтение и размышление над прочитанным улучшают результат обучения. Если текст читается быстро и гладко, то не факт, что через три дня из этого текста что-нибудь вспомнится.

У нас есть в пользу принятого решения о явной демонстрации синонимизации через «паровозики» терминов ещё и благодарности от инженеров-менеджеров, которые осваивали системное мышление, используя наше руководство. Инженеры-менеджеры, знакомясь с первой половиной руководства, обычно не понимают, зачем так сложно выражать мысли через синонимические ряды, но где-то при освоении работы по второй половине наставлений у них набирается опыт

использования системного мышления в реальных рабочих проектах, где царит многообразие терминологии. И тут им становится понятно, как важно перестать пользоваться единственны «точным термином» и привыкнуть мыслить содержательно: в понятиях, а не «по форме» — по лексике, терминам. Синонимические ряды как раз и вырабатывают эту привычку понимать смысл высказываний, обращаться к понятиям, независимо от употребляемых терминов.

Слова-термины важны и не важны

Парадокс «обращайте внимание на слова и не обращайте внимание на слова»/**«термины важны и не важны»** возникает тогда, когда с одной стороны, мы признаём, что у нас основной способ получения информации о мире — это слушать или читать чью-то не очень формальную и строгую речь, изучать какие-то формальные модели (но эти формальные модели тоже описываются словами). Не знаешь терминов — не поймёшь, о чем тебе говорят. С другой стороны, слова не важны, термины ничто — нас-то интересуют обозначаемые ими концепты! И обольщаться цепочками букв на бумаге или цепочками фонем в устной речи (то есть обольщаться знаками-терминами) нельзя: у разных людей они означают разное. Например, традиционная проблема в администрировании организаций (финансы, кадры, и прочие службы с превалированием учёта и госучёта, подробнее на эту тему будет в руководстве по системному менеджменту) — это обозначение абсолютно разных объектов одним термином. Например:

- «Командировка» для посылающего в «командировку» сотрудника начальника — это «сделать дело удалённо». В неё входит какая-то деловая поездка, а также выполнение дела, которое невозможно сделать без поездки.
- «Командировка» для командированного сотрудника — это «деловая поездка». Скорее всего, дело из этой поездки он считает просто работой, а вот сама поездка (транспорт, гостиницы, недомашняя еда и всё с этим связанное) — это и есть «командировка».
- «Командировка» для бухгалтерии — это «отчётная часть деловой поездки», то есть только тот транспорт, те гостиницы, то питание, которые будут оплачены и должны быть как-то подтверждены.

Если обсуждается «командировка», то один и тот же термин будет обозначать три разных объекта, путаница и споры неминуемы! Похоже, такие «три разных объекта», именуемых одним словом-термином, существуют для большинства ситуаций, по поводу которых сотрудникам и их начальникам приходится взаимодействовать с администрацией. И ровно поэтому так трудно обсуждать вопросы с финансовыми, кадровыми и прочими административными службами: все говорят о разном, употребляя одно слово.

Мы используем в жизни слова-термины в два приёма:

- **Они дают наводку на то, в какой роли какой деятельностной/культурной ролевой игры говорит сейчас человек, употребляющий эти слова.** Если кто-то говорит «баланс», «отчёт о прибылях и убытках», «кассовый разрыв», то мы догадываемся, что перед нами человек в роли финансиста, бухгалтера или экономиста. И дальше мы должны быть аккуратны в выборе слов-терминов, связанных с финансами: в этой предметной области «затраты», «издержки» и «расходы» означают разное (это вы должны знать из менеджерского кругозора)! Но вы

предупреждены, вы уже догадались, что разговор идёт с человеком-в-роли, и будете осторожны, не будете путать эти слова. Так что слово-термин тут используем в ряду других слов-терминов как маркер того, о какой дисциплине/теории/учении какого метода идёт речь, какие концепты используются в разговоре, концепты ведь редко используются по одиночке! Нужно смириться и считать, что окружающие вас люди говорят не столько на иностранных языках, но когда они находятся в каких-то деятельностных/культурных/профессиональных/проектных ролях, то используют хитрые диалекты с хитрыми значениями каких-то вроде бы знакомых слов (как в случае «командировки», надо быть внимательным). Это миф, что под словами естественного языка (даже если это слова-термины) скрываются в речи разных людей в разных ролях одни и те же концепты (места в пространстве смыслов[70]). Особенно это верно, если говорим о переводах на русский! Переспросите собеседника, чтобы понять, как он употребляет какой-то подозрительный вам в плане его значения термин в своём профессиональном диалекте — пусть приведёт какие-то примеры ситуаций, где ему нужно будет использовать это слово (не просите дать определение! Это бесполезно, а часто даже вредно). Всегда помним, что собеседники вовсе не обязательно используют термины в тех значениях, в которых вы привыкли — и когда вы им эти термины говорите, они вас поймут не так, как ожидаете, и когда они вам говорят эти термины, вы их поймёте не так, как они ожидают. **Термины важны! Выбирайте слова, когда говорите! Думайте, что значат термины, которые слышите от собеседников!**

- Когда становится понятно, какую деятельностную роль играет собеседник, то можно восстановить по слову-термину его понятие/концепт. Мы прицельно используем слова в качестве терминов для понимания или высказываний — уже зная с использованием других упоминавшихся слов-терминов, в концептах какой дисциплины искать этот концепт. Понимание термина при слушании/восприятии и чтении, а также употребление вами термина в речи или на письме при высказывании обозначаемого концепта сильно зависит от того, какую деятельностную роль сейчас отыгрывает ваш собеседник — то есть о концепте вы можете судить, рассмотрев эту роль и используемый этой ролью метод. Если речь идёт о «балансе», то вы сначала по другим словам-терминам в речи должны понять, что речь идёт не о балансе административных сил в правлении компании (бытовое использование слова, вряд ли тут была какая-то «теория/дисциплина»), не о балансе спроса и предложения (роль экономиста, термин из дисциплины экономики), а бухгалтерском балансе (термин из дисциплины бухгалтерского учёта). И дальше вы должны говорить с этим человеком (пока он находится в выявленной вами его роли), используя его диалект. Продолжайте говорить на его языке. Переучивать исполнителя роли на предпочитаемый вами для обсуждения ваших предметов интереса язык и нельзя, и вредно: вы разрушите язык, на котором исполняющий профессиональную роль человек имеет максимальную интуицию про свою предметную область — если у него есть эта интуиция. Если играет деятельностную роль случайный человек (или ненастроенный AI), то внятного профессионального предмета интереса у

[70] Подробней про пространство смыслов, концепты и термины см. в книге «Визуальное мышление», https://ridero.ru/books/vizualnoe_myshlenie/

него нет, важных для него характеристик системы и предпочтений в них он не знает, «роль не выучил», слов из роли не знает, то есть профессиональным диалектом не владеет, интуиции его нет, а если и есть, то она ошибочная и т.д. — тогда или меняйте исполнителя роли на другого исполнителя, или уж учите какому хотите диалекту, вложите ему в уста ваши любимые слова для каких-то концептов. Ну, или прекратите обсуждение, оно всё одно будет бесплодным: если это «случайный прохожий в интернете», прекращайте разговор, уходите.

Определения: гробик для умершей мысли

Помогут ли нам в понимании смысла/концепта, стоящего за словом, «словарные определения»? Приведение в руководстве глоссария, словарика? Нет, словарные определения не помогут! Об объекте ничего нельзя сказать, кроме того, как он соотносится с другими объектами. Определения чаще всего строятся «по Аристотелю», как родовидовые определения — определяется, какого **рода** объект, а затем его характерные отличительные признаки как **вида** для этого рода (отношение специализации). «Апельсин — это вид рода цитрус семейства рутовые, а также плод этого дерева». В жизни такие определения помогают мало, ибо разные люди выделяют разные характерные видовые признаки, полезные для разных вариантов использования понятия, а отношения их интересуют отнюдь не родовидовые, а часть-целое, реализации, классификации и т.д. Получаются всё время ситуации работы с определениями, похожие на описываемые в анекдоте: «Чебурашка, как бы тебе объяснить, что такое вертолёт? Апельсин знаешь? Так вот вертолёт совсем на него не похож!».

Значение слова определяется не его определением, а его употреблением в контексте языковой игры[71]. Нейронная сеть человека выучивает значения слов в контексте их употребления, на примерах использования в каких-то методах/практиках/культурах. Единственный способ уяснить себе значение понятия — это получить множество примеров его использования в каком-то практическом/деятельностном/культурном/стилевом/«применения метода работы» контексте говорения, каждый новый пример использования будет уточнять область значения этого понятия в данном контексте. Определение тут не поможет, но только запутает: чаще всего с определениями поступают просто как с поводом дать несколько контрпримеров к этим определениям, причём эти контрпримеры чаще всего будут указывать на другие контексты задействования обсуждаемого метода/практики.

Системное мышление не сводится к математике, хотя в последние годы физики и дали математическое обоснование для многих замеченных в разных предметных областях общих закономерностей поведения систем, ведь системы — это прежде всего физические системы. Дать строгое физическое, а в рамках физического и математическое определение его понятиям не получится.

В естественном языке и даже в формальных языках всё равно будет присутствовать неопределённость в понятиях, неопределённость в присвоении типов. Хотя эту неопределённость можно снимать разными методами до бесконечности, но она

[71] https://en.wikipedia.org/wiki/Language_game_(philosophy)

останется. Выпячиваться разными ролями для одной и той же системы будут разные свойства, одними и теми же словами будут называться разные системы, ибо каждая роль имеет собственный метод/культуру/стиль работы, задающий его типичные ситуации, и поэтому слова в определениях будет использовать по-своему.

Мы не будем давать поводов путаться в «определениях», а потом разбираться с их противоречивостью, недостаточностью, неточностью, неизбежной многозначностью, искажением при переводе с одного естественного языка на другой из-за отсутствия в другом языке каких-то понятий и соответствующих им терминов.

В системном мышлении «определения из глоссария» часто — лекарство, оказывающееся болезнью. В ситуации недоговорённостей по терминологии используйте не определения, а выработанные в современной философской логике приёмы:

- Если вы (или ваш собеседник) не хотите изменить что-то в физическом мире (и для этого объяснить, то есть разобраться с причинами и следствиями), а просто «найти истину» в порядке любопытства, то спор надо заканчивать: споры про «истину» и «про термины» теоретически бесплодны. Если непонятно, как вы будете использовать термины для какого-то действия, изменяющего мир (а не изменяющего описания мира, какая разница какое описание мира, если сам мир не изменяется, «бумага всё стерпит»!), то можно и не договариваться, незачем ведь договариваться: мир не изменится и от того, что хоть как-то договоритесь, и от того, что не договоритесь. **Если от результатов спора физический мир не изменится, то можно не волноваться, прекратить спорить и разойтись сразу, не тратить время впустую. Но если мир от результатов спора изменится (оба договорились о совместных действиях), продолжаем спорить и договариваться о следующих шагах, следующих действиях, искать объяснения (то есть обсуждать причины и следствия).**
- **при описании малопонятного идите не к более общим понятиям (путь «словарного определения» по Аристотелю), а «заземляйтесь» (grounding): описывайте физический мир. Уход в более общие понятия и описание непонятного как специализаций (аристотелевские определения «X это Y с такими-то особенностями») — тупик.** Чтобы договориться, нужно конкретизировать, а не абстрагировать — не сводить к известному спорщикам общему, а сводить к известному спорщикам физичному. Для частного подразумевается противостоящее ему плохо определимое общее или обсуждается непонятно как реализованная в физическом мире «функциональная роль». Абстрактные понятия — это чаще всего ролевые/функциональные объекты, они отсылают к какому-то поведению. Поведение обсуждать всегда сложней, чем статичные физические объекты, ведущие себя как-то вещи. Поэтому для каждого ролевого объекта подставляем в виде примера какой-то понятный конструктивный/материальный объект (вещь), играющий роль функционального объекта — и обсуждаем его поведение, оно сразу будет понятно.
- **на время разговоров о терминологии сам термин табуируйте[72] (всё**

[72] https://lesswrong.ru/wiki/Табуирование

равно его использование не ведёт к пониманию, и только замедляет дискуссию), вместо него давайте развёрнутые формулировки. Если вам нужно продолжать разговор, а слово собеседнику явно непонятно, то продолжайте дискуссию без этого термина! В дискуссиях не про термины, а про дело, термины нужны просто для того, чтобы обсуждение было компактнее, но сам спор о терминах убирает все преимущества использования термина, без него оказывается проще. Если спор превращается в спор о терминах (про «истину», и «как правильно называть», а не про «что делаем»), то это тревожный симптом бесплодного спора, немедленно прекращайте спорить и выходите из разговора.

При объяснении незнакомых понятий через другие незнакомые понятия помогает понимание **«понятийных расстояний»**[73]. Эти расстояния обычно короткими не бывают: попробуйте объяснить третьекласснику, как строить вольт-амперную характеристику для тиристора. Расскажите ему про вольт, ампер, характеристику и что такое тиристор. Это чистое враньё, популизм про «настоящий профи всё может понятно объяснить даже ребёнку». Вы же не верите в сказки? Чтобы объяснить что-то сложное ребёнку, надо этого ребёнка учить много лет, чтобы он стал образованным. Ребёнок станет за это время взрослым.

С третьеклассником и объяснениями для него сложных понятий все всё понимают, но два профи часто свои «тиристоры» и прочие сепульки пытаются всё-таки «определить» в трёх фразах — разница в возрасте ведь небольшая, и кажется, что объяснить можно, выдавая «определения, понятные ребёнку». Но нет. Сколько определений не давай, пользы от них не будет. Бойтесь, бойтесь определений — это то лекарство, которое оказывается болезнью чаще, чем лечит. **Георгий Петрович Щедровицкий любил повторять: «Определение — это гробик для умершей мысли».** И он был далеко не одинок в этом, разве что говорил о недостатках определений более часто, красочно и броско, чем многие другие специалисты по методам мышления в составе интеллект-стека.

В нашем руководстве нет никакой опоры на определения, нет глоссария (толку с него?!). Последний раздел руководства даёт сжатое изложение онтологии системного подхода третьего поколения, но он специально носит название «Вместо глоссария: онтология системного подхода третьего поколения» и представляет собой связный рассказ с отсылками к литературным первоисточникам вместо набора определений.

Жирным шрифтом в нашем тексте выделены термины в тех местах текста, где впервые много рассказывается о концепте, обозначаемом этим термином. Конечно, в тексте руководства есть и какие-то фразы, похожие на определения, но нельзя считать их именно определениями. Потом в тексте руководства для этих же терминов и обозначаемых ими понятий обязательно будут другие фразы, тоже похожие на определения. А потом ещё и ещё раз: текст наставления тренирует вашу нейронную сетку в мозгу, формирует в ней значения для незнакомых слов, ведёт языковую игру.

В пространстве смыслов для вас с каждым словоупотреблением всё чётче и чётче определяется место понятия, скрывающегося за термином и его синонимами. Ещё ведь в терминологии бывают и синонимы как термины из других контекстов, с другими оттенками значений и другими ассоциациями. Но употребления

[73] https://lesswrong.ru/w/Ожидая_короткие_понятийные_расстояния

терминов — это всё не определения, нет. Результат такой понятийной работы по тренингу нейронной сетки живого мозга развёрнутыми текстами с использованием слов в деятельном/культурном/практическом/трудовом контексте выполнения работы каким-то методом (всё по Витгенштейну: смысл слов определяется их употреблением! не определениями!) достигается в ходе обучения, и он обычно хорош. А от определений в обучении, в спорах и везде в общении (если только речь не идёт о математике!) результат — только споры и просьбы уточнить/изменить определение.

Системное мышление — про жизнь, это не математика, где у точек нет размера, а параллельные линии не пересекаются даже в бесконечности и длины они тоже бесконечной. В жизни всё по-другому, в существенной мере определяется нейронными сетями и нейросетевыми представлениями, поэтому попытки сразу перейти на строгие формализации обречены на провал. Системное мышление не математично (в той мере, в какой это не математика, описывающая пространство смыслов, например, математика нейросетевых представлений в больших языковых моделях), оно про не слишком формально устроенную и не слишком формально описываемую жизнь!

Значения терминов (да и любых других слов, даже если их не называть торжественно «терминами») определяются вероятностно, а не точно — и это делается путём их употребления в разных ситуациях «из жизни», то есть язык определяется не тем, что надо что-то описывать, а тем, что надо что-то описывать для того, чтобы что-то сделать. В любой работе важны её объекты, с которыми выполняются действия метода/практики/деятельности/культуры/стиля/инженерии/труда — вот это задаёт контекст для языковой игры, в которой определяются значения слов.

Мы считаем, что ваши догадки о значении терминов из руководства (а потом ваши догадки о значениях терминов, которые вы будете встречать в рабочих проектах) будут строиться путём изучения развёрнутых текстов, описывающих разные практические/деятельностные/культурные ситуации (ситуации работы по какому-то методу), изучения разных отношений обозначаемых этими словами концептов с другими концептами, обозначаемыми другими рядом использованными словами. При определении значений слов-терминов читаем не определения, а разбираемся прежде всего с текстами и речами как наборами развёрнутых высказываний, содержащих интересующий нас термин (если даже речь идёт о фотографии или диаграмме, то всё равно дело закончится тем, что вы будете описывать их словами, и опираться на термины в этих описаниях — вы же не будете обсуждать их мычанием!).

Так что сначала всегда **разбираемся со словами-терминами (обращаем на них внимание, и не по одному слову, а по использованию в контексте! Слова важны!)**, которые нам встретились, т.е. определяем их смысл, вытаскиваем из этих слов безымянный концепт, который имеет самые разные имена в разных профессиональных языках самых разных сообществ, придерживающихся самых разных методов работы/действий, поклоняющихся самым разным словарям (в том числе неписаным, «жаргон») и другим терминологическим стандартам — и **дальше работаем с концептом (не обращая внимания на слова-термины! С этого момента начальные термины можно и заменить! Слова не важны!)**.

Жизнь сложна, и мы её упрощаем, рассматривая каждый раз только пару-тройку

взаимосвязанных понятий/концептов (ещё и опуская длинные дискуссии о нюансах концептов, принятых в теориях/объяснениях тех или иных методов). Для этого мы уходим от дискуссий о длинных списках имён этих концептов в разных диалектах разных деятельностных сообществ. **О концептах узнаём по их терминам. Термины важны! Без них беда, ничего не узнаем! Мычать нельзя, это неинформативно! Но мы никогда не придираемся к самим терминам, они для нас неважны, а важны обозначаемые ими концепты!**

Задания: обходимся без определений и не спорим о терминах.

Поставьте отметку о выполнении дел:

1. Найдите литературу, которая предлагает отказаться от определений. Какие в этой литературе высказываются основные аргументы против выдачи определений для введения новых понятий в разговор? Напишите пост с эссе по поводу традиции избегания определений в текстах (речь не идёт о математических текстах).
2. Напишите пост, где дайте три примера ваших попыток преодолеть далёкое понятийное расстояние (не забудьте перед написанием поста освежить в памяти https://lesswrong.ru/w/Ожидая_короткие_понятийные_расстояния, эта работа была упомянута в предыдущем подразделе).
3. Вспомните десяток ваших последних дискуссий. Какие из них были «спорами о терминах» и поэтому заведомо бесплодны? Напишите об этом пост.

Как выбирались термины для наших руководств

Наука традиционно порождала новые термины (обозначения для появляющихся новых понятий) двумя способами:

- Бралось обычное («бытовое») слово, и нагружалось специальным («научным») значением. «Работа» в физике — отнюдь не «работа» в бытовом значении этого слова. Более того, «работа» в тексте наших руководств — это тоже не бытовое слово. Так, нельзя путать «работу» и «метод/практику/деятельность/труд», это не синонимы. Более того, кавычек при этом не будет — вы можете не заметить, что это термин! Работу и практику путать в мышлении нельзя (это будет выглядеть примерно так, без кавычек и указания уточняющих типов, просто работа и практика, легко перепутать). И работа, и практика ещё и оба означают поведение, общий для них тип. Но это разные подклассы одного класса «поведение». «Работа» и «практика» тут термины со строгой/формальной типизацией, и они разных типов! «Бытовое слово» как «термин со специальным значением в каком-то методе» самый частый способ, но он легко приводит к путанице со словами из бытовой речи, ибо это же «ложные друзья переводчика» при переводе с научного языка на бытовой и наоборот.
- Чтобы сделать речь точнее, термином делалось слово, для которого в бытовой речи не было известных значений. Для этого необычное для родного уха слово бралось из иностранного языка (чаще всего — с греческим или латинским корнем) и нагружалось специальным значением. Сегодня в русском языке прихватываемым словом может быть английское слово, а не

латинское или греческое — в русском-то оно бытового значения не имеет. Но тут тоже не без проблем. Если вы говорите греческое слово «эпистемология», то его нужно разъяснять, и придётся разъяснять долго, задавая «словоупотребление в контексте языковой игры в каком-то деле». А «учение о том, как делать исследования» — вроде сразу понятно, о чём идёт речь, но не факт, что под этими словами будет понята именно «эпистемология», с равным успехом это может оказаться «гносеология», и в ней вы вдруг встретитесь не только с научными исследованиями в целях познания мира, но и с художественными и богословскими «исследованиями». В эпистемологии художественного и религиозного познания мира обычно нет, а в гносеологии они обычно есть. Поэтому-то и используются очень редкие слова, чтобы не путаться в близких, но всё-таки разных значениях слов.

У нас в руководствах термины выбраны (в том числе при переводе иноязычных текстов — стандартов, методик, учебников) для максимизации понятности их употребления в деятельности. При выборе терминов учитывалось: кто поймёт это слово, из какого он профессионального сообщества, на каком диалекте предпочитает говорить? Это другой принцип, нежели «взять термины из близкого авторам стандарта и игнорировать все другие варианты».

Например, мы легко можем использовать в руководствах жаргонные слова. Скажем, «айтишник», а не «программист» — ибо нас заботит не только красота речи и привычные термины, но как можно более точное указание на значения терминов в реальном мире. Ведь «программист» более узкий термин, чем «айтишник». Администратор базы данных, модельер данных и инженер данных, системный администратор, IT-архитектор, электронщик — все они не программисты, но айтишники. Можно было бы слово «айтишник» заменить словом «компьютерщик» — кому-то это было бы ещё понятней, но кто-то стал бы возмущаться. С учётом всего этого мы могли бы написать программист/айтишник/компьютерщик — чтобы никому не было обидно и было бы понятней, какое значение всех этих терминов мы имеем в виду.

Бывает и так, что определённый термин, значение которого очень легко понять неправильно, уже закрепился в языке узкой профессиональной группы. Например, таков перевод «управление» для термина governance. В таких случаях в наших руководствах будет использоваться наш собственный вариант, который ведёт к меньшему числу ошибок понимания. Например, governance (в руководстве по системному менеджменту) будет переводиться как «подконтрольность» или «поднадзорность» (или даже «контроль» и «надзор», или даже «присмотр»), и никакие словари и стандарты тут не указ. Похожая ситуация с frustration. Это не «психологическая фрустрация», а геометрическая неустроенность/неустаканенность. Поэтому никаких «фрустраций» в переводе тут не будет, они ведь только будут сбивать с толку.

Если какой-то «процессный стандарт» (например, системноинженерный ISO 15288:2023) под словом process имеет в виду концепт, в различных других местах обозначаемый как метод/method/практика/practice (характерной для процессов развёртки во времени в этом «процессе» из ISO 15288 нет, там перечисляются именно «методы/практики/процессы жизненного цикла», это стандарт не успел отразить метод «непрерывного всего»), то в нашем руководстве это будет «метод работы», а не «процесс». Если вы попали в сообщество людей, исповедующих «процессный подход» и говорящих на диалекте, используемом в этом сообществе,

смело используйте слово «процесс» или «рабочий процесс», или «организационный процесс» (но лучше бы не «бизнес-процесс», выход уборщицы на мытьё полов — это явно не «бизнес-процесс», а в государственных организациях «бизнес» тоже не поощряется) вместо слова «метод» — но знайте, что при этом вы теряете информацию по различению процессов (с пошаговой развёрткой во времени) и методов работы (без такой развёртки), и речь ваша будет время от времени вызывать недоумение: тогда не ленитесь уточнять значение использованного вами термина. Если мы различаем работу и метод/практику (а мы будем различать!), то труд — это работа или метод? Чаще всего мы выбираем, что труд — это функция/практика/культура/способ/метод работы (именно на это указывает термин «разделение труда», division of labour с акцентом на «разделение рабочей силы по видам труда»), а не работа. И поэтому в наших руководствах мы будем стараться не считать труд и работу синонимами. Но вы легко можете найти такое сообщество, где их различать не будут, и вам нужно будет в ваших разговорах в этом сообществе это учитывать.

Обычно за одним и тем же термином даже в одном речевом сообществе оказывается закреплено множество разных значений, поэтому уточнить значение даже очень распространённого термина никогда не бывает лишним. Например, Andries van Renssen выделил[74] следующие часто используемые сообществом айтишников значения для термина «**функция»** (function):

- подтип активности (поведения), процесса или события;
- некий объект, находящийся в определённой роли или сделанный для определённой роли;
- сама роль объекта (обычно это роль конструктивного/материального объекта, вещи), участвующая в активности (поведении). Играемая роль (функциональный объект) и конструктивный/материальный объект, играющий роль — это разное! Роль — Гамлет, играющий эту роль объект — Высоцкий;
- указание на корреляцию/зависимость, обычно как на физическую связь между какими-то аспектами: «если высота растёт, то давление падает»;
- математическое отношение между числовыми объектами, определяющее их отображение друг на друга/mapping.

Это означает, что часто вместо слова «функция» можно использовать слово «действие», или «роль», или даже «зависимость» — и разговор от этого станет точнее.

Часто встречающийся термин «**мета»** (meta) используется в шести разных значениях, выражая шесть разных типов отношений[75]:

- экземпляризация (отношение типа и экземпляра);
- группирование (отношение типа и подтипа), оно же категоризация (философская, а не из теории категорий, термин «категория» любим самыми разными речевыми сообществами, и обозначает в них разное!);

---

[74] стр. 79 в https://repository.tudelft.nl/islandora/object/uuid:de26132b-6f03-41b9-b882-c74b7e34a07d/datastream/OBJ/download (Gellish: A generic extensible ontological language - design and application of a universal data structure)

[75] https://ailev.livejournal.com/1053878.html

- описание (отношение описания и описываемого объекта);
- применение/стереотип (отношение шаблона и его воплощения);
- варьирование (отношение основной модели и кастомизированной);
- реализация (отношение абстрактного синтаксиса и соответствующего ему выражения. Это в информатике, ибо цитируемая работа относится к информатике. Но помним, что у нас будет и другое значение термина «реализация» — это отношение выполнения роли функционального объекта каким-то конструктивным объектом).

Поэтому каждый раз, встречая слово «мета» нужно разбираться, что именно из этих шести значений имелось в виду.

Никогда не зацикливайтесь на выбранных другими конкретных словах-терминах! Слова-термины крайне важны, ибо кроме них у нас ничего нет для общения. Но слова как цепочки букв в письменном тексте или последовательности фонем в речи никогда не однозначны, они лишь «ярлыки» на понятиях. Каждый раз пытайтесь понять, о чем в действительности идёт речь, какое значение слова имелось в виду в каждом конкретном случае, добирайтесь до концепта/понятия/значения термина. Слова-термины не так уж важны, они нужны только для того, чтобы добраться до значения или понять, о какой предметной области мы вообще разговариваем!

Использование терминов из стандартов не гарантирует однозначного понимания собеседником, но и использование многозначных слов не обязательно ведёт к сложностям. В наших руководствах не будет как попыток дать точные определения, так и попыток выбрать «правильные термины» для понятий. Мы постараемся передать понимание наиболее важных понятий в развёрнутых описаниях и предложить разные слова, которыми их можно обозначать. На вопрос «сколько будет дважды два?» будут приниматься ответы и IV, и 4, и «четыре», и four. Но не нужно обольщаться: ответы «горшок», 5, «per aspera ad astra» — приниматься не будут.

Вопросы. Терминология

В одной книжке по системной инженерии конструктивный объект называют компонентой, а функциональный — модулем, а в другой книжке конструктивный объект называют модулем, а функциональный объект — компонентой. Авторы пишут примерно об одном и том же, только используют для этого прямо противоположные термины. Какой книжке верить?
--- Не верить никакой, ибо сами не знают, о чём пишут, если так вольно поступают с терминами. Перед написанием книжки они должны были справиться с каким-то общепризнанным словарём или стандартом. ###
Нет, верить можно обеим книжкам. Просто авторы пользуются разной терминологией, но разные слова обозначают одни и те же концепты. А ещё можно ожидать, что найдутся разные словари и стандарты, все они будут «общепринятыми» в разных сообществах, и в них можно будет найти совершенно разные определения терминов «модуль» и

«компонента». В некоторых даже отдельно укажут, что «компонент» и «компонента» — это синонимы!
+++ Верить обеим книжкам, только каждый раз пытаться понять, какое значение скрывается за этими терминами. ### Да, разные книги (и даже разные стандарты) вполне могут использовать одни и те же слова-термины в прямо противоположных значениях, даже если их авторы хорошо знакомы с предметной областью.
--- Найти стандарт, определяющий слова, и верить той книге, в которой используются определения по стандарту. Другой книге не верить. ### Нет, вполне может существовать и другой стандарт, и какой-нибудь словарь, в котором термины определяются и как в другой книге. Или книга может оказаться такой, что после неё именно эти значения слов попадут в очередные стандарты и словари. Нет повода не верить другой книге только из-за выбора терминологии.

Определение программного средства из п. 2 разд. 1 ГОСТ 28806-90: «Объект, состоящий из программ, процедур, правил, а также, если предусмотрено, сопутствующих им документации и данных, относящихся к функционированию системы обработки информации. Примечание - Программное средство представляет собой конкретную информацию, объективно существующую как совокупность всех значимых с точки зрения её представлений свойств каждого из материальных объектов, содержащих в фиксированном виде эту информацию». Укажите верные утверждения по поводу использования термина «программные средства» в нашем руководстве по системному мышлению:
--- Авторы руководства обязаны использовать ГОСТ 28806-90 и поэтому в руководстве используется термин «программное средство» (по ГОСТ 28806-90) ### Нет, авторы руководства не обязаны использовать какой-то ГОСТ или иной международный стандарт или публичный документ, или учебник, или монографию, или статью, где вводится какая-то терминология. Поэтому будут использовать также и другие термины, если не ожидается проблем с их восприятием — «софт» (и даже «софтина»), software, программа, приложение, хотя также может быть использован и термин «программное средство».
--- Авторы руководства не поощряют использование канцеляритов из различных ГОСТов, поэтому термина «программное средство» в руководстве не встретишь. ### Нет, у авторов никаких предубеждений к использованию терминов из различных стандартов и публичных документов нет. Термин «программное средство» вполне можно встретить в руководстве по системному мышлению, равно как встретить и самые разные другие формальные и неформальные термины из стандартов, публичных документов, самой разной другой литературы.
--- Авторы руководства знают о существовании термина «Программное обеспечение» из самых разных переводных ГОСТов, например, ГОСТ Р 56713-2015 (ISO/IEC/IEEE 15289:2011) и используют его, потому что это перевод английского термина software. ### Нет, в разных ГОСТ переводчики очень по-разному переводили термин software, а у авторов

руководства нет предубеждений по преимуществу каких-то одних стандартов над другими.
+++ Авторы руководства используют самые разные термины для программного обеспечения, программных средств, софта, программных приложений, компьютерных программ — и даже могут в одном месте использовать несколько синонимов, чтобы быть понятными разным словарным сообществам. ### Да, термины сами по себе не важны, важно их значение. Авторы руководства используют самые разные термины для обозначения того же, что в ГОСТ 28806-90 называется «программное средство».

3. Творчество: догадки и их критика

Формальность системного мышления

Знание в отличие от «просто фактов» — это то, что можно использовать в разных ситуациях, что можно взять с собой из проекта в проект. Чаще всего это какие-то объяснительные теории. **Факты** же могут характеризовать конкретные проекты и объекты в них. Знание о метрах как единицах измерения общее для всех проектов. Длина пути в каком-то проекте 14 метров — это нельзя применить к другим проектам, так что это не «знание», это просто «факт».

Логика (правила рассуждений, ведущие к правдоподобным суждениям из правдоподобных посылок) науки и инженерии совсем необязательно булева/дискретна со значениями «истина» и «ложь». Современная логика вероятностна, при этом при использовании логического вывода в рациональном мышлении речь совсем недавно шла о байесовской вероятности, а не частотной. А теперь оказывается, что и байесовская вероятность тут не последнее слово, а речь идёт о «квантовой вероятности» из математики квантовой физики, хотя эта вероятность не имеет отношения к поведению частиц в микромире, а характеризует совсем другие масштабы в мышлении (иногда это называют квантовоподобностью)[76].

Современная логика использует вероятностные (байесовские и квантовоподобные) рассуждения, а эксперименты в конечном итоге не доказывают что-то или опровергают, а лишь сдвигают вероятность правдоподобности[77]. Жизнь не состоит из «истин» и «лжи», она оценивается высказываниями о вероятностях, и даже более: жизнь состоит не столько из высказываний, сколько из действий, выбранных рациональным способом — высказывания тут носят вспомогательный характер! Мы уже обсуждали, что «здание состоит из кирпичей» — это абсолютно верное в логике высказывание, но в жизни это может привести к тому, что вам придётся обсуждать конструкцию здания, указывая место каждого кирпича. Много проще будет объединить кирпичи в стены, а затем обсуждать место стен в здании — и системное мышление работает как раз с этим, «удобным группированием». Не всё, что логически верно будет уместным в системном мышлении, требуется связь с жизнью — с какими-то методами/практиками/культурами/стилями работы, для которых ведутся рассуждения.

---

[76] Небольшой обзор по современной рациональности: https://ailev.livejournal.com/1619025.html
[77] См. литературу в пунктах 1 и 2 http://ailev.livejournal.com/1311261.html.

Но сейчас мы рассмотрим другую ветку обсуждения связи системного мышления и логики. Формально/строго (выражено в символической форме, доступной для строгого логического вывода через значения «истина» и «ложь») ли системное мышление? Или оно неформально, т.е. интуитивно и тем самым в нём большая вероятность ошибки? Другая формулировка того же вопроса: можно ли считать, что системное мышление требует строгой/формальной типизации его объектов, как в математической логике?

Главное, что нужно тут обсудить — это наличие и важность для трансдисциплинарного мышления (а системное мышление — трансдисциплинарно) неформального, интуитивного и невыразимого словами и иными знаками знания в распределённых (например, нейросетевых — в живом мозгу или в компьютерных системах AI) представлениях, в том числе распределённого знания о системах. Это знание быстрого мышления S1 по Канеману. Сегодня такое знание могут иметь не только люди с «мокрой нейросетью» (их и изучал Канеман), но и компьютеры, запрограммированные для работы в рамках коннекционистской/нейросетевой парадигмы. Современные достижения искусственного интеллекта связаны с развитием именно «компьютерной чуйки» (а не развития логических языков программирования) в рамках машинного обучения в целом и направления глубокого обучения (deep learning) в частности.

В **коннекционистской (connectionism) парадигме**[78] знание представляется существующим не как набор связанных какими-то отношениями дискретных понятий из формальной логики, а как распределённое по множеству определённых простых однородных элементов (сегодня это обычно «нейроны» в нейронных сетях как искусственных, так и естественных). Это довольно мощная концепция **распределённых представлений** (distributed representations) в их отличии от локальных/символьных/symbolic представлений. Символическими представлениями занимается семиотика в составе семантики, распределёнными представлениями пока занимаются в рамках проектов по изучению AI, больше всего ими занимаются в предметной области **глубокого обучения** (deep learning), занимающаяся иерархическими распределёнными представлениями (например, в виде многоуровневой/глубокой нейросети).

Идею о том, что знания/информация может быть представлена в рамках распределённых представлений через теории информационной физики легко обобщить до признания безмасштабного знания, разлитого во всей вселенной (и включающего также знания людей), см. подробней в работах по представлению вселенной в виде своеобразной нейронной сети, в которой идёт вечный процесс познания[79].

Человеческий мозг для мышления использует нейронную сеть и поэтому в существенной мере работает с распределёнными, а не локальными представлениями, хотя на нейронной сети он и может организовать «виртуальную логическую машину, оперирующую с типами и отношениями между ними», то есть мозг медленно и с ошибками может работать и как классический логический вычислитель.

---

[78] См. подробнее в http://ailev.livejournal.com/1228029.html.

[79] Наиболее явно это можно посмотреть в работах Виталия Ванчурина последней пары лет, https://scholar.google.com/citations?hl=en&user=nEEFLp0AAAAJ&view_op=list_works&sortby=pubdate, есть специальный вебсайт для проекта «мир как нейросеть», https://artificialneuralcomputing.com/wann

Человеческий мозг плохо работает с формальными/математическими/строгими/символьными представлениями знаний и фактов, его вычисления вероятностны и требуют проверок. Мозг животных практически не может работать с символическими/знаковыми/локальными представлениями, мастерство работы со знаками как раз и отличает людей и животных.

Современные системы машинного обучения и искусственного интеллекта используют для своей работы принципы, похожие на принципы работы человеческого мозга. Знания сегодня понимаются отнюдь не только как формальные/строгие математические модели. Объединение методов формальной, «научной» работы со знаниями и методов «неформальной» интуитивной работы в нейронных сетях (искусственных или естественных, в мозгу человека — это тут неважно) представляет собой научный и технический фронтир, мы не будем касаться этих вопросов в нашем руководстве. В любом случае, мы можем резко уменьшить число ошибок неформальных вероятностных рассуждений, если просто дадим больше времени на вычисления. Это сейчас горячий вопрос в исследованиях по большим языковым моделям — можно ли обратиться за ответом на чуть изменённый вопрос несколько раз, чтобы получить более надёжный и безошибочный ответ? Можно![80].

Даниэль Канеман утверждает[81], что у человека есть два механизма мышления: быстрое малозатратное интуитивное S1 и медленное трудоёмкое осознанное S2, включающееся при появлении каких-то проблем при использовании «быстрого» интуитивного мышления. Вот S2 представляет собой возможность последовательного задействия актов мышления S1 и тем самым возможность логического последовательного/пошагового мышления.

По факту речь идёт о целой шкале мышления от интуитивного неформального через вероятностное (с какими-то оценками этих байесовских или квантовоподобных вероятностей по самым разным источникам априорных свидетельств и данных эксперимента) к классическому формальному на основе математической логики. Вот схема Прапион Медведевой, иллюстрирующая эту шкалу (подробней это всё объяснялось в руководстве по рациональной работе):

---

[80] https://arxiv.org/abs/2311.16452 с описанием вариантов автоматического подбора запросов, интуиции Jason Wei https://twitter.com/_jasonwei/status/1729585618311950445

[81] Даниэль Канеман, «Думай медленно… Решай быстро», https://ru.wikipedia.org/wiki/Думай_медленно..._решай_быстро

Обычно интуитивные догадки на уровне «ощущений» вытаскиваются в качестве явно сформулированных эвристик, а эвристики проверяются критикой: логическими (включая вероятностную байесовскую логику или квантовоподобную логику). В случае подтверждения догадок формальное медленное мышление S2 о каком-то типе задач потом можно натренировать (в ходе решения множества задач) так, что оно становится автоматическим и «интуитивным» S1, не требующим особых мысленных усилий, решение этого класса задач перемещается из части шкалы с «обдумыванием» и «направленным вниманием» в зону быстрого интуитивного без особого задействования дорогого ресурса сосредоточенности. У физиков появляется «физическая интуиция», у математиков «математическая интуиция», у системщиков — «системная интуиция». Мозг пластичен, нейронная сеть мозга хорошо тренируется (а в последнее время эти «интуитивные» рассуждения начали потихоньку передавать и нейронным сетям в компьютерах).

По этой шкале формальности, конечно, можно взять спектр мышления какого-то конкретного агента: и тогда мы увидим, предпочитает ли агент строго формальное мышление, или вообще отрицает в мышлении всякую формальность, или всё-таки работает главным образом на среднем уровне формальности, но может быть и вариант, когда спектр мышления более-менее равномерен по всей шкале формальности. Не путаем при этом шкалу со спектром (шкала одномерна, спектр — двумерен, это распределение некоторой характеристики по шкале, например, спектр какого-то источника света — это распределение интенсивности излучения этого источника по шкале длин электромагнитного излучения). Спектр формальности мышления может быть у какого-то конкретного мыслителя, а вот шкала формальности — она одна на всех.

Системное творчество

Медленное, «формальное», рассудочное мышление при всех его достоинствах может испытывать содержательные проблемы, даже когда люди готовы тратить на него достаточно времени. Хорошо сформулированная проблема обычно содержит в себе явное формальное противоречие, которое необходимо «снять» — только в этот момент включается **творческое мышление**, только в этот момент нужно «сесть и подумать» (а не «вспомнить и применить», рутинное, автоматическое мышление). Иногда говорят, что мышление появляется тогда, когда нужно «перевести проблемы в задачи», т.е. создать список работ, которые понятно, как выполнять, и

которые вместе решают проблему, снимают противоречие, убирают коллизии.

Решение проблем путём формулирования и снятия противоречий присуще и теории ограничений Элияху Голдратта («грозовая туча»[82]), и методологии ТРИЗ Генриха Альтшуллера[83], и системомыследеятельной методологии школы Георгия Щедровицкого[84]. Все эти школы мысли утверждают, что они основаны на системном подходе, отсюда и общность мыслительных приёмов.

Откуда вообще возникают противоречия? Они могут быть двух типов:

- **Противоречия в смысле логических противоречий, формальной несовместимости утверждений в рамках какой-то логики.** Это может быть булева логика с двумя конфликтующими высказываниями (x=3 и x=5, причём оба высказывания верны одновременно!), но может быть и вероятностная байесовская логика, в которой согласно работам E.T.Jaynes булева логика является частным случаем[85]. Более того, неразрешимые байесовско-логические (Bayesian inference) противоречия могут быть и в байесовской логике, тогда приходится вычислять по квантовоподобной (quantum-like) логике, это рассказывается в работах Андрея Хренникова[86].
- **Противоречия/конфликты во взаимодействиях систем разных уровней, ведущие к «неустроенностям» и вследствие этого сложности системного устройства на этих уровнях.** Хороший обзор того, как это устроено в биологии, можно найти в работе Вольфа, Кацнельсона, Кунина «Physical foundations of biological complexity»[87], 2018 год — и читать её хорошо бы, читая литературу к этой статье, иначе многое просто непонятно. В 2022 году этими же авторами вместе с добавившемся к ним физиком Ванчуриным была высказана идея о том, что эти противоречия/конфликты между системными уровнями, которые ведут к неустроенностям в этих уровнях и порождают эволюцию, в которой мир/world/вселенная оказывается физически нейросетью и «познаёт/learn сам себя». Вернее тут говорить, что одни куски мира (системы) познают/learn какие-то другие куски мира (системы), а термодинамический баланс не нарушается из-за того, что знания в этих хорошо познавших мир кусках мира могут «разучиваться/забываться», но эволюция всё одно берёт верх и уже другие куски мира приобретают много знаний[88]. Эволюция происходит главным образом с неэргодическими системами, то есть системами с памятью. На самых медленно меняющихся в эволюции уровнях память становится цифровой как точной в многократном копировании (без накопления ошибок) памятью, например ДНК как раз такая «цифровая память». В случае меметической эволюции у людей для обеспечения памяти и точного копирования служат речь и письменность, что позволяет накапливать и бесконечно копировать знания без их искажения. И то же самое происходит в технологической эволюции: идеи, лежащие в основе

[82] https://en.wikipedia.org/wiki/Evaporating_Cloud
[83] https://ru.wikipedia.org/wiki/Теория_решения_изобретательских_задач
[84] http://www.fondgp.ru/
[85] http://www.med.mcgill.ca/epidemiology/hanley/bios601/GaussianModel/JaynesProbabilityTheory.pdf
[86] https://scholar.google.com/citations?hl=en&user=wdhkzPMAAAAJ&view_op=list_works&sortby=pubdate, см. краткий рассказ про квантовоподобную рациональность в https://www.preprints.org/manuscript/202103.0395/v1
[87] https://www.pnas.org/doi/10.1073/pnas.1807890115
[88] https://artificialneuralcomputing.com/wann

устройств, проявляются в их конструкции. Центральная догма молекулярной биологии, что информация идёт от генотипа к фенотипу, от информационных моделей в КБ (конструкторском бюро) к экземплярам изделий в одну сторону, а в другую сторону абсолютно другие способы передачи (не «биологический рост» или «заводское изготовление», а мутации, инженерное проектирование и другие методы изменения генотипа — но точно не путём копирования адаптаций фенотипа) была подтверждена идеями из физики-математики в работе 2022 года «Toward a theory of evolution as multilevel learning»[89].

И творчество тем самым может быть тоже разное: если речь идёт о логическом/математическом противоречии, то можно переописать мир в рамках какой-то одной математики/логики, а не разных (сделать его через выделение других объектов внимания, других понятий — перейти к другой «логике»), и тем самым снять противоречие. Скажем, если вы пришли к выводу, что x=3 и x=4, то вы можете рассмотреть физический смысл написанного и обнаружить, что речь идёт о том, что в первом и втором случае использовались разные единицы измерений. Это простой случай: описываем мир в одних и тех же единицах измерений, чтобы избежать противоречий! Ещё один простой случай: если вы узнали, что один и тот же человек — физик и менеджер, то это не обязательно два разных объекта, а просто два разных описания. Описываем мастерство как профессиональные умения, то есть как умения ролей (мастерство тут будет часть функциональной декомпозиции личности, а личность — функциональная часть человека, его «софт», а ещё у человека будет организм, его «аппаратура»). Людей же описываем как агентов, исполняющих роли — и противоречие оказывается снятым.

Если речь идёт не о логическом/математическом противоречии, а о конфликте между системами разных уровней, то придётся менять поведение этих систем. Тут многое зависит от того, насколько вменяемы как агенты эти системы. Если это механическая деталь, придётся менять её конструкцию механически. Если кошка — интеллект сильнее, вменяемость повыше, можно надрессировать. Если человек и интеллект сильный, то вменяемость высока, можно «уговорить», убедить рациональными аргументами, поменять нейросемантическим программированием (текстом!) только «софт», а не «аппаратуру». Это подробней описывается в руководстве по интеллект-стеку. Изменениями описания тут не поможешь. Если конфликт в том, что вирус своим поведением в сторону выживания конфликтует с организмом, которому тоже нужно как-то выживать, то выходом будет появление иммунной системы в организме и наличие изменчивости в вирусе, и огромное разнообразие реализаций иммунной системы и способов реализации изменчивости вируса, чтобы обманывать иммунную систему. Это пока эволюционное решение, его делает эволюция, хотя это занимает много лет (иногда и миллиарды), но появляется иммунная система как «нементальный мозг»[90]. Это всё жизнь, дарвиновская эволюция, где геном в каждой клетке.

Альтернативное решение — это инженерное, в порядке техно-эволюции. Если вы инженер и нашли противоречие в окружающем мире, вы можете создать новый механизм или изменить имеющийся, предложить смарт-мутацию к мемому. Мемом — это как «геном общего вида», мем/meme — это как ген/gene, только необязательно в ДНК. О таких решениях думают не только люди, они могут

[89] https://www.pnas.org/doi/10.1073/pnas.2120037119
[90] https://www.frontiersin.org/articles/10.3389/fnint.2023.1057622/full

возникать спонтанно, эволюционно в ходе меметической и техно-эволюции. Например, мемовирусы (идеи, которые люди считают интересными для распространения) могут быть такими же, как и обычные вирусы: подчинять себе поведение людей (в природе такие вирусы, меняющие поведение хозяина, зовут «зомби», а ещё есть бешенство — вирус, меняющий поведение заражённого им в агрессивную сторону, увеличивающую распространение вируса). John Doyle говорит, что человечеству нужно инженерно задуматься над вопросом об иммунной мемосистеме, а не ждать, пока мемовирусы погубят человечество[91].

Примером инженерного решения, которое уменьшает число смертей, является демократия. Когда демократии (выборов с целью бескровной смены диктатора) не было, власть или не менялась вообще (самодержавие того или иного рода), или вопрос власти решался гражданской войной — ибо «бог всегда на стороне больших батальонов» при равном вооружении и уровне знаний. Смена власти происходит, но довольно много людей при этом гибнет. Вот и появилось инженерное решение: вместо боя проводить голосование, чтобы определить, чьих сил больше[92]. Можно это считать «эволюцией», но само решение это было придумано и одобрено людьми, так что смело можно его считать одним из решений социальной инженерии. Но для решения проблемы может быть много разных вариантов, сложность общества при этом выросла и проблемы никуда не делись, при демократии просто проблемы другие! Но при смене власти зато люди уже не умирают просто для того, чтобы власть сменилась, монархий на Земле осталось не очень много, а во многих случаях монархи перестали иметь реальную власть, а власть оказывается выборной. Конечно, этот «вирус бескровной смены власти» добрался не до всех мозгов на планете, но прогресс очевиден. А для чего менять власть? Карл Поппер даёт ответ на этот вопрос: если власть всех устраивает, то это ОК. Если уже не устраивает, при прошлом выборе была ошибка, то демократия даёт бескровный способ устранить ошибку. Механизм коррекции ошибок общественного выбора — вот что главное в демократии!

Инженерные решения как «умные мутации для техно-организма» опираются на знание того, что противоречия/конфликты в системных уровнях неизбежны, но оптимизация этих противоречий/конфликтов для всех системных уровней, ведущая к общему уменьшению неустроенности/фрустраций за счёт роста сложности систем, может быть достигнута за счёт инженерного проектирования, а не за счёт проб и ошибок эволюции.

Системное мышление ничего не говорит про то, как снимать противоречия. В нашем руководстве нет никаких «методов творческого мышления», таблиц решений, способов проводить мозговые штурмы, приёмов развития воображения (но они могут быть в других руководствах. Например, приёмы развития творческого воображения используются в рамках прикладных методов системной инженерии под общим названием ТРИЗ).

Чудес не бывает, думать при решении проблем тут приходится не меньше и не больше, чем в любых других школах мысли. Системное мышление позволяет понять и принять неизбежность проблем за счёт неустроенностей/frustrations от конфликтующих взаимодействий разных системных уровней, а также при решении

---

[91] https://www.youtube.com/watch?v=Bf4hPlwU4ys и больше ссылок о его идеях в https://ailev.livejournal.com/1622346.html

[92] https://ailev.livejournal.com/1616855.html

этих проблем удерживать ви́дение всей системы в целом как на несколько уровней над ней, так и на несколько уровней внутри неё. Системное мышление позволяет при решении проблем не терять за деревьями леса, не терять за листьями дерева, не терять лист дерева за растительными клетками.

Системное мышление позволяет целенаправленным образом находить противоречия, требовать их решения, документировать эти решения. Подвести к важному противоречию, не пропустить его, не дать проигнорировать — вот задача системного мышления. А дальше нужно брать голову (а часто и моделер в компьютере) в руки и думать, используя разные другие методы.

Как это работает? В реальном мире агентам не очень понятно, как нарезать его на объекты. Вот метафорический пример — попробуйте назвать те объекты, которые вы видите и указать важные объекты:

Дальше работает цикл из быстрой генерации догадок и медленной трудоёмкой их критики. При этом генерация догадок/объяснений/знаний/теорий идёт в форме порождающих моделей — по таким моделям можно предсказывать будущее

состояние мира, а критика идёт по дискриминативным моделям, по которым можно говорить о принадлежности к какому-то классу (и тем самым находить «логическое противоречие»). На картинке цикл познания/творчества, который подробней объясняется в руководстве по интеллект-стеку, хотя и без этой картинки. Впрочем, разные его элементы обсуждаются в разных руководствах программы рабочего развития, тут мы только заметим, что на шаге формализации в качестве важных объектов берутся именно системы, суть системного подхода главным образом во влиянии на этот шаг в цикле:

Это всё значит, что системное мышление подразумевает не только неформальные рассуждения в рамках коннективистских/нейросетевых/S1 представлений и их выражения в рамках теории прототипов (метафорами), но и вполне формальные рассуждения с использованием знаковых представлений теоретической теории понятий, в которой мы говорим об объектах и отношениях, и в которых все объекты и отношения имеют типы. Не обольщайтесь, что текст нашего руководства на естественном языке: в тексте руководства довольно жёсткая типизация всех использующихся понятий. Так, мы не путаем «метод работы» и «работу», «систему» из системного подхода и «систему» из системы Станиславского. В системном мышлении примат работы объектами и отношениями с задействованием логики, а не метафор и аналогий с задействованием художественности. **Следите за типами объектов и отношений, они важны! При этом учитывайте, что вы можете сами не замечать, что у вас проблемы с «машинкой типов»: если вас кто-нибудь не проверит на эту тему, вы можете заблуждаться, что у вас всё в порядке. Многие инженеры-менеджеры с удивлением узнавали на разборах/review с нашими мастерами=наставниками, что их машинка типов по факту не работает! Поэтому следите за типами объектов и отношений, но убедитесь сначала (не изнутри себя, а получите внешнюю оценку), что это слежение у вас работает!**

Иногда в системном творчестве выделяют особо «изобретение» — какую-то систему

разбивают на части (выделяют в ней системные уровни) разными способами, один из которых обязательно включает функциональное описание (поведение, которое ожидается от системы в момент её использования), а другой — конструктивное описание (те материальные/конструктивные объекты из многочисленных объектов физического мира, которые можно приспособить для воплощения функциональных объектов из функционального описания: аффордансы/подходяшки) при ограничениях со стороны пространственного нахождения этих частей и стоимости сборки аффордансов в работающую систему.

Это иногда называют и системным творчеством, и «модульным синтезом» (аффордансы — это модули), и изобретением. В любом случае, системное мышление позволяет подробно обсуждать эти «изобретения», подробно обсуждать суть творчества.

Интересно, что в инженерии обсуждают и «защиту от дурака»: невозможность неправильного задействования каких-то предметов. Системное творчество тогда заключается в том, чтобы «подходяшка» не только подходила для нужной функции, но и не подходила для ненужной[93]:

[93] https://sketchplanations.com/affordance

В любом случае, системное мышление — это прежде всего творческое мышление. Конечно, это не безалаберное творчество, случайные придумки. Нет, случайности (в основе эволюции всё-таки лежат мутации) в системном мышлении место есть, но мы стараемся, чтобы это были smart mutations/продуманные мутации, с наибольшей вероятностью ведущие к успешности системы[94].

Вопросы: аффордансы.

Олег задумался о том, что надо бы позавтракать. Заглянул в холодильник, а там только борщ и огурец. Взял в холодильнике огурец и съел его. Что тут аффорданс для завтрака (завтрак-— порции пищи, которые обычно съедаются утром)?

--- Олег. ### Нет, Олег тут система-создатель для завтрака (а затем будет целевой системой, для которой сам завтрак входит в систему создания, и тогда завтрак — аффорданс для поддержания жизни Олега, но это могла бы быть и капельница с глюкозой!).

[94] https://arxiv.org/abs/2206.08896

--- Холодильник. ### Нет, холодильник — это не завтрак, это часть системы создания для завтрака.

--- Борщ. ### Нет, борщ не подошёл Олегу в качестве завтрака. Аффорданс — это «подходяшка», подошёл в качестве завтрака только огурец.

+++ Огурец. ### Да, роль завтрака исполнил огурец, это и есть «подходяшка».

Задание: эволюция.

Поставьте отметку о достижении состояния:

1. Написан пост, в котором вы трактуете упомянутую в руководстве работу «Toward a theory of evolution as multilevel learning» в контексте своего рабочего проекта.

Предметные специализации системного мышления

Системное мышление развивается по мере его употребления как в «истинных», естественных науках (которые вырабатывают объяснительные теории по поводу мира — физика, химия, биология и т.д., на английском sciences), так и в разных инженериях: от самой общей системной инженерии до прикладных её вариантов — инженерии предприятий (менеджменте), системной химии[95] (которая изучает не столько поведение молекул в реакциях, сколько поведение множества самых разных молекул в реакционных сетях, в том числе биохимических системах — и выходит на инженерию таких сетей реакций), системной биологии[96] (которая выходит за пределы редукционистской системной химии в изучении биологии и учитывает более высокие уровни организации существ — макромолекул, клеток, тканей, органов, организмов, популяций — и тоже с прицелом на инженерию, в том числе генную инженерию), инженерии личности (самые разные методы, связанные с обучением вменяемых систем различным видам мастерства, этому посвящён отдельное руководство по инженерии личности).

Системное мышление пробуют применить и в социальной инженерии, зачастую безуспешно, хоть и в опоре на социальные науки. Социальные науки обычно являются нормативными, то есть не столько объясняют, сколько предписывают, «как надо», и ещё эти науки не учитывают системного/многоуровневого характера происходящих явлений и наличия неснимаемых простыми методами неустроенностей, происходящих именно от многоуровневости системной организации социума — личностей, организаций, сообществ, обществ. Можно привести в пример только одну из утопических идей социальной инженерии: строительство коммунизма, совершенно неслучайно заканчивающееся каждый раз неудачей в силу слабости его теоретических посылок[97]. Тем не менее, системное

---

[95] https://en.wikipedia.org/wiki/Systems_chemistry, см. пример создания размножающейся искусственной бактерии, ДНК которой дополнен несуществующими в природе основаниями https://www.bbc.com/news/science-environment-27329583

[96] https://en.wikipedia.org/wiki/Systems_biology

[97] См., например, критику теорий социализма/коммунизма в https://www.sotsium.ru/ekonomicheskaya-teoriya/socializm-ekonomicheskij-raschet.html, В этой работе Х. Уэрта де Сото опирается на теорию невозможности экономического расчета при социализме, разработанную ещё в 1920-х годах Людвигом фон Мизесом и Борисом Бруцкусом, концепцию неявного и неартикулируемого знания Майкла Полани, теорию предпринимательства Мизеса-Кирцнера, концепцию стихийного порядка Хайека и идеи последнего о роли знания в экономике.

мышление пробуют задействовать в том числе для заведомо ущербных инженерных идей, хотя ограничения такого использования встроены в системное мышление — и по мере смены поколений системного мышления они исключают совсем уж грубые ошибки. Пока же советуем относиться к идеям инженерного переустройства общества с большим подозрением: благими намерениями тут вымощена дорога в ад. Всегда надо помнить, что в концентрационных лагерях газовые камеры были самыми продуманными в части инженерной мысли, всё остальное (бараки, дороги) были погрубее. Так что осторожней с использованием системного мышления для обустройства того, как по вашему (или ваших заказчиков) мнению должны жить люди.

Есть и другие, менее распространённые специализации системного мышления в самых разных областях инженерии, где слово «инженерия» традиционно не используется, но по факту это именно системная инженерия, причём опирающаяся на системное мышление. Например, есть специализация системного мышления для танцевальной импровизации (инженерии танца) Viewpoints[98], на системном подходе также основан текст «Танцевальное мышление и его развитие»[99].

Во всех этих многочисленных прикладных инженериях, базирующихся на системном мышлении, накапливаются знания по типовым инженерным («железной» и программной инженерий, менеджмента, танцевального творчества, государственного строительства и т.п.) решениям для какого-то одного или даже нескольких видов систем какого-то определённого одного или даже нескольких системных уровней.

Но когда вам нужно что-то делать впервые в мире (как когда-то летели на Луну, а с 2015 в SpaceX делают первые возвращаемые на Землю и с 2017 года повторно используемые ракетные системы, или пытаются создать распределённые автономные организации DAO[100], или с ноября 2022 года в OpenAI делают AI на основе больших языковых моделей[101]), то есть два варианта — изобретать что-то беспорядочно, «по интуиции», или мыслить системно, чтобы как-то последовательно ставить и решать проблемы, находить и устранять противоречия (системное творчество), снижать риск забыть что-то важное в многолетнем проекте (системное моделирование и основанное на нём управление конфигурацией[102]), выполняемом сотнями и тысячами человек.

Системное мышление, используемое в методологии, помогает поделить решение какой-то проблемы между разными агентами (например, людьми) в команде, организовать выполнение работ по методу/«виду труда» по разными агентами, специализирующимися в подметодах/«подвидах труда». Это **разделение труда**: развитие той линии, по которой появилась английская мануфактура (когда вместо ремесленного производства, где все операции изготовления булавки вёл один человек, выполняющий почти три десятка различных операций, стало производство со многими людьми разной специализации, каждый из которых выполнял только операцию одного-двух-трёх видов) и тейлоровский заводской конвейер (в котором

---

[98] https://en.wikipedia.org/wiki/Viewpoints
[99] http://ailev.livejournal.com/1332624.html
[100] https://en.wikipedia.org/wiki/The_DAO_(organization), это был 2016 год, после этого было множество случаев повторения такой конструкции с разными вариациями, https://en.wikipedia.org/wiki/Decentralized_autonomous_organization.
[101] https://en.wikipedia.org/wiki/ChatGPT
[102] https://sebokwiki.org/wiki/Configuration_Management

каждый рабочий должен был делать буквально одну посильную ему операцию, в пределе даже инвалиды могли работать на конвейере, настолько сильно был разделён труд). Разделение труда включает и выполнение некоторых методов работы, изначально массово выполнявшихся людьми, машинами (автоматизацию). Системное мышление позволяет как-то ориентироваться в возникающем разнообразии систем-создателей (предприятия с огромным числом людей, AI-агентов, оборудования), которые создают какую-то сложную продукцию — ракеты или ледовые буровые установки.

Часто инженерное решение и его реализация принципиально не могут быть получены в разумные сроки одним человеком или компьютерным AI-агентом, даже если речь идёт о гениальности человека или сверхгениальности AI. Требуется работа больших коллективов из множества организованных людей совместно со множеством усиливающих их возможности компьютеров и другого оборудования, а также с добавкой AI-агентов.

Для этого агенты, выполняющие роли «железных» инженеров и инженеров-программистов, менеджеры/инженеры организаций/enterprise engineers, инвесторы, танцоры[103], члены правительства или избирательного штаба, другие члены той или иной команды проекта/организации/предприятия/«органа власти»/«общественной инициативы» (различия тут есть, но в данном случае не важны — это всё «организации» как создатели систем самого разного системного уровня) явно обсуждают метод своей работы, занимаются методологией. При этом они не просто «генерируют основные инженерные, менеджерские, творческие, политические, художественные и иные решения», а задействуют методы системной инженерии, то есть в ходе изобретательской своей деятельности разрабатывают полноценную **концепцию использования системы, концепцию самой системы** и ряд других моделей системы и моделей систем создания этой системы. Основанный на системном подходе профессиональный язык прикладных инженеров, менеджеров и даже танцоров, позволяет быстрее, чем на бытовом языке, договариваться о том, что в каком порядке делать при постановке и решении многочисленных проблем в ходе создания самых разных систем — космических кораблей, организаций, танцевальных перформансов, профессиональных объединений, стран, т.е. всего того, что делают люди, сегодня поддерживаемые AI-агентами и изощрённым оборудованием.

Системное мышление ничего не говорит про содержание мышления (какие действия с какими именно системами продумываются), оно касается только его формы — использования концептов системного подхода как типов объектов внимания, которые будут использоваться в мышлении.

Скажем, тип «система» (из нашего руководства системного мышления, мета-мета-модели мира). Вам нужно понять, какой тип объекта в вашей предметной области (мета-модели мира, из учебника по этой предметной области) ему соответствует. Эта система — киберфизическое устройство? Существо? Вменяемый человек? Организация? Сообщество? Общество? Их какие-то части? Как в учебнике этой предметной области называют эти объекты, которые в руководстве по системному

[103] Пример такого танцевального проекта, использующего системное мышление — системное моделирование социального мультиданса, https://vk.com/buffdance

мышлению называют «система»?

В рабочих проектах нужно также разбираться, какие объекты в реальном мире (экземпляры) соответствуют этим типам «из учебников», то есть отмоделировать экземпляр.

Системное мышление управляет вниманием через наведение его по верхним уровням иерархии типов/классификации/онтологии, уровни выше уровня предметных областей (понятие «система» из мета-мета-модели, но не мета-модели прикладной системной области — там будут «ракета», «растение», «мастерство не бояться пауков»). Сами объекты внимания могут быть из абсолютно разных прикладных предметных областей, определяемых абсолютно разными мета-моделями (уровня учебников по объяснениям/теориям прикладных методов и уровня сложившегося понимания в конкретном предприятии в конкретном проекте). Но все эти предметные мета-модели реального мира будут типизированы объектами более абстрактной мета-мета-модели реального мира из нашего руководства. Это будет не просто «дверь закрыта», а «дверь::система закрыта::состояние». И всё это моделирование (системное моделирование!) обязательно надо документировать/записывать, то есть удерживать внимание техническими средствами, а не надеяться на то, что вы это проделываете «в уме». Если вы не поняли содержание текущего абзаца, то вы плохо освоили наставления руководства по рациональной работе в части моделирования. Работа на разных уровнях абстракции (разных уровнях мета-моделирования) и есть «связь системного мышления с жизнью», заземление/grounding системного мышления!

Системное мышление в этом плане фундаментально/трансдисциплинарно: оно не затрагивает понятия прикладных дисциплин из прикладных методов работы, но даёт типы важных объектов для понятий этих дисциплин. Если прикладной метод работы (помним, что это дисциплина/знание/теория и инструментарий её поддержки) — кулинария (теория/дисциплина изложена в учебниках кулинарного искусства, инструментарий реализован кухонной утварью), то целевой системой может быть, например, борщ. Само понятие борща — оно из предметной области кулинарии как метода работы со своими знаниями/объяснениями и инструментами, но то, что борщ будет типа «целевая система» в каком-то проекте — это уже из системного мышления.

Если в мета-модели есть борщ::система, то в мета-мета-модели есть отношение часть-целое между борщом и надсистемой. Типы «система», «надсистема» — они из системного мышления. Что тут будет надсистемой для борща? Скажем, надсистемой будет званый обед в ресторане (а борщ уточняем: это название одной порции борща, а не вообще тип «борщ» как тип кулинарного изделия). А ещё в мета-мета-модели есть отношение создания между целевой системой и системой-создателем. И тогда я неминуемо спрошу, что будет системой-создателем для борща? Борщ::система создаётся какой-то системой-создателем, этот ход мышления не из кулинарии, а из системного мышления. И дальше я могу предположить, что это шеф-повар::система-создатель (а не домохозяйка или кашевар, ибо я уже знаю, что этот борщ как порция будет съеден в ресторане на званом обеде, а не дома или где-то в поле). Системное мышление не даст мне забыть о шеф-поваре, дальше можно будет задавать вопросы о степени мастерства шеф-повара в кулинарии.

Если я сначала начну думать не о пище, а о том, кто готовит пищу::система-создатель, то системное мышление не даст забыть о том, что именно готовится, заставит подумать о пище::целевой-системе, и дальше с необходимостью поведёт сначала не внутрь, в подсистемы, в состав пищи, а в надсистему: задаст вопрос о том, что будет окружать эту пищу в момент использования. И если это организм едока::надсистема, переваривание::использование, то нужно будет подумать о диетических характеристиках пищи::система — и тот, кто готовит пищу::система-создатель должен будет выполнять какие-то методы диетологии/нутрициологии, возможно даже методы диетической геномики/nutritional genomics[104]. А если в системном окружении целевой системы обнаружится ситуация званого обеда в ресторане::надсистема, то всё-таки это будет шеф-повар::система-создатель.

Системное мышление не пустит думать о бликах солнца на стенах кухни или звоне кухонной посуды, особенностях обеденной скатерти и прочем не слишком важном. До этого, может, когда-нибудь и дойдёт дело, но не сразу. Сразу системное мышление заставит подумать о целевой системе в момент использования/работы/функционирования её в составе надсистемы «как она работает, что делает с надсистемой» (например, борщ приводит едока во вкусовой экстаз, а потом поставляет нужные вещества организму для переваривания). После этого системное мышление заставит подумать о составе целевой системы (из чего и как делаем, какие методы/практики/стили/культуры работы задействуем), а затем о системе создания в момент создания (кто будет выполнять эти методы работы).

Если эта последовательность рассуждений **«сначала целевая система в её окружении, потом из чего она сделана, потом система создания»** как-то будет нарушена (например, разговор начнётся не с целевой системы в момент использования, то есть не с борща в момент его поедания или в момент переваривания), то системное мышление просто заставит обратиться к этому рассмотрению целевой системы в её окружении (в составе надсистемы) как можно раньше — и не пустит в рассуждениях дальше. Это и есть мощь системного мышления: оно заставит вас продумать важное, не даст забыть о важном.

Мир составлен из каких-то систем, их множество — но в мире выделяют вниманием особые системы, «целевые», и дальше используют «целевые системы» как опорные для внимания, разворачивают переход внимания с рассмотрения одной системы на рассмотрение другой системы по отношению к целевым системам: от целевой системы вверх и вниз в момент её эксплуатации по системным уровням, а в момент изготовления от целевой системы по пути в графе создателей. В любом случае, внимание проходит по каким-то системам, отталкиваясь каждый раз от целевой системы проекта как начале рассмотрения.

Понятия системного подхода из мета-мета-модели, взятые как типы, управляют мышлением. Эти типы, как и последовательность рассуждений с ними, берутся из нашего руководства, выучиваются один раз, затем используются во всех предметных областях: выращивании репса, производстве заклёпок, обучении сотрудников, организации сообществ, разработке AI-систем.

Когда вы начинаете заниматься чем угодно, вы уже довольно много знаете про это «что угодно»! Вы знаете, что там будут объекты типов из системного подхода, которые будут заставлять вас обращать внимание на

[104] https://en.wikipedia.org/wiki/Nutritional_genomics

важные аспекты ситуации и даже подсказывать, о чём думать сначала, а о чём думать потом. Например, всегда сначала надо думать о целевой системе в её окружении, а если её не нашли, то её надо искать!

Если прикладной способ/метод работы — квантовый компьютинг, то целевой системой может быть квантовый компьютер. И там повторится ровно это же рассуждение: вы должны будете сначала рассмотреть квантовый компьютер в момент его использования, затем понять, из чего он состоит, чтобы быть так использованным, затем — как сделать этот компьютер (какие там методы его создания), затем — какая система-создатель будет у этого компьютера. И как бы вы не уворачивались от ответов на эти вопросы (а это очень трудные вопросы!), системное мышление заставит вас на них ответить. Если ответов нет и не предвидится, то это «дохлый проект», бегите из него. Если ответы могут быть получены, то их получение — это и есть системное мышление в ходе проекта. **Системное мышление поможет не забыть о важном, оно будет управлять вашим вниманием.**

Если прикладной метод работы — госстроительство, то системой в предметной области (domain) этого метода работы может быть министерство нападения (ни одно государство своё министерство нападения не назовёт в целях пропаганды иначе, чем министерство обороны, но мы-то можем тут назвать вещи своими именами!). Это означает, что вас системное мышление буквально заставит подумать о том, что делает министерство нападения в момент его использования (кем использования? Системное мышление не даст возможности уйти от ответа на этот вопрос. Не позволит, например, сказать «народом». Подробней об этом будет дальше в нашем руководстве). Системное мышление заставит подумать о том, как устроено это министерство нападения внутри, из чего сделано. И дальше заставит подумать, какими методами оно создаётся и кто будет создавать (кто умеет выполнять работу такими методами?). Это всё буквально «вопросы из руководства по системному мышлению», не вопросы по курсу госстроительства. Один наш инженер-менеджер сказал: «в ходе стажировки по системному мышлению я получил задание, затем я задал начальнику по этому заданию вопросы из руководства — и тут же получил место в правлении компании, ибо знаю, о чём спрашивать, что самое важное в окружающем бедламе». И это не единичная, а типичная история.

Системное мышление будет управлять вниманием к объектам прикладной инженерии, в которых причинно-следственные связи имеют объяснительные теории из самых разных прикладных методов работы. **Системное мышление будет указывать на важное, о чём нельзя не подумать. Вы обязаны будете указать целевую систему, указать её надсистему, предположить деление на части целевой системы минимум четырьмя способами, указать систему создания или даже целый граф создателей, и т.д.** При этом порядок этого перечисления имеет значение. Если вы начали с системы создания, например, начали описывать структуру фирмы, то мастерство системного мышления сначала спросит, какую систему создаёт фирма — и вас будет интересовать момент, когда эта система работает у клиента, а не момент, когда её фирма делает. Но потом вы обязательно спросите (себя или участников обсуждаемого проекта), как именно делает эту систему фирма. А потом вы обязательно спросите и про структуру фирмы, но только после ответов на предыдущие вопросы. Ибо если не знаешь, что фирма делает, никакая структура фирмы не поможет тебе оценить, хорошо ли

устроена фирма, чтобы это делать. Подробней о том, чего нельзя не делать системному мыслителю, о чём ему нельзя не думать, будет в нашем руководстве дальше.

Развиваемые на основе системного мышления системная инженерия и её прикладные варианты (программная инженерия, робототехника, инженерия личности, системный менеджмент, социальная инженерия и т.д.) делают всё, чтобы в каждом рабочем проекте не нужно было много придумывать, каждый раз «изобретать велосипед». Нет, цель как раз в максимальном использовании лучших известных концептуальных (какие функциональные части какими конструктивными частями/аффордансами реализовывать) решений для неизбежных проблем, возникающих от конфликтов между системными уровнями (надсистема-система-подсистемы — и дальше по этим уровням вверх и вниз). Эти конфликты ведут к неустроенностям в этих системных уровнях. Инженеры придумывают разные оптимизации, устраняющие по максимуму эти конфликты между уровнями.

Это же относится и к системам создания, методы их работы (прикладные варианты инженерии) тоже решают какие-то проблемы. Системная инженерия, поддерживаемая системным мышлением — нормативный метод работы. Системная инженерия — это метод того, как надо «как нужно/предписано делать, чтобы получить успешную систему», это не просто метод размышления о том, как бывает устроено выполнение работы (метод размышления о том, как работать по методу — это методология). Если вам поручено создать хоть скамейку в парке, хоть завод по производству сверхчистых полупроводников, вам не надо с нуля изобретать то, что надо при этом делать. Это уже в самом абстрактном виде сказано в руководстве по системной инженерии, метод эволюционной системной инженерии для систем самой разной природы не надо изобретать самостоятельно. Лучше потратить силы на то, чтобы конкретизировать этот метод на условия вашего проекта.

Не нужно изобретать с нуля и то, что «модели системы обязательно нужно документировать» в самых разных предметных областях. Например, нужно документировать стратегию предприятия как одну из моделей системы. Например, «стратегия предприятия» — это три разные модели (подробней об этом будет в руководстве по системному менеджменту). «Обязательно документируйте стратегию» — не надо заучивать это наизусть специально для стратегии предприятия. Это просто часть моделирования, выучивается один раз, а для моделей предприятия (или робота, или личности, или цветка, или сообщества) это просто применяется (не изобретается!), это общее знание мета-мета-модели из наших руководств, это знание интеллект-стека, а не конкретных прикладных методов работы, прикладных культур инженерии, менеджмента, продвижения продуктов и т.д. Но как лучше записывать модели — вот это можно уже уточнять в каждом конкретном предмете, что-то тут можно и изобрести. Но если у вас сильный интеллект, то есть вы образованы и используете системное мышление, то вы не сделаете ошибки, не будете моделировать только «в уме» а потом ужасаться от неизбежных ошибок такого моделирования. Вам не нужно будет из опыта приобретать бесценное знание «модели должны быть не в головах, а отчуждёнными. Не записано — не модель». Это знание можно просто взять и использовать.

Это наставление про важность документирования моделей на отчуждённом от мозга разработчика носителе было уже многократно обсуждено в разных проектах, об этом, конечно, напоминается в руководстве по системной инженерии, его нужно

просто задействовать: если у вас инженерный проект, то модели к нему вам надо обязательно документировать, потратить на это время и силы, как и указано в руководствах! Если не выполнять такие наставления, то будет плохо, про такие «новичковые ошибки» в системном мышлении и системной инженерии заранее известно. Нужно экономить мышление и сразу документировать модели, не ждать, пока в проекте будут проявлены эффекты конфигурационных коллизий, возникающие от халатности в документировании.

Особенности документирования моделей системы, иногда на уровне типа этой системы, иногда даже на уровне экземпляра — вот тут надо будет привлекать знание прикладной предметной области, а также думать. А до этого момента можно экономить рассуждения, просто действовать по наставлениям руководства, не изобретать другой способ работы — это и есть лучший на сегодня метод: «модели обязательно документировать». Конкретный вид модели и приёмы документирования моделей системное мышление и системная инженерия не подскажут, системная инженерия даст самые общие представления о необходимости разработки некоторых типов моделей, а подробный рассказ о моделировании будет только в руководствах по прикладным методам работы в каких-то определённых предметных областях. Например, для моделей орг-стратегии будет раздел в руководстве по системному менеджменту, но и там особенности документирования будет предложено додумывать самому, ибо неизвестно, например, какой универсальный моделер будет вами использоваться в условиях конкретного оргпроекта.

И таких наставлений (читай: предписаний, обязательных для следования им в работе) в системной инженерии и её прикладных вариантах много. Системная инженерия — нормативный метод, она предписывает, что надо делать, это не просто «системная инженерия — метод инженерии, который иногда бывает в проектах». Не надо изобретать системную инженерию с её специфическим современным разделением труда, можно просто следовать руководству — и использовать это руководство как норму/предписание, обязательное наставление! Надо экономить мышление, не изобретать методы, которые слишком сложны, чтобы с первого раза вы их изобрели удачно, и их слишком много, чтобы у вас было достаточное время для разбирательства с каждым вновь изобретённым вами вариантом!

Вам не нужно придумывать с нуля, что является целью операционного менеджера, который по факту эксплуатационный инженер предприятия. Его целью является увеличение прохода/throughput работ и исходных материалов в выходные продукты и сервисы: чем большее их число за единицу времени уйдёт из предприятия, тем больше навстречу будет денежный поток за продукты и услуги. Это уже давно обсуждено, все копья на эту тему были поломаны, на лишнем мышлении тут может быть сэкономлено, вряд ли вы (даже с вашей командой) так гениальны, что придумаете что-то лучше.

Но вот как именно увеличить проход/выпуск в конкретном предприятии (скажем, предприятии по разработке программных средств, использующихся в робототехнике, или маркетинговом агентстве, взявшемся увеличить впятеро клиентуру какого-то банка), вот над этим нужно много, долго и тщательно думать, а потом-таки реализовывать придуманные меры по его увеличению. **Экономим на одинаковом: берём лучшее из методов мышления на верхних уровнях абстракции (мета-мета-модель как типы трансдисциплин интеллект-**

стека; эту мета-мета-модель дают руководства по рациональной работе, системному мышлению, методологии, системной инженерии), а потом берём лучшее из методов прикладной инженерии (руководства по инженерии личности, системному менеджменту, подставьте дальше руководства по методам работы в вашей предметной области — а хоть и корпоративные стандарты/регламенты рабочих процессов,), и только затем «включаем мозг»: про особенности вашего проекта вы нигде прочесть не сможете, над ними придётся думать.

Полная творческая мощь системного мышления будет проявляться только в тот момент, когда известных типовых решений не будет и нужно будет делать **первую из нового вида** (first of a kind, FOAK) систему, или обходить какие-то жёсткие ограничения, которые не встречались раньше, или избегать каких-то часто встречающихся ошибок в деятельности — например, не забывать в суете выполнения какой-то работы подумать о чём-то важном, для чего нужно заранее знать — что именно является важным. Так, важным является системное окружение/system environment в момент использования/работы/эксплуатации системы. О нём нельзя забывать подумать. Функциональное разбиение системы тоже важное, оно первично. Методология говорит, что для системы важны ещё и методы создания и развития системы, которые выполняются создающей системой/создателем. Например, если целевая система «мастерство», то создание и развитие нового мастерства у личности проводится создателем мастерства: провайдером обучения, проводящего курс или стажировку на рабочем месте. Дальше про мастерство можно думать примерно так же, как о роботе, космическом корабле или борще (хотя и есть особенности создания: робота собирают, а затем обучают, космический корабль собирают, хотя сегодня может уже идти речь об обучении каких-то его AI-подсистем, а мастерству так только обучают, его не соберёшь).

Методы работы переводят какие-то изменяющиеся объекты (их будем называть альфы) по состояниям. Скажем, табуретка проходит через состояния «ничего нет», «сиденье и ножки сделаны», «сиденье и ножки собраны», «табуретка установлена в месте эксплуатации». Не надо придумывать, как об этом думать. В руководстве по методологии уже прописаны правила, по которым можно удобным для отслеживания этих состояний способом называть эти состояния. Меняться может что угодно: табуретка, мастерство, команда проекта, коммерческая возможность проведения проекта. Оказывается, удобней всего называть состояния в их привязке к методам работы создателя с текущим состоянием («табуретка эксплуатируется») или ещё лучше — в привязке к окончанию применения какого-то метода работы, приводящего к состоянию, чтобы указать на то, что с системой уже что-то произошло, прошедшее время: («табуретка введена в эксплуатацию»).

Такие изменяющиеся объекты проекта надо уметь находить (прежде всего это создаваемая и развиваемая целевая система! Но есть и много других объектов — описание целевой системы, команда проекта, внешние проектные роли и т.д.), уметь описывать последовательности смены их состояний и называть их (нет названий — нельзя обсуждать ход проекта, нельзя ведь сказать будет, в каком состоянии важный изменяющийся объект проекта!), уметь подумать о том, какие практики/методы работы могут привести к смене состояния важного объекта в нужную сторону. И нужно не только уметь, нужно ещё не забыть (в том числе не лениться!) сделать это прямо в ходе работы по проекту, в условиях множества

отвлекающих внимание вас и вашей команды событий.

И такого, о чём нельзя забывать думать в стрессовой ситуации сложного проекта, довольно много. Этому посвящены наставления руководств по рациональной работе, системному мышлению, методологии, системной инженерии, интеллект-стеку, других руководств, которые обучают интеллект-стеку фундаментальных методов мышления. А дальше прикладные руководства разного уровня детальности (например, руководство по системному менеджменту, которое даёт общий подход к инженерии организации и его сообществ, руководство по инженерии личности) дополнят эти наставления. Что-то придумывать нужно будет только тогда, когда вы выйдете за пределы этих знаний, столкнётесь с проблемами конкретного проекта.

Речь идёт тут и о простых проектах «на одного» (всё равно это будет ощущаться как коллективный проект, надо будет делать коллектив из самого себя — играть разные профессиональные/проектные роли и как-то устраивать продуктивные споры между разными ролями внутри себя), и о реально сложных проектах, в которых задействованы многотысячные коллективы. Системное мышление спасает такие проекты, ибо управляет вниманием как одного человека, так и тысяч агентов (людей и AI-агентов, коллективных агентов-предприятий), предотвращает глупые ошибки типа «забыл подумать, придётся всё переделать».

В нашем руководстве задаётся только форма для мышления, типы мета-мета-модели: взятые из стандартов и публичных документов и только слегка авторски доработанные в целях лучшего понимания понятия системного мышления. Но в руководстве ничего не говорится про содержание мышления, оно в конечном итоге уникально для каждого проекта. Даже если вы второй раз будете делать какой-то похожий на первый проект, то ваше мышление по его содержанию уже будет другим: вы получите какой-то опыт выполнения проекта в ходе первого раза, у вас будут какие-то новые прикладные мыслительные интуиции, но и окружение проекта для второго раза тоже может успеть поменяться. И результаты вашего мышления и вашего действия могут быть для «повторения» совсем другими, «в одну реку нельзя войти дважды». Вместе с тем, не всё так плохо: рассуждение о проекте на высоких уровнях абстракции будет повторяться, и его типы (типы мета-мета-модели, а то и типы мета-модели) будут управлять рассуждениями на более низких/конкретных/заземлённых уровнях абстракции (прикладные предметные области в их понимании в конкретной ситуации, конкретные объекты в проекте, которые будут менять своё состояние). Но уйти от мышления не удастся: время от времени результаты работы будут не совпадать с ожиданиями «по учебнику», «по руководству/регламенту/стандарту/«описанию рабочего процесса», и тогда надо будет включать мозг (усиленный мозгами других людей и AI-агентами!) и заниматься творчеством, решать проблемы, превращать их в задачи, которые можно просто брать — и выполнять, не задумываясь.

Если вы используете методы системного мышления, которые мы даём в нашем руководстве, вам всё равно потребуется знать не только интеллект-стек безмасштабных «всесистемных» фундаментальных методов мышления (мета-мета-модель), но и вашу предметную область (domain, мета-модель), относящуюся к системам какой-то определённой природы, определённого масштаба, и даже просто заниматься мышлением вне всяких методов: от неожиданностей и проблем в рабочих проектах не уйдёшь. Но системное мышление всё равно потребуется, одних прикладных знаний не хватит. Системное мышление уменьшает

вероятность того, что вы забудете подумать о чём-то важном, особенно пришедшем по линии других методов работы, смежных с вашим прикладным предметом.

При этом системное мышление не скажет, «куда идти, что делать». Мета-мета-модель системного мышления даёт типы системного подхода как «модель», карту. А уж куда вы пойдёте «по карте» (пойдёте по территории, ориентируясь по карте) — это отдельное размышление, это рациональность как выдвижение гипотез о действии, принятие решения о том, каким методом действовать, а затем переход к действию (ибо ещё можно принять рациональное решение о том, что надо делать, а потом просто… не делать).

Скажем, если вы занимаетесь разработкой корпоративного софта, то системное мышление заставит вас подумать не только о предметной области программной инженерии, но и инженерии предприятия, ибо без учёта того, как вы в предприятии будете использовать софт (ход от целевой системы в окружение), вы хорошего софта не сделаете. И вы будете заниматься системным мышлением письменно, будете моделировать, так что это «не забудете подумать» будет даже более надёжно, чем «просто мысли в голове». А потом вы предложите варианты действий (несколько!), прокритикуете их и выберете из них лучший, а затем обязательно будете действовать по этому варианту, не останетесь просто «мыслителем», станете «деятелем».

Моделирование. Предметные области

Заполните таблицу, указав три последних курса (возможно, это были просто пройденные вами учебники или прочтённые монографии, форма тут не так важна) по прикладным методам работы, а также методам мышления (методам усиления интеллекта). Не забудьте указать год создания этих учебных материалов и год, когда вы проходили этот курс.

Номер п/п	Курс	Прикладность	Методы	Год создания курса	Год прохождения курса	Использовано ли знание в рабочих проектах
Пример	Стажировка по рациональной работе	фундаментальный	Операционный менеджмент, понятизация, собранность, семантика, теория понятий, онтология	2025	2025	Да

1.		Фундаментальный				
2.		Фундаментальный				
3.		Фундаментальный				
4.		Прикладной				
5.		Прикладной				
6.		Прикладной				

Можно ли научить мышлению?

Знания, в том числе знания системного подхода, в голову укладываются ступенечками, от простого к сложному. Как выделить такие ступеньки? Что нужно тренировать, какую непривычную мозгу мыслительную работу делать привычной?

Тут мы должны ввести понятие **контринтуитивности**. Мы живём в интуитивно понимаемом мире. Наши мозги ездят по интуитивным, невесть откуда взявшимся мыслительным рельсам «быстрого мышления S1» по Канеману, как трамвай — одним и тем же маршрутом. Мы родились, постепенно откуда-то у нас эти рельсы в мозгу проложились (в случае многих лет работы про это говорят «опыт», в искусствах — «насмотренность», в литературе — «начитанность»), и мышление по ним ездит, и ездит обычно мимо известных сегодняшней цивилизации эффективных современных способов решения задач, делая невозможным решение задач сложных. Ибо в «насмотренность» и «начитанность» новое знание не попадает, его просто вы ещё не видели! Нельзя вспомнить то, что не запоминал! Можно придумать, но не факт, что вы каждый раз будете гениально придумывать что-то действительно эффективное.

Откуда у нас появляются интуиции, откуда мы знаем, что именно так нужно мыслить? Чаще всего мы не можем даже вспомнить, где мы в первый раз узнали, что зубы чистят щёткой с пастой, а не разжёванной щепочкой священного дерева удумбар, как это было принято в Индии. Рефлексия (осознанность по отношению к прошлым ситуациям мышления) заставляет предположить, что могут быть и другие варианты, кроме интуитивных — контринтуитивные.

Вот, посмотрел в окно — а там Земля плоская. Когда нам говорят, что Земля круглая, что мы отвечаем? «Это неправда, посмотрите в окно». Нам отвечают: «вы что, Земля круглая, посмотрите за горизонт…», но мы упорствуем: «Вы рассказываете много всего лишнего, что там за горизонтом, но за горизонтом всем видно, что ничего нет. Не нужно говорить про далёкий горизонт и что за ним, давайте говорить про Землю, вот же она — Земля плоская». Вся жизненная интуиция показывает, что Земля плоская, люди по ней ногами ходят, и уж ноги-то точно знают, что Земля не круглая! Но каким-то людям, которых заботят масштабы не только 10 километров, но и 1000 километров, в голову откуда-то приходит мысль про «Земля — круглая», они начинают так мыслить — и не находят никаких

противоречий в таком мышлении[105]. Через некоторое время выясняется, что кроме Земли ещё и Космос с его вакуумом есть, космические корабли там летают «всё время падая, но никогда не падая». Вот это уже непонятно, потому что при идее плоской Земли летание космических кораблей по кругу с достаточной скоростью, чтобы не падать никогда — это понять невозможно. Появилось расхождение с результатами эксперимента/измерениями. Объяснения про круглую Землю контринтуитивны, они не соответствуют «народной теории» (folk theory) плоской Земли, но её нужно будет принять всерьёз (то есть строить свои проекты на её основе), ибо она даёт предсказания лучше, чем теория плоской Земли.

Понятие «контринтуитивности», в котором можно и нужно услышать «антинародность» и «культурность/практичность/следование методу», важно. Каждый раз, когда появляются проблемы с пониманием того, как работают гении, обладающие каким-то искусством, которое никто не может понять и после понимания повторить и улучшить, можно ожидать найти глубоко контринтуитивные объяснения.

Трамвай мысли у гениев идёт по совсем другим рельсам, нежели проложенным в мозгу большинства людей. Найти эти «другие рельсы» в чужом мозгу гения, проложить их в своём мозгу и пустить по ним свой мозговой трамвай обычно очень трудно.

Гений почему-то, сам часто не осознавая, сделал что-то совсем не так, как все остальные, он просто начал что-то делать в противоречии с интуицией всех остальных, и у него начало получаться. А все остальные действуют интуитивно, «по-народному», «как все», и у них не получается. И пока на уровень сознания гения, или тех людей, которые пытаются отмоделировать мышление гения, не вышло в понятийном мышлении, в чём именно эта контринтуитивность, новые объекты не получили свои имена-термины, вы не можете передать это знание другим людям, не можете никого этому знанию научить.

Вы не можете научить директора по развитию (тут мы даём, конечно, условное название должности) крупного холдинга, танцора или политика, если вы не знаете понятийно на уровне сознания, что он должен делать в ходе своей работы, следуя лучшим методам/практикам/процессам/культуре/стилю/стратегии. А ведь надо научить тому, о чём он должен думать, какие понятия в мышлении использовать. Директор по развитию должен быть научен, он не должен вырасти как-то сам собой, как дерево в лесу. От охоты и собирательства талантов надо переходить к осёдлому земледелию, выращивать таланты. Это инженерия, изменение мира к лучшему. К тому же такой подход позволяет потом автоматизировать труд директора по развитию, переложить на AI часть работы, или переложить часть работы на сотрудников, если пока не нашлось соответствующего AI. Но это возможно, только если понимаешь, какую часть труда можно перекладывать, а какую нельзя, как описать перекладываемую часть труда по развитию предприятия, почему эта деятельность по развитию вообще должна проводиться, в чём её суть, какие понятия будут задействованы в мышлении по поводу развития этого крупного холдинга.

Чем отличается искусство от инженерии? В искусстве — один раз свезло, вдохновение было, получился шедевр. Другой раз не свезло, вдохновения нет — не будет шедевра. В инженерии («железной», программной, генной, предприятия,

[105] https://theflatearthsociety.org/

танцев, личности, сообщества и т.д.) мы так не можем, **нам нужно работать стабильно хорошо, нам нужны лучшие (а не любые) методы мышления, дающие неизменно превосходный результат в типовых случаях. И только от этого уже высокого «типового» уровня качества в типовых случаях начинается искусство, отличающее гениальные прорывные работы в нетиповых случаях. Конечно, и в инженерии не обходится без искусства, но всё-таки в инженерии мы хотим стабильного успеха и чаще всего реализуем это желание!**

Пожелания «не стеснять свободу творчества шаблонами» тут не подходят: люди, массово выдумывающие мыслительные велосипеды, с большой вероятностью получат не самое лучшее решение. Аргументы «творчества вместо шаблонов» верны только для единичных гениев, да и у тех в единичных случаях. В большинстве же других случаев шаблонные мыслительные решения дают высокое качество при минимизации умственных усилий. Вряд ли вы тот самый единичный гений, и у вас тот самый единичный случай «выхода за рамки известного цивилизации», когда вы сможете продемонстрировать высочайшее искусство работы с придуманными лично вами решениями «мимо учебников». Скорее всего, вы не очень понимаете, как именно вы думаете, каким методам мышления следуете. Это нужно срочно исправить: осознать то, как вы думаете, и выучить способы мышления получше.

Опять же, гениями называют не всех «творцов», а только тех, которые предъявляют качество мышления лучше, чем по лучшим известным на текущий момент (state-of-the-art) мыслительным шаблонам — и дальше уже их решения становятся шаблонными, а не гениальными. Эти вновь появляющиеся в цивилизации шаблоны хорошего мышления нужно сразу делать явными, документировать (желательно в форме руководств, регламентов с шаблонами чеклистов, стандартов, описаний рабочих процессов, учебных курсов как адекватной форме для передачи знания о методе, и подумайте, легко ли по ним научить работать тех живых или не очень живых агентов, для которых они были написаны. По регламентам, пошаговым инструкциям и справочникам обычно плохо учиться работать, по ним хорошо только проверять, как уже работают! Поэтому для того, чтобы научить работать, применяют более свободные формы, например, руководства. Впрочем, AI-агенты тут быстро меняют ситуацию).

Выделять шаблоны/«выполняющиеся алгоритмы» в методах/стилях хорошего мышления — это работа **культуртрегера**, описывать их — **методолога**, оформление этих шаблонов в виде учебных курсов — работа **методиста/instructional designer.** Подробнее про эти (и многие другие) типовые роли в инженерии какого-то мастерства в людях и AI см. в руководстве по инженерии личности.

Как передаётся неотрефлексированное, неосознанное искусство или ремесло? Ученик смотрит на десятки, сотни, тысячи работ мастеров, научается понимать сленг профессионалов как научаются родному языку (без учебников и словарей, просто «из разговоров»), постоянно смотрит, как работают настоящие мастера и пытается это копировать — прямо по пословице «обезьянка видит, обезьянка делает» (monkey see, monkey do). Именно про это термин «опыт» (пошутим тут про «двадцатилетний опыт — это однолетний опыт, повторённый потом двадцать раз»),

«насмотренность» (которая у многих была в студенчестве, а дальше они застыли в своём развитии). Далее у трёх из десяти учеников в голове появляются какие-то правильные рельсы для трамваев их профессиональных мыслей, и они начинают мыслить быстро и делают мало ошибок в типовых ситуациях. А у семи из десятка — не появляются, и они продолжают делать много ошибок в типовых ситуациях. Обучение искусству или ремеслу — это не «передача понятийного знания», не используется вменяемость человека.

Нам надо, чтобы какому-то новому методу (скажем, методу системного мышления) могли научиться девять из десяти инженеров-менеджеров. Вполне можно представить, что будет один из десяти таких людей совсем неспособный (сильно недоученный в детстве, «из джунглей»), или крайне ленивый, или просто не выделивший достаточного времени на обучение в силу каких-то непреодолимых внешних обстоятельств (например, заболел), поэтому научим не всех десятерых даже в лучшем случае. Но вот научение всего троих из десятка инженеров-менеджеров нас не устроит, надо научить больше, девятерых! Это означает, что мы должны взять для обучения такое контринтуитивное знание, которое само не может быстро прийти в голову ученикам, сделать его минимальное компактное и понятное описание (понятийное! Чтобы можно было выразить словами!), а затем его каким-то образом передать ученикам, чтобы оно встроилось им в голову. Вопрос: бывает ли такое в тех областях, которые традиционно считались «искусством», и которым считалось, нельзя научить рационально? Да, бывает, сплошь и рядом! Это и есть **путь западной цивилизации: превращать «искусство» (в том числе искусство мышления) после его моделирования и рационализации в быстро понятийно передаваемое от человека человеку в ходе структурированного обучения мастерство.**

Когда вы находите правильные понятия и правильные мыслительные операции с ними, и правильные упражнения для ускорения «автоматизации» мышления (повторение ходов мысли помогает перевести медленное осознанное трудозатратное мышление S2 в быстрое нетрудное S1), то после обучения даже не будуте понимать, что вам было трудно делать до обучения. Вы будете мыслить бегло по-новому (S1), но будете неспособны вспомнить, по каким «народным рельсам в никуда» катилось ваше мышление до обучения (S1) и как вы трудно мыслили в ходе обучения (S2). Поэтому мастера (сами вчерашние новички) будут изумляться поведению необученных новичков, в том числе изумляться собственному поведению в период до освоения той или иной культуры/практики/стиля/метода/способа работы. Спросите ребёнка, почему он очень плохо умножал всего год назад — он не сможет объяснить, почему. Сейчас умножение для него вполне естественно, и не требует напряжения всех его умственных сил, как это было год назад. Но дайте ему перемножить матрицы — и всё повторится: сначала это будет непонятно, потом трудно, потом опять плохо будет помниться, что там было непонятного и трудного.

Метанойя

Назовём это свойство прохождения какого-то порога понимания, после которого трудно вспомнить затруднения, **метанойей**. Слово удивительное, попробуйте его написать в разных падежах, да ещё и во множественном числе, получите очень интересные эффекты. Этот термин пришёл из религиозных практик и означает «перемену мыслей», полный разрыв прошлого и текущего мышления. Ты

занимаешься, занимаешься в какой-нибудь семинарии, и вроде как мышление у тебя не так поставлено, как это ожидают от тебя священники. Потом вдруг в какой-то момент щёлк — и ты демонстрируешь всем, что у тебя такое же мышление, как это принято у священнослужителя, с этого момента ты «настоящий», а не притворяешься. Это и называли «метанойя», такой малый западный вариант просветления. Слово «метанойя» рекомендовал использовать вместо слова «обучение» гуру системного менеджмента Peter Senge[106], ибо слово «обучение» с его точки зрения уже совсем затасканное и не означает коренную смену образа мышления в результате обучения. Он позаимствовал его у психологов, которые использовали его для обозначения коренных позитивных изменений личности[107]. Нам не нравятся религиозные коннотации метанойи, но обозначаемое этим словом явление «забывания трудностей обучения» вполне светское.

Когда метанойя произошла, то в новом состоянии его мышления, с «новыми рельсами в мозгу» человеку (ну, или в более общем случае — агенту разной природы) совершенно непонятно, в чём была проблема раньше, «со старыми рельсами мышления». Представим: я знаю, что Земля плоская, я долго спорю, что Земля никак не может быть круглая, но меня в какой-то момент в этом убеждают. И я каждый раз в своих проектах сначала автоматически действую, как будто Земля плоская, потом усилием воли вспоминаю, что рационально вроде бы она должна быть круглая, потом делаю это уже на уровне рефлекса, и при этом вижу тысячу свидетельств этой круглости Земли. И вот в этот-то момент я уже не могу понять, почему я считал, что Земля плоская. Рационально вспомнить, что я когда-то считал Землю плоской, я могу. Но понять, как я перешёл из состояния знания «интуитивной теории» в состояние владения «контринтуитивной теорией» я уже не могу. И поэтому я не могу осознать те учебные действия, которые нужны для того, чтобы я добивался этой метанойи круглости Земли у своих учеников. Но сам факт обращения внимания на эту прошедшую метанойю (вывод её в сознание) даёт шанс разобраться. Работа по составлению правильных упражнений для получения такой метанойи у учеников — это трудная работа, но возможная. Создание адекватного учебного курса по уже имеющейся теории вполне может занять пару-тройку десятков лет, а если кроме самих упражнений нужно создать ещё и новую теорию, то и сотен лет, как это было в случае перехода от птолемеевского к коперниковскому пониманию движения планет. Все эти рассуждения про трудность создания теории/объяснений/знаний и руководства/регламента/стандарта/«описания рабочего процесса», по которому было бы возможно быстро достичь беглости/метанойи в мышлении инженеров-менеджеров в полной мере относятся и к системному мышлению, и ко всем методам интеллект-стека, и к прикладным инженерным методам.

Особое внимание нужно обратить на то, что в случае метанойи речь обычно идёт об обучении не любым методам мышления и действия, но «контринтуитивным», которым мозг сопротивляется особо, он же в этом случае «интуитивно знает», как должно быть, и активно сопротивляется новому знанию! Заново чему-то обучить много легче, но если уж вы уже подхватили где-то «народную интуицию», то научить вас чему-то более эффективному новому будет весьма проблемно: вам придётся пройти метанойю, а это требует наличия как-то документированной модели целевого мышления, а затем организованной в какую-то

[106] https://sardertv.com/metanoia-peter-senge/
[107] https://en.wikipedia.org/wiki/Metanoia_(psychology)

последовательность работ заданий, времени для выполнения всех этих заданий, а также недюжинной **собранности** — ибо вся интуиция вам будет показывать, что учат-то какому-то безумию!

Шансов пройти эту метанойю «самоучкой» практически нет, если вы не гений. Так, вы ни за что не будете учить по самоучителю объяснения круглости Земли, если вы придерживаетесь теории плоской Земли. Шанс у вас тут только попасть на какие-то курсы или пройти стажировку в какой-то организации, где вас буквально заставят ознакомиться с аргументами про круглость Земли, а затем ещё и потренируют (мимо вашей воли! Просто заставят!) на беглость в применении этих знаний в реальных ситуациях.

Контринтуитивность может поджидать в самых простых действиях. В школе учили прыгать через планку «ножницами» — подбегаешь, и прыгаешь. Но если нужно прыгнуть очень высоко, то после разбега к планочке нужно поворачиваться спиной, и прыгать назад-вверх (Fosbury Flop, изобретение 1968 года[108]).

Это абсолютно неинтуитивно, но даёт возможность перелетать и через двухметровую планку. Нужно огромное доверие к тренеру, чтобы вы начали тренировать такой прыжок — ибо в этот момент кажется, что много-много тренировок дадут возможность преодолевать дополнительные десятки сантиметров «ножницами» или «перекатом», что совсем не так. А потом будет метанойя: вы будете не понимать, почему вообще через планку люди ещё где-то прыгают не техникой Дика Фосбери — даже если вы уже не помните, что начало таким прыжкам положил именно Дик Фосбери, и прыгать так люди начали всего на год раньше, чем высадились на Луну в 1969 году. И речь идёт о том, что люди делали тысячелетиями: прыжках в высоту!

В мышлении есть такие же контринтуитивные способы, которые позволяют мыслить по спортивному девизу: «быстрее, выше, сильнее», выводят в «спорт высоких достижений». **Системное мышление — это набор придуманных разными людьми специфических контринтуитивных приёмов мышления, которые позволяют мышлению быть сильней (универсальней и эффективней), чем его предыдущие, «народные» варианты. В частности, системное мышление — это прежде всего эффективный способ управления вниманием в проекте, он позволяет шаблонным образом выделять в**

[108] https://en.wikipedia.org/wiki/Fosbury_Flop

проекте объекты, достойные первоочередного внимания. Вы будете знать, что важно в проекте ещё до того, как им занялись! Понятия системного мышления задают типы важных объектов, которые потом вы должны будете находить в проекте, это позволяет вам не блуждать в надежде, что вы рано или поздно столкнётесь с чем-то важным, а активно искать и находить это важное.

Главная содержательная метанойя системного мышления в том, что вы начинаете думать о мире, как состоящем из вложенных друг в друга по отношению часть-целое систем, которые ещё и взаимодействуют друг с другом, а ещё одни системы создаются и развиваются другими системами, это связь по отношению создания. В третьем поколении к этому добавляется ещё и понимание эволюционного масштаба времени, а также неизбежности конфликтов между системами разных уровней.

Если понимать систему не как «любой объект, который мы рассматриваем», а как «система из системного подхода», то это оказывается крайне контринтуитивным, поэтому требует специального обучения и последующей длительной тренировки такого системного мышления.

Так, «интуитивно» при попытке о чём-то рассказать (например, о борще, о министерстве нападения, о квантовом компьютере) люди рассказывают прежде всего о частях обсуждаемого предмета (составе борща, организации министерства нападения, составе компьютера). В системном мышлении ровно наоборот: обсуждаемый объект-система сразу объявляется частью, и рассказ ведётся прежде всего об окружении в момент функционирования/работы целевой системы. Это ситуация обеда и переваривания пищи в случае борща, ситуация нападения в случае министерства нападения, ситуация использования квантового компьютера: борщ, министерство нападения и квантовый компьютер тут будут только частями своих ситуаций использования. И только после выяснения системного окружения (что окружает борщ в ходе обеда и потом в ходе переваривания, что окружает министерство нападения в ходе нападения, что окружает квантовый компьютер в ходе его использования), после «взгляда наружу от системы» предметом интереса системного мыслителя будет «взгляд внутрь», части системы. Сначала — что происходит снаружи от границы системы, и только потом — что происходит внутри! Это очень, очень контринтуитивно!

В математике термин «интуитивное» часто подменяется термином «тривиальное» — возможность повторения «любым» человеком (с AI математики ещё серьёзно не работали) в данном сообществе, а нетривиальность — невозможность повторения (спасибо за обсуждение этого вопроса математику Роману Михайлову). Демонстрация чего-то интересного в силу нетривиальности делает это «что-то» тривиальным через пару тактов тренировки заинтересовавшихся, ибо в определение «интуитивности/тривиальности» и «контринтуитивности/нетривиальности» неявно входит момент времени «прямо сейчас». Любое «контринтуитивное/нетривиальное» одного поколения становится «интуитивным/тривиальным» для другого поколения мыслителей. Эту «тривиальность» вполне можно добавить в список синонимов к «интуитивности». Интуитивность/тривиальность и контринтуитивность/нетривиальность — это «движущиеся мишени», присвоение этих типов каким-то объектам вполне динамично, результат существенно зависит от времени.

Кто знает, может сегодняшнее системное мышление для будущих поколений людей

и мыслящих машин будет «народным», «интуитивным», «тривиальным» и его надо будет преодолевать, получать метанойю в совершенно по-другому устроенном мышлении. Но пока системное мышление глубоко контринтуитивно в его нынешнем варианте, осваивать его трудно, метанойя оказывается неизбежна.

В наших рассуждениях мы используем и бытовое, и небытовое значение слова-термина «интуитивность». Под «интуитивностью» в быту часто имеют в виду не результат рационального логического рассуждения, а использование «чуйки» — получение результата рассуждения инсайтом, вдохновением, озарением, причём этот результат может быть весьма нетривиален. Мы о таких результатах мышления говорим, что они ровно наоборот — «контринтуитивны», то есть нетривиальны, невоспроизводимы легко разными людьми, эти результаты не относятся к «общенародному знанию», популярным мыслительным автоматизмам, типовой интуиции окружающих людей, а также собственной интуиции. Но в нашем руководстве мы часто обращаемся ко второму «народному» словарному значению «интуитивности», когда говорим о быстром «интуитивном» мышлении S1. Надеемся, что из контекста будет понятно, какое имелось в виду значение в каждом случае использования авторами руководства термина «интуитивный».

Задания. Контринтуитивность и метанойя.

Поставьте отметки о достижении состояний:

1. Написан пост о контринтуитивности и метанойе на примере вашей работы по руководствам рациональной работы и системного мышления (вы освоили уже примерно 20% наставлений по системному мышлению!).

Состояния ученика в ходе обучения мышлению

В обучении системному мышлению инженер-менеджер как ученик/студент/стажирующийся проходит через следующие состояния:

1. Осознал некомпетентность.
2. Заинтересован.
3. Начитан.
4. Понял.
5. Освоил.

Это только пример классификации состояний, критиковать можно по той же линии, что и любые другие системы с «уровнями развития»: трудно подтверждать экспериментально, достижение какого-то уровня трудно подтвердить («освоил арифметику»: сложение и умножение освоил, вычитание и деление — нет. Это «освоил» или «не освоил»? Как сравнить?), сам предмет быстро меняется, и что вчера было «понял», сегодня будет «и не начинал» (физика ньютоновская и квантовая: понял ньютоновскую физику — а вот с квантовой как быть?). Тем не менее, мы можем говорить о какой-то умозрительной последовательности, framework, способе думать — наборе понятий, который позволяет как-то развёрнуто обсудить какой-то предмет, в данном случае происходящее с учеником в ходе обучения мышлению. Обсудить — это или обсудить происходящее в конкретном проекте, или даже спроектировать (обсудить то, что только планируется, что будет происходить).

Эти состояния ученика удобны для обсуждения обучения самым разным

методам/способам работы самыми разными способами: по учебникам, курсам, руководствам, регламентам или вовсе без них, по объяснениям от коллег. Обратите внимание, что состояния сформулированы по уже обсуждённому паттерну именования, наподобие «дверь закрыта», тут вместо двери будет «ученик», а вот что с ним происходило — это и будет дальше в перечислении состояний:

1. Осознал некомпетентность. Вы::ученик поняли (обратите внимание, что не «понимаете», то есть в процессе, а «поняли», это как «дверь закрывается» и «дверь закрыта»), что системный подход вам зачем-то нужен. Это переход от **неосознанной некомпетентности** («я и не догадываюсь, что я не умею системно мыслить») к **осознанной некомпетентности** («я знаю, что я не умею системно мыслить»). **Это самая трудная ступенька на пути к беглости мышления, осознать своё незнание.**

2. Заинтересован. Ученик захотел потратить время, нервы и разные другие ресурсы (например, деньги) на обучение. Это состояние трудно удерживать. Нет мотивации — не будет и вложений труда, не будет hard fun[109] в его определении Seymour Papert, не будет использован fun-критерий[110] David Deutsch, никакой метанойи Peter Senge не случится. На коммерческие курсы взрослые люди приходят уже заинтересованные, и у них дальше всё получается (хотя и не у всех). Юные студенты приходят «на лекции» обычно никак не заинтересованные — и не все из них становятся заинтересованными даже к концу курса. Но если речь идёт о системном мышлении и нашем руководстве, то часто бывшие студенты, познакомившиеся с нашим руководством в качестве вузовского учебника (такое тоже есть, например, в учебных программах МФТИ), возвращаются разобраться с ним второй и даже третий раз. Заинтересованность появляется уже после вуза: чаще всего это происходит после случайного использования приёмов системного мышления, известных по вузовскому курсу — и получения неожиданно сильного результата. В этот момент и появляется желание закрепить успех. Если заинтересованность есть, то будет потрачено достаточное время на достижение результата. Если заинтересованности нет — времени на достижение результата не будет потрачено, результата не будет. Главный критерий, отличающий «освоивших системное мышление» от «не освоивших» — это достаточность потраченного времени, а даже не личная генетическая предрасположенность. Увы, «усидчивость» тоже во многом генетически определена: одному скучно чем-то заниматься много часов подряд, а другому — не скучно, это и даёт «талант».

В классической педагогике у методиста в instructional design выделяют motivational design, и у преподавателя ведущий метод работы — лидерство, то есть умение удержать человека в роли ученика[111]. Если вы сами себе наставник, то есть занимаетесь самообразованием/самостажировкой («учитесь по самоучителю»), то умение собраться и «уболтать себя учиться», потратить время на обучение вашей нейронной сетки — это будет ваше ведущее умение, ведущая способность, «лидерство себя». Можно много лет осознавать, что что-то не умеешь делать, например, не умеешь танцевать, и хотеть этому научиться — но так и не приступить к обучению. Или приступить к обучению — и бросить. Есть много самоучителей

---

[109] https://dailypapert.com/hard-fun/

[110] fun — это когда сам процесс эволюции идей идёт беспроблемно в плане совмещения без «дребезга» явных, неявных осознанных, а также ещё не осознанных идей в рассуждении (рассуждение с явными идеями — это "логика"), https://www.youtube.com/watch?v=uQ2GHzFYxaI

[111] http://ailev.livejournal.com/1316601.html

игры на гитаре, но относительно мало «самонаучившихся». А вот если учишься не сам, а кто-то тебя учит, таких научившихся довольно много.

3. Начитан. Вы знакомы с каким-то фрагментом набора понятий системного подхода. Материал нашего руководства (или какого-то другого руководства, стандарта, регламента, учебника) освоен на этой стадии в части знания значений слов, умения пересказать какой-то фрагмент текста, воспроизвести какое-то рассуждение, поддержать разговор про содержание текста. Например, вы уже сейчас начитанны: вы знаете, что системное мышление имеет дело с частями и целыми, выстраиваемыми во много уровней. Вы знаете, что есть создатели (системы создания систем, системы-создатели). Вы знаете о существовании синонимического ряда «метод/функция/культура/стиль/деятельность/…» и даже можете добавить пару-тройку терминов в этот ряд, а ещё и помните, что нельзя путать метод с «работой», которая ведётся по этому методу. Вы ещё не знаете, что с этим делать и как применить в ваших проектах, но эти знания/объяснения мета-мета-модели из нашей программы рабочего развития у вас уже есть!

Правильно думать о состоянии «начитан» как о том, что ученик начитан учебником по езде на велосипеде. Начитанный о поездках, но ни разу не ездивший сам, ученик может долго вам рассказывать о равновесии, о необходимости крутить педали и объезжать камушки. Но продемонстрировать езду он не сможет.

Начитанность, насмотренность для мышления нужны, в том числе и начитанность-насмотренность примеров рассуждения с высказыванием догадок, их отверждением — примеры обязательны не только «правильных рассуждений», но и неправильных, обнаружения ошибок, примеры коррекции ошибок. Но для уверенной беглости в мышлении начитанности и насмотренности примеров мышления совершенно недостаточно. Текст руководства как раз и даёт начитанность, и в нём даже учтено, что читать будут люди, а не только AI: довольно много материала повторяется в разных контекстах, а не просто излагается как в справочнике. Но «прочесть руководство/регламент/стандарт по езде на велосипеде» даже с большим количеством примеров — этого не хватит для того, чтобы уверенно ездить на велосипеде, демонстрировать наличие мастерства езды на велосипеде.

Начитанность — это даже ещё не переход к **осознанной компетентности**, когда можно самостоятельно и осознанно провести какое-то рассуждение из руководства/стандарта/регламента (или даже в учебном контексте — учебника) на каком-то материале своего (это трудно, ибо множество деталей заслоняют главное) или чужого (это проще, ибо множества деталей не видно, можно сразу обращать внимание на главное) проекта. Из руководства берём при этом типы мета-мета-модели, из рабочего проекта — типы мета-модели и экземпляры для заземления/grounding. Начитанность позволит порассуждать, но не даст беглости в операциях присвоения типов, поэтому затраты времени на эти операции может быть контрпродуктивным. «Хорошая мысля приходит опосля», «задним умом мужик крепок»: эта стадия характеризуется ровно вот этим высказыванием. Начитанным людям приходят в голову хорошие идеи, интересные вопросы, но они приходят не вовремя. В проектах же «дорога ложка к обеду», «кто не успел, тот опоздал». Тут не зря столько разных пословиц и поговорок в объяснении этой стадии.

4. Понял. Студент/стажёр/ученик понял, что означают термины системного

подхода в их многочисленных вариантах разных школ, может продуктивно использовать понятия системного мышления при обсуждении самых разных ситуаций. Кроме памяти тут уже появляются некоторые мыслительные интуиции, часть нужных мыслительных ходов делается «автоматически».

Часть этого умения достигается «**сержантским методом»**[112], то есть путём решения простых и похожих друг на друга, многочисленных **тренажёрных задач**, которые методисты формулируют для тренировки, а не для контроля знаний[113]. Пример такой задачи, сформулированной в виде вопроса с выбором ответа (quiz): «Пётр утверждает, что нужно уже начинать закупать функциональные части системы, а Елена утверждает, что закупать нужно вообще не функциональные, а конструктивные части. Кто из них прав? А) Пётр Б) Елена». Ответить на такой вопрос/quiz можно, только если знать про различия функциональных и конструктивных частей системы — для ответа нужно хоть как-то сопоставить ситуацию в задаче с местом из текста руководства, где говорится о таком различии. После нескольких повторений разных вариантов вопроса ответ будет самоочевидным, никаких отсылок к тексту не потребуется. При решении тренажёрных вопросов как раз и формируются «рельсы в голове», усваивается набор понятий, по которым потом поедет понятийное мышление. Другое дело, что эти рельсы ещё не уложены в путь: по ним ещё не поедешь. Но это уже рельсы, из них можно строить путь — хотя это «строительство путей» отдельная работа от «получения рельсов» как заготовок для рельсовых путей.

Важно, что в этих вопросах/quiz специально тренируется контринтуитивность, отличие предлагаемого способа мышления «по заданному методу» от использования народных/бытовых интуиций/«здравого смысла». Проверка следования заданному, а не «абы какому, из народных верований» методу делается через использование метода **понятийной описи**[114] (concept inventory, иногда conceptual inventory). Метод проверки состоит в том, чтобы обязательно предлагать в замерах понимания студентами/учениками/стажёрами материала учебника/руководства/регламента вопросы с ответами-ловушками, соответствующие «народному мышлению».

Впервые понятийная опись была предложена для оценки понимания ньютоновской физики по сравнению с «народной» аристотелевской. В аристотелевской физике палец давит на стол (он же живой!), а стол не давит на палец (он же не живой!). В ньютоновской они давят друг на друга с одинаковой силой, что контринтуитивно, не соответствует «здравому смыслу» (зато соответствует ньютоновской физике). Вопросы, составленные по принципам понятийной описи, проверяют — что на эту тему после прохождения учебного курса по физике думает выпускник курса. И если оказывается, что он отвечает на вопросы по третьему закону Ньютона с правильным применением формул, но при этом считает, что стол не давит на палец, то что-то в обучении пошло не так, и нужно доучиваться.

---

[112] Подробней о «сержантском методе» можно почитать по ссылкам в http://ailev.livejournal.com/1287293.html.

[113] Основное использование руководства — как регламента для работ в реальном проекте. Эти работы делаются «мышлением моделированием» и «мышлением письмом». Сначала это занимает много времени, а затем всё меньше и меньше, а моделирование происходит не только быстрее, но и точнее. Просто прочтение текста руководства как «текста в учебнике», увы, мало чему учит. Просто прочтение регламента/руководства какой-то сложной работы не научит эту работу выполнять.

[114] http://modeling.asu.edu/R&E/Notes_on_Modeling_Theory.pdf

Вопросы/quiz для облегчения понимания системного мышления составлены таким же образом: они проверяют системность в мышлении, правильность рассуждений с контринтуитивными понятиями системного мышления, отход от бытового и привычного мышления «здравого смысла». Опять же, «понял» в какой-то одной предметной области ещё не даёт общей способности к мышлению. Например, если понял что-то в физике, это не делает тебя умнее в других задачах[115].

Один из стажёров предыдущих версий руководства отметил, что его настроение во время ответа на вопросы/quiz было — «ярость». Ибо его первый ответ стабильно оказывался неправильным, а правильные ответы, оказывается, были чуть ли не точными цитатами из текста руководства, которые были «просмотрены по диагонали», на типы картины мира инженера-менеджера в них не было внимания! И так задача за задачей: задачи заставляли перечитывать текст руководства, вычитывая из него новое и новое содержание.

Какие бы ни были вопросы/quiz, они просто дают возможность запомнить значения терминов и какие-то идеи о связи понятий — не более. На них хорошо будут отвечать «зубрилки», которые хорошо помнят текст учебника. Ситуации в этих вопросах просты, все отвлечения внимания убраны: в поле зрения только нужные для правильного ответа ситуации, а также ограниченный набор «ловушек мышления». Это как в учебнике по езде на велосипеде спросить, где у велосипеда тормоз — и показать несколько правильных и неправильных ответов. Не факт, что студент, правильно указывающий тормоз (он ведь изучал учебник! Видел картинку!), в жизни сможет воспользоваться этим тормозом. При езде на велосипеде всё его внимание и умение будет сначала занимать удержание равновесия, и мысль про тормоз если и придёт в голову, то вряд ли будет реализована.

Ещё одна проблема с учебными задачами в том, что там вопрос сформулирован, а в жизни эти вопросы надо уметь сформулировать — в условиях огромного количества отвлекающих внимание инженера-менеджера мелких значащих и ничего не значащих деталей. Конечно, никакие «тренировки с учебными кейсами» тут не помогут, поможет только моделирование ситуаций из реальных проектов: чужих (где ты не знаком с деталями, поэтому внимания хватает применить материал руководства к важным объектам проекта) и собственных (где тебя как раз отвлекает знакомство с деталями и требуется больше собранности, чтобы сформулировать вопрос перед ответом на него).

Ещё одна проблема с учебными задачами и «понял» — это изолированность фрагментов материала, для которых «знаешь ответ на вопрос». В жизни требуется обычно составить длинное рассуждение, проходящее через понятия, раскрываемые во множестве руководств. В жизни надо не отвечать на отдельные вопросы, а уметь удерживать внимание на длинных последовательностях рассуждений (примером этого являются мантры/канвы/сюжеты, о которых будет рассказано подробней чуть позже).

5. Освоил. Мастер (уже не ученик/студент/стажёр) может системно мыслить и

---

[115] Появились материалы, обсуждающие beyond concept inventories towards measuring how students think — http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2830154/ (concerns about measuring student thinking as opposed to student knowledge, но все эти попытки плохо превращаются в воспроизводимый метод по сравнению с методом concept inventory). Больше на эту тему в тексте "Заметки к "Заметкам по теории моделирования" Давида Хестенеса" — https://ailev.livejournal.com/1197467.html.

действовать (принимая результаты мышления всерьёз, то есть кладя их в основу своего действия по изменению физического мира) in the wild, в реальных проектах, **причём бегло, а не натужно выбирая каждый мыслительный шаг, как в состоянии «понял».** Это совсем отдельное качество: уметь решать уже поставленные задачи (даже олимпиадного уровня сложности) из задачника против умения ставить задачи. **Ставить задачи** — это выделять из запутанного, шумящего, быстро меняющегося окружающего мира с огромным числом незначимых и кричащих о себе деталей правильные объекты, присваивая им типы, описанные в регламенте/стандарте/руководстве/«описании рабочего процесса»/учебнике. После этого можно **решать задачу**, которая сформулирована уже в терминах предметной области рабочей ситуации, причём объекты для решения задачи уже получили типы понятий системного подхода. Впрочем, тут потребуются и типы других трансдисциплин, а ещё надо хорошо будет представлять себе типы объектов самой предметной области — объекты этих типов в жизни тоже найти не просто. Системное мышление — это просто часть полного мышления личности в целом (включая интеллект и прикладное мастерство). Полностью преодолеть какие-то рабочие проблемы только за счёт системного мышления будет невозможно (чудес не бывает), но много легче, чем без него. Системное мышление делает человека умнее, но не настолько умнее, чтобы быстро решать абсолютно все проблемы.

В любом случае, **в жизни нет задач из учебника, нет даже слов из учебника, в жизни «объекты из учебника» нужно выявить, обнаружить, т.е. поставить задачу, «приложить» книжное знание к жизни.** В многообразии самых разных объектов окружающего мира в вашем проекте/вашей ситуации вам нужно будет найти, например, объект типа «целевая система». Но объектов-то тысячи и тысячи, а в больших проектах типа проекта создания и производства авиалайнера — их многие миллионы! Потом, после присвоения типа мета-мета-модели (вы знаете эти типы и для чего они нужны, ибо прошли уже состояние «понял») можно уже в привычных терминах из учебника решить поставленную задачу.

Если нужно узнать, сколько яблок в двух кучках, нужно догадаться, что яблоки — типа «счётные объекты из учебника арифметики», а не типа «фрукт: то, что обычно едят». Если вы хотите описать разрабатываемый вами супер-пупер-продукт, то нужно выявить, какие объекты в мире являются для него надсистемой (то есть присвоить каким-то объектам тип «надсистема» из нашей мета-мета-модели, описанной в руководстве), и какие роли (тоже тип из нашей мета-мета-модели) агентов (людей, организаций) заинтересованы в этой системе, что агенты (чаще всего на сегодня — люди) в этих ролях хотят от этой надсистемы, а не от вашего супер-пупер-продукта! После этого можете выдвинуть гипотезу о том, какая система (назовём её целевой) может выполнить какие-то функции в этой надсистеме, чтобы та удовлетворила интересы найденных вами людей (это всё надо будет найти в жизни: какие-то объекты, которым вы присвоите эти типы из учебника/типы мета-мета-модели). Дальше вы задокументируете гипотезу о целевой системе и надсистеме в момент их функционирования/работы/эксплуатации/использования в виде концепции использования (это тоже тип — тип модели, описывающей поведение целевой системы в её окружении, подробней об этом будет рассказано дальше в нашем руководстве по системному мышлению, а также в руководствах по методологии и системной инженерии). Реальные рабочие проекты в случае руководства появляются с самого начала, но основная польза для этих рабочих

проектов появляются только тут, после появления состояния «освоил» — если и не весь материал, то хотя бы какие-то его составляющие.

Заниматься рабочим проектом приходится прямо в ходе освоения материала руководства/«стандарта лучшей практики»/«описания рабочего процесса»/регламента/учебника. Это отличает рабочий контекст от сугубо учебного контекста. Грубо говоря, будете сразу ездить не на учебном велосипеде, не заниматься учебным проектом, а сразу осваивать езду на настоящем велосипеде, с больными падениями поначалу: заниматься системным мышлением для вашего рабочего проекта. Без наличия рабочего проекта разобраться в системном мышлении нельзя, и у этого проекта есть две особенности:

- Это коллективный проект, а не одиночный (чтобы понять что-то про разделение труда, ибо системное мышление — оно про коллективную работу, командную, а не отшельническую)
- Это не проект личного развития, где целевая система — «я сам, любимый». Для личного развития в мастерской инженеров-менеджеров есть программа личного развития, там другие руководства.

Только в ходе достижения состояния «освоил» после «понял» тренируется **главный навык системного мышления: системное моделирование как понятийное (управляемое типами) наведение внимания на главное/важное в рабочих проектах и игнорирование неглавного/неважного. Это системное моделирование как понятийное наведение и удержание внимания (собранность) на главном и отбрасывание неглавного нужно для борьбы со сложностью реального мира, для экономии мышления. Нужно уметь найти иголку в стоге сена: выбрать своим образованным/обученным/настроенным мышлением важные объекты реального рабочего проекта.** Образованное/обученное/настроенное мышление — это знающее/понимающее и бегло умеющее сопоставлять объекты рабочего проекта с типами мета-мета-модели/понятиями из руководства по методам усиления интеллекта, в нашем случае — понятиями системного подхода, прежде всего перечисленными в разделе «Основные понятия системного подхода».

Дальше вы будете размышлять об этих объектах, а типы этих объектов будут подсказывать, о чём ещё нужно подумать, что ещё нужно учесть для выдачи как можно более точного прогноза будущей ситуации в проекте и планирования как можно более эффективных действий, которые приведут к успеху проекта.

В составленных какими-то авторами тренажёрных задачах тепличные условия, так как ничто не отвлекает от применения наставлений руководства — ни обилие незначимых деталей, ни отсутствие важной информации, которую ещё нужно найти, ни эмоциональное вовлечение в ситуацию. Например, вы понимаете, что надо определить метод, который задействует агент, работая в какой-то неизвестной вам проектной роли. Но попробуйте сообразить, каким методом пользуется начальник, который орёт на вас из вот этой неизвестной пока роли! Скорее всего, вы поначалу не будете способны это проделывать, будете отвлекаться на форму, ор начальника, ситуацию очень нервного разговора! Он орёт как «операционный менеджер»::роль для работ по «операционному менеджменту»::метод, потому как ты ломаешь ему ожидания по срокам выполнения работ, или как «инженер по требованиям»::роль в методе «инженерия требований», потому как не может

сориентироваться до сих пор, как жить в мире без требований, ибо слишком давно учился и застрял в давно несуществующей профессии, инженерия требований как метод работы ведь давно умерла? Отвечать-то этим ролям нужно по-разному, это же ор не от того, что «я начальник, ты дурак», а от каких-то проблем в проекте! В тренажёрных задачах на вас никто не будет орать, будет время подумать — это не похоже на ситуации в реальных проектах.

У тренажёрных задач заведомо есть решение, а в жизни, в реальных проектах, существование приемлемого решения — не факт. Также не факт, что вообще решаешь правильную задачу. Сначала надо правильную задачу поставить, только потом решать!

«Проектное обучение» происходит именно тут, результат выполнения заданий руководства по работе в ваших реальных рабочих проектах и даёт искомую метанойю: нейронная сеть мозга инженера-менеджера научается думать системно, системное мышление после этого уже не требует осознанных усилий при рассуждениях, в том числе не требует усилий и для привязки его понятий к объектам окружающего мира. Это переход к **неосознанной компетентности**, мы можем также назвать попадание в состояние «освоил» при обучении системному мышлению **системной метанойей.** После системной метанойи вы смотрите на мир, в котором агенты-системы в графе создателей создают и развивают какие-то целевые системы, выполняя какие-то методы/практики/культуры работы в каких-то ролях, причём это происходит рекурсивно, то есть на многих системных уровнях. И при таком взгляде на мир предельно сложный мир будет представляться чуть менее запутанным и чуть более известным. Вы уже многое знаете о мире: вы знаете типы объектов, которые вы обнаружите в этом мире (системы: целевые, создатели, подсистемы, надсистемы), так что вас будет ждать чуть меньше неприятных сюрпризов, вы будете делать чуть меньше ошибок.

Поможет ли это вам в жизни? Да, вы будете конкурировать с другими агентами, у которых не будет такого преимущества. И вы гением не станете, но всё-таки станете умнее, и это даст вам определённое конкурентное преимущество перед другими агентами.

Особенности ответов на тренажёрные вопросы (quiz)

В практике/деятельности/труде/инженерии по какому-то методу обычно делается **предположение об открытости мира**[116] (open world assumption): «что явно не описано, то просто не описано, но в мире может быть». Вася имеет три яблока. Можно ли ожидать, что у него четыре яблока? Почему бы и нет. Вы знаете, что у него есть три яблока, но про четвёртое яблоко ничего не сказано. Его у Васи может не быть, но оно может и быть. У Васи может быть и сто яблок, вам просто сообщили о наличии трёх.

Это существенно отличается от **предположения о закрытости мира** в задачах из учебников: «что не сказано, того просто нет». Алан имеет три яблока. Если у него четвёртое? Нет, конечно. Сказано же, что у него три яблока, а не два и не четыре!

Тренажёрные вопросы чаще всего составляются из предположения о закрытости мира, а опытные инженеры и менеджеры в предположении об открытости мира при

[116] https://en.wikipedia.org/wiki/Open-world_assumption

ответах на эти вопросы начинают придумывать всё более и более необычные и маловероятные обстоятельства, логично ведущие к неправильным ответам. Их задача при этом не потренировать себя, а показать ошибку авторского ответа.

При этом неявно используется сила системного мышления. От ситуации, описанной в вопросе, идём мыслью наружу, в окружение. Но поскольку окружение не прописано явно, существенно меняем его, сочиняем: «вот если речь идёт о Юпитере, и пилот ракеты не боится огромной силы тяжести и играет на саксофоне в метановой атмосфере, то ваш правильный ответ будет неправильным, а мой неправильный правильным».

Желание инженера придумывать всё более и более экзотические обстоятельства, делающие наиболее вероятный ответ не слишком вероятным, побороть нельзя. Его как раз на такое тренировали много лет! Он же должен уметь представить свою систему работающей в самом разном окружении, представить самые разные причины, которые могут привести к поломке целевой системы — и затем изменить конструкцию системы так, чтобы она продолжала работать. Единственное, что можно посоветовать такому инженеру: помнить, что **тренажёрные вопросы составлены исходя из посылки закрытого мира, а изобретение «возможных неучтённых обстоятельств» — это разворачивание посылки открытого мира, инженерная или менеджерская работа, а не работа по тренировке мозга в системном мышлении путём ответов на тренажёрные вопросы/quiz.**

Действительно, маленькая вероятность обстоятельств к чисто формальной правильности ответа отношения не имеет (даже исчезающе маловероятное событие может быть формально верным, «логичным» в математической логике) и формально «математически» ученик может быть прав. Но в учебной жизни (а не в идеальном мире математики!) генерирование таких дополнительных маловероятных условий исходя из посылки открытого мира не помогает отвечать на тренажёрные вопросы, а только мешает это делать. В рабочей жизни, наоборот, может помочь учёт самых разных маловероятных условий, надо уметь представить себе самые разнообразные ситуации, в которые может попасть будущая система, которую ты только-только проектируешь и хочешь предусмотреть её работу в самых разных внешних условиях. Нужно чётко разделять эти две ситуации: ответы на тренажёрные вопросы в ходе обучения против проектирования/сценирования в ходе конкретного инженерного, менеджерского, культурного проекта.

Особое внимание нужно уделять тренажёрным вопросам на начальных стадиях обучения — когда правильный ответ интуитивно не ясен, не является шаблонным. **Когда инженер-менеджер материал регламента/руководства/учебника/«описания процесса» знает плохо, он включает «смекалистый мозг» изменения предположений в том, что спрашивается.** Он смотрит на 2*2 и начинает: «Это может быть любое число больше 1.0 и меньше 9.0, ибо мы же не знаем, насколько и как округлили исходные числа. И это может быть в ответе вообще что угодно, начинающееся и заканчивающееся на 2, ибо звёздочка не всегда означает знак умножения. Часто звёздочка означает любое количество символов. А ещё речь может идти о символьном умножении, поэтому ответом будет 22. И давай не будем разбирать ситуации, когда система счисления недесятичная, так и быть». Конечно, он достаточно смышлён и знающ, чтобы заподозрить в ответе 4, но и недостаточно уверен в этом ответе, чтобы не предположить дополнительных подвохов и не

посочинять какие-то привходящие обстоятельства.

Обратите внимание на разницу:

- Подкручивать условия задачи, чтобы подогнать заранее выбранный тобой ответ под эти «дополненные условия», «трактовку текста задачи». В учебных кейсах это легко! Там же условие дано на трёх строчках текста, можно хоть две страницы приговорить «уточнений» и «трактовок».
- Подкручивать твой ответ, ибо он не подходит под жёстко заданную ситуацию из жизни: в жизни множество взаимосвязанных деталей, они известны, поэтому протрактовать условие тем или иным образом, чтобы «совпало с ответом» не получается.

А как надо? Надо уметь и одно, и другое!

Двое из десятка изучающих системное мышление инженеров-менеджеров текста руководства не читали (то есть читали по диагонали с экрана смартфона, когда у них было на это минут пять времени), но они хорошие инженеры-менеджеры, у них подвешен язык, они скептичны по отношению к материалу руководства (это ничего плохого, просто skeptic thinkers), но именно они обычно самые активные в группе. Эти «активисты со смекалистым мозгом» демонстрируют хорошее знание своего рабочего проекта (уровень мета-модели, знание предметной области) и плохое знание системного мышления (уровень абстракции мета-мета-модели). Присвоения типов системного мышления объектам рабочего проекта не происходит, а бойкий рассказ про проект на языке прикладной системной области без демонстрации типовых ходов именно системного мышления — он только доказывает, что овладения системным мышлением не произошло, понимания понятий системного мышления нет.

Результат — два балла по системному мышлению (что в мастерской инженеров-менеджеров не бывает, там оценки не ставят. А вот в МФТИ вполне может быть). Прикладное знание, то есть знание мета-модели (а также модели и даже экземпляров «для примеров») при этом может быть отличным! Но системности, то есть использования понятий системного мышления в рассуждениях с целью управления вниманием в проекте — нет. Отсюда вполне возможна эта парадоксальная ситуация: пять баллов за прикладное мышление, и два балла за системное. Хорошо делать своё дело, плохо вписываться в общий проект.

Цель «опытных инженеров» часто не столько поупражняться в системном мышлении и использовании его концептов, сколько попробовать «прогнуть» предлагаемые вопросы вместе с самими идеями системного мышления, испытать их на прочность формально-логичного «здравого смысла». Этим людям хорошо работать Беспристрастными Свидетелями (Fair Witness) из Хайнлайна: «*Энн стояла на трамплине. Джабл крикнул ей: - Тебе виден тот дом на горе? Какого он цвета? Энн посмотрела и сказала: - С нашей стороны белый. Джабл обернулся к Джилл: — Вот видишь, Энн не стала говорить, что дом белый целиком. И вся королевская рать не заставит её сказать это до тех пор, пока она не пойдёт и не посмотрит. Но даже и тогда она не сможет утверждать, что дом остался белым после того, как она ушла»*[117].

И помним, что математически и физически правильные ответы могут быть системно неправильными: «дом состоит из молекул дома» — это абсолютно правильно в физике и математике, но в системном мышлении вам сразу ткнут в рассуждение с игнорированием системных уровней, укажут на ошибку. «Дом из кирпичей» тоже будет плохо. «Дом из стен, а стены из кирпичей» — вот это подойдёт. «Смекалистый мозг» тут же начнёт указывать, что пропущена крыша, крылечко, интерьеры и жильцы дома — дом состоит не только из стен, которые состоят из кирпичей. «Смекалистый мозг» не хочет разбираться с тем, что именно тренирует тренажёрный вопрос. А зря, надо просто учитывать: в вопросах — закрытый «игрушечный» мир, тренировка использования какой-то мыслительной операции из руководства по системному мышлению, а не тренировка смекалки. Вопросы на понимание наставлений руководства, знание терминологии и основных идей, а не на проявление смекалки. Не олимпиадные задачи, а тренажёрные вопросы.

[117] https://ru.wikipedia.org/wiki/Чужак_в_чужой_стране

Как мы могли бы с этим бороться? **Очевидный неправильный ход — строго формализовать вопросы, добиваясь однозначности правильного ответа.** Чем формальней будут поставлены вопросы «из учебника», тем дальше они будут от реальной жизни! Жизнь не формально-логична, жизнь вероятностна! Интересуют общие закономерности, наиболее часто встречающиеся случаи, а не маловероятные исключения и особые ситуации! Если рассмотреть вопрос о том, чем забивать гвоздь — молотком, камнем, подушкой, рукой, микроскопом, то ответ тут должен быть «молотком» (а не «молотком, камнем, микроскопом, и ещё рукой, потому что рука может быть в стальной перчатке от кольчуги»). Но если задача «чем забивать гвоздь, если вы оказались в дороге, и молотка нет?», то она будет иметь другие ответы! Вопросы ситуационны, они не абсолютны, их решения вероятностны, в учебных задачах требуется ответ с максимальной вероятностью, а не абсолютно верный единственный.

Наиболее надёжный способ избегать всего этого изменения предположений о ситуации («подкрутки условий задачи») вместо ответа на вопрос в целях тренировки какого-то отдельного наставления руководства — это использовать для решения задач ситуацию в собственном проекте. Реальную жизнь так просто не подкрутишь в части обстоятельств, поэтому придётся-таки решать задачу, а не подгонять ситуацию под предлагаемое какое-то решение, как происходит с учебными задачами. Так что в нашем руководстве значительное число заданий — это не ответы на вопросы с «ловушками» для «народного мышления», а предложение помоделировать свои рабочие проекты, задание поработать, но не абы как, а в соответствии с предлагаемыми методами мышления, методами моделирования. Рабочие проекты — это которые на работе, а не «учебные проекты» или «кейсы учебного курса». Всё из жизни, причём из вашей рабочей жизни (для личной жизни есть руководства программы личного развития), а не предложенное руководством.

Все учебные **вопросы** и даже рабочие **задания — учебные/тренажёрные и рабочие, а не экзаменационные**. Они дают лишь повод осознать и обсудить наставления руководства, наставить на правильный способ мышления, формализм «единственно правильного ответа» для них непринципиален и даже неверен.

Ещё один источник возможной «формальной нелогичности» системного мышления в том, что системный подход был изложен в стандартах и публичных документах, откуда мы берём его положения. В каждом таком стандарте и публичном документе набор понятий системного подхода представлен не весь, а небольшим кусочком. Эти кусочки, взятые вместе, не представляют из себя логически непротиворечивую систему концептов и их отношений. В разных текстах стандартов и публичных документов ещё была использована и разная терминология. Это удалось частично преодолеть при переводе на русский язык, но не полностью. Каждый стандарт имеет свой небольшой набор понятий, не претендующий на полноту и непротиворечивость с другими наборами понятий, только «для целей настоящего документа», и не более. Часто стандарт в разделе терминологии так и начинается, «для целей настоящего документа применяются следующие термины и определения/For the purposes of this document, the following terms and definitions apply» — предупреждение, что авторы стандарта не претендуют на всеохватность термина.

Вероятностная природа понятий и изложение на естественном (чаще всего — английском, а не русском) не слишком формальном языке только добавляет

неопределённости. Перевод на русский язык тоже не добавляет формальности в изложении. В реальных проектах «из жизни» тоже очень трудно составить непротиворечивое системное описание ситуации (в реальных проектах разные люди используют разные версии самых разных методов, в том числе и разные версии методов системного мышления), так что использовать системное мышление будет много тяжелей, чем в специально составленных учебных вопросах. Поэтому задачи намеренно составлены не слишком формально, они тренажёрны, в них не ожидается «подвоха» на особенность и уникальность ситуации, но ожидается хорошее знание наставлений руководства.

Если вы не очень поняли, что требуется в качестве ответа на вопрос или какой должен быть результат моделирования в задании по заполнению табличек, задайте себе вопросы: «Что хочет проверить автор руководства из того, что я смог понять из его наставления? На понимание чего меня хочет натренировать автор руководства»?

Переход к использованию мышления

Чтения наставлений руководства абсолютно недостаточно — это как читать учебник по езде на велосипеде. Без практикования самой езды на физическом велосипеде это чтение не сильно помогает. Уметь отлично отвечать на тренажёрные вопросы/quiz тоже недостаточно — это как уметь ездить на велосипеде только по прямой, на специально оборудованной дорожке, да ещё к велосипеду в учебных целях прикручено третье колесо, для помощи в удержании равновесия. Нет, нужно сразу работать, а в ходе работы тренироваться в постановке задач в ваших рабочих проектах и системному мышлению в реальной жизни (ездить на велосипеде по бездорожью в горах) — и только тогда цветущая сложность рабочих проектов начинает отступать и поддаваться тренированному в системном мышлении мозгу. Все аналогии с программированием компьютера — ложь, у вас нейронная сетка, её не программируют, а её обучают, ей нужны демонстрация правильных и неправильных паттернов, и этих демонстраций должно быть некоторое количество, и эти паттерны надо уметь находить в жизни. Как сказал один из наших инженеров-менеджеров, который долго изучал анатомический атлас, а затем пошёл в анатомичку и препарировал там пару тел: «оно в жизни совсем не так выглядит, как в атласе». Действительно: возьмите карту и местность. Они выглядят совсем не похожими друг на друга! Читаете о мета-мета-модели, в жизни видите экземпляры объектов, они совсем по-другому выглядят!

В ходе системного мышления вам прежде всего нужно заниматься мета-моделированием, то есть «составлять таблички», и моделированием, то есть «заполнять таблички». Красивые диаграммы из квадратиков и стрелочек — это попсовое/бытовое/народное понимание системного моделирования. Таблицы и даже тексты тут более адекватны, их много проще менять в ходе эволюционной инженерии. Это «создавать и заполнять таблички» и есть системное мышление, всегда не просто «в голове», а «мышление письмом», «мышление моделированием». Задания на моделирование (заполнение табличек) в нашем руководстве — это и есть первые шаги в системном мышлении.

Чтение наставлений руководства — это вы не делаете «первые шаги», это вы просто читаете про «первые шаги». Даже ответы на вопросы — это ответы на вопросы о первых шагах, а не прошагивание этих первых шагов! А вот когда начали

составлять и заполнять таблички примерно такие же и так же, как в наших заданиях на моделирование на материале ваших проектов — всё, это и есть «системно думать», это вы не читаете о первых шагах, это вы их уже делаете, уже шагаете по бесконечной дороге системного мышления!

В ходе освоения наставлений нашего руководства, освоения работы по методам мышления из нашего руководства вы уже получите пользу для ваших рабочих проектов от занятий моделированием, поэтому смело занимайтесь этим моделированием в рабочее время, а не в отдельное время «учёбы»! У нас стажировка, «обучение на рабочем месте», а не «учёба»! Конечно, вы можете читать наставления руководства и даже тренироваться в ответах на тренажёрные вопросы в отдельное «сверхурочное» рабочее время, которое вы хотите инвестировать в своё рабочее развитие, но вот выполнением заданий на моделирование по рабочим проектам надо заниматься прямо в рабочее время, это и есть приложение системного мышления к рабочим задачам. Если вы откладываете «на потом» задания по моделированию из нашего руководства, то это прокрастинация в квадрате: как прокрастинация вашего рабочего развития, так и прокрастинация сильного мышления в рабочем проекте, прокрастинация нанесения пользы вашему рабочему проекту!

Основных идей системного подхода немного, каждая из этих идей довольно быстро понимается. Если выразить системный подход в одном абзаце, то получится что-то типа *«для успеха системы, то есть удовлетворения интересов внешних проектных ролей нужно понять функции работающей системы в её окружении (сделать концепцию использования), затем разобраться в том, как система функционирует внутри себя и предложить конструкцию системы, увязав с пространственной компоновкой и оценками совокупной стоимости владения (сделать концепцию системы). Принять архитектурные решения (принципы разделения на модули и организации связей между модулями, совокупность таких решений и есть «архитектура системы»), оценив множество конфликтов между системными уровнями и выбрав наименее плохие решения, ибо хороших решений не будет (никакой устаканенности, сплошные неустроенности!). Добиться детализации описания системы с точностью, достаточной для изготовления на выбранной производственной платформе — и изготовить. Получить основанное на каких-то свидетельствах обоснование того, что система успешна. Эксплуатировать систему, снова и снова повторяя всё сделанное для улучшения каких-то частей системы, а то и для полного изменения системы. И для всего этого надо иметь систему-создателя (чаще всего — организацию), которая всё это делает, выполняя самые разные работы по лучшим известным методам. Мышление про создателя устроено примерно так же — функциональные части там проектные роли, конструктивные части — оргзвенья. Это мышление и связанные с ним проектные действия по изменению физического мира проходят рекурсивно на многих уровнях рассматривания систем как частей надсистем и рассмотрения их как состоящих из частей-подсистем, а также выстраивания цепочек создания из систем-создателей».*

Мы ещё несколько раз сформулируем эту цепочку рассуждений о системе, чуть более структурировано, в нашем руководстве она также называется «системная мантра».

Проблема даже не в том, что эти предложения в предыдущем абзаце сплошь состоят из терминов (вроде «концепция использования», «архитектурные решения», «производственная платформа»), значения которых нужно знать. Все

эти положения глубоко связаны друг с другом и крайне редко используются поодиночке. Так что требуется некоторая **беглость** (fluency) в их одновременном и совместном применении — примерно в том же смысле, что и беглость пальцев в игре на рояле или наборе текста на клавиатуре, беглость в говорении на иностранном языке. Каждая клавиша на рояле или клавиатуре понятно как нажимается, любой котёнок может нажать клавишу! Клавиш не так уж много. Проблема только в том, чтобы разные клавиши нажимать вовремя, быстро и такие, какие нужно для получения музыки, а иногда нажимать несколько клавиш сразу (аккорд). На освоение компьютерной клавиатуры уходит несколько дней тренировки, но на освоение рояльных клавиш уходит несколько лет. В освоении системного (и любого другого) мышления, как и в освоении игры на рояле, нет царских путей, кроме как множества повторений каких-то операций мышления в ходе моделирования в разных ситуациях, получение опыта мышления с использованием системных понятий в жизни. Тренируются операции мышления прежде всего при составлении и заполнении табличек ровно вот так, как это происходит при выполнении заданий из руководства.

Каждая табличка задействует минимум три онтологических уровня моделирования:

1. тип типа, указанного в имени колонки таблички, в табличке не представлен никак, удерживается в голове составителем таблички
2. Имя типа в колонке таблички,
3. Объект в ячейке таблички типа, указанного в колонке таблички.

Обычно в рабочих проектах тип типа в колонке таблички — это тип из мета-мета-модели (набора понятий трансдисциплин интеллект-стека методов фундаментального мышления, даётся в наших руководствах по усилению интеллекта в рамках рабочего развития, включая наше руководство по системному мышлению), тип в колонке таблички — тип из мета-модели (модели прикладной предметной области рабочего проекта), а в ячейках — объекты предметной области. В нашем руководстве много табличек с похожей раскладкой типов по уровням абстракции (уровням мета-моделирования), но поскольку мы учим ещё и составлять таблички, в задании могут быть и другие варианты раскладки по уровням абстракции.

Увы, тренировка мышления с использованием типов мета-мета-модели интеллект-стека фундаментальных методов мышления занимает время. Поэтому интеллект желательно усиливать: учиться не только читать художественную литературу, обращающуюся к быстрому мышлению S1, но и литературу со строгой/формальной типизацией для развития медленного строгого мышления S2, а ещё надо привыкать самому писать тексты со строгим использованием типов, а также не только «продумывать в голове», но и документировать строгие модели.

Повторим ещё раз, это важно: не пишешь — не мыслишь, в том числе не мыслишь системно! Составление и заполнение табличек для рабочего проекта — это системное мышление, которое заключается в работе с трёхуровневой онтологией (трёхуровневой иерархией типов, трёхуровневой классификацией), и именно это и есть форма, в которой проходит системное мышление. Конечно, тут может быть множество вариантов, но наиболее типовой вариант выглядит так:

В этом примере:

- типы мета-мета-модели (понятия из наших руководств по рабочему развитию) как типы для колонки таблицы, удерживаются «в голове» и не представлены в таблице, вы их не обсуждаете с сотрудниками в рабочем проекте, если они не знакомы с системным мышлением. Но вы их используете сами для мышления! **Системное мышление как раз в этом: оно будет невидимо в таблице, но будет присутствовать, оно нужно для присвоения типов объектам «колонка таблицы»!**
- типы мета-модели (учебник по методам работы в какой-то прикладной предметной области проекта как метаУ-модель/«модель из Учебника для всех», типы каких-нибудь корпоративных регламентов/инструкций или даже стандартов как метаС-модель/«Ситуационная модель — как у нас», на картинке показали только тип метаС-модели) как тип ячеек, указываемый в имени колонки таблицы — и тем самым вы делаете присвоение типа для всех объектов в ячейках этой колонки.
- значения модели (никаких «мета») для каких-то объектов в ячейке из строк этой таблицы. Ещё нужно учесть отношение репрезентации между описанием в ячейке и каким-то описываемым объектом проекта, его в табличке нет, как нет типа мета-мета-модели. Но описываемый объект есть, в ячейке только его описание!

Это всё требует беглой работы не с понятиями системного мышления, а с типами — беглого владения «теоретической теорией» (theory theory в теории понятий)[118], а также иерархиями типов (онтология). Наставления по такой работе типами и онтологиями в программе рабочего развития даются в руководстве по

[118] https://iep.utm.edu/th-th-co/

рациональной работе, освоение такой работы — обязательный пререквизит к освоению системного мышления по нашему руководству.

Беглость в мышлении важна. Если вам трудно заполнять таблички, а каждая табличка ожидается как приключение на три дня, то вы просто не будете системно мыслить. Если заполнять таблички — это для вас дело пяти минут на каждую табличку, то системное мышление у вас будет. Разница? В первом случае — нет беглости, во втором — беглость есть, ведь разница в скорости между «делаю первый раз» и «делаю сотый раз» вполне может быть на порядок, а то и больше. Это как перевод с иностранного языка: если вы за каждым словом лезете в словарь, на перевод одной страницы может уйти несколько дней, а если вы уже знаете язык, то вы просто внимательно читаете исходный текст и пишете текст на иностранном языке за несколько минут. В какой-то мере системное мышление — это тоже иностранный язык, вам нужно уметь переводить «с системного на предметный, и назад» (с разговора в типах мета-мета-модели/трансдисциплин на разговор в типах мета-модели/«прикладных дисциплин методов работы в предметной области проекта», и назад).

Вот из материалов Viewpoint Research Institute[119], тут обращаются к понятию hard fun (мы уже упоминали это понятие, предложенное Seymour Papert), системное мышление тоже hard fun:

> Мы хотим помочь детям развить реальную беглость (fluency) во многих областях образования, включая мышление, математику и науки. Каждый из этих предметов не поддаётся «естественному обучению» (как учатся ходить и говорить). Довольно много времени и энергии нужно потратить, чтобы получить беглость выше пороговой. Тут интересное сходство с искусством, музыкой и спортом, для каждого из них также требуется довольно много времени и энергии, чтобы получить беглость. Эти искусства могли бы называться «трудное развлечение» (hard fun). Математики и ученые знают, что они занимаются искусством, равно как трудным развлечением. «Мышление» — это более высокая категория, чем «просто» математика, наука и искусства. Оно представляет синтез интуитивного и аналитического подходов к пониманию мира и поведения в нем.

Peter Senge в книге «Пятая дисциплина»[120] (1990) писал:

> Недавно в ходе пятидневного вводного курса, проводимого Обучающим центром МТИ, одна женщина-менеджер из конструкторского отдела компании Ford лаконично сформулировала ситуацию: «*Спустя пару дней, — сказала она, — я начинаю понимать, о чем вся эта история с системным мышлением и интеллектуальными моделями. Мне это напоминает время, когда я только начала знакомиться с высшей математикой. Сначала я чувствовала себя совершенно потерянной. Все это было мне совершенно чуждо. Но потом я начала «схватывать» суть. Через год я уже вполне владела основами этого дела. Через пять лет это стало основой моей профессии».* Потом она добавила: «*Если бы высшую математику изобрели сегодня, ни одна из наших корпораций не смогла бы ею овладеть. Мы бы посылали каждого на трехдневные курсы. Затем каждый получал бы три месяца на то, чтобы посмотреть, работают ли «все эти штуки». А когда выяснялось бы, что они не*

[119] http://vpri.org/

[120] https://en.wikipedia.org/wiki/The_Fifth_Discipline

работают, мы бы начинали пробовать что-нибудь другое».

Если заниматься языками, то любой из них можно довести до уровня C1 (достаточный для поступления в европейский вуз) за год, если интенсивно заниматься — для языка без флексий (английский, испанский) нужно на это потратить 600 часов, с флексиями (русский, немецкий) 1100 часов, для языков совсем другой структуры (китайский, японский) 2200 часов. Если заниматься год, то в день нужно тратить примерно 1.6, 3 и 6 часов соответственно, и в интернете можно найти достаточно примеров, как мотивированные люди выделяли примерно такое время в своём расписании и достигали успеха. Чтобы достичь в языке мастерства, нужно потратить порядка 10000 часов (это весьма спорное утверждение, но порядок величины верный) — то есть заниматься языком несколько лет. И в случае иностранного языка это даже не «мыслить» и не узнать о каких-то новых вещах и их связях, это просто «переназвать известные уже вещи другими словами»! Системное мышление относится к того же сорта методам: его нужно практиковать, чтобы добиться беглости использования в жизни, а не «мыслить со словарём» в тех случаях, когда «решили, что в этом случае нужно применить системное мышление». И помним, что кроме системного мышления нужно добиться беглости ещё и в других приёмах качественного фундаментального мышления. Например, представлять содержание методов системной инженерии, да и просто быть собранным, чтобы удерживать внимание в длительных раздумьях и мочь одолеть в три (а не тридцать три) приёма по нескольку часов каждый книжку на 200 страниц, а не отвлекаться на социальные сети каждые пятнадцать минут и мочь одолеть в три приёма только сериал на пять сезонов.

Ситуацию, когда системное мышление имеется на уровне осознанной компетентности («понял») но, увы, не беглости во владении («освоил»), при этом в рабочих проектах не применяется, несмотря на его понимание, мы вообще рассматривать не будем. **Вопрос «а в каких проектах использовать системное мышление?» примерно такой же, как «а в каких проектах нужно думать?». Управление вниманием при помощи набора понятий системного мышления используется во всех проектах, нет ведь проектов, которые лучше выполнять с безалаберно выбираемыми из окружающего мира объектами, бездумно забывая самые разные важные для выполнения проектов методы работы — то забыть проверить воплощение системы на работоспособность (выполнить работы метода инженерных обоснований — тестирование, испытания, трактовку результатов испытаний), то забыть необходимость набрать и организовать команду для выполнения проекта (удивительно, но это бывает довольно часто: проект вроде открыт, а командой в нём никто не озабочен, ресурсов на это не выделяется, ожидается выполнение его уже на 100% загруженными другой работой агентами).**

Хотя да, проект чистки зубов утром можно выполнять без системного мышления. Но такие проекты доступны даже детям в детских садах, эти проекты вообще не требуют сильного интеллекта. А вот где требуется интеллект, там осмысленно задействовать системное мышление. Нет в жизни деления на проекты, где мозг включать осмысленно, и где его включать не осмысленно. А уж когда в проекте участвуют несколько человек, то системное мышление поможет даже в минимальных дозах: оно хотя и не скажет, куда именно надо идти, но даст карту местности и тем самым не даст заблудиться или уйти в центр болота.

Задания: невидимость системного мышления для окружающих.

Поставьте отметки о достижении состояния:

1. Написан пост о невидимости системного мышления для окружающих. В руководстве написано: «типы мета-мета-модели (понятия из руководств нашей программы рабочего развития) как типы для колонки таблицы, удерживаются «в голове» и не представлены в таблице, вы их не обсуждаете с сотрудниками в рабочем проекте, если они не знакомы с системным мышлением». В посте отражён свой опыт как ответы на вопросы: Удаётся ли вам не использовать типы из наших руководств в обсуждениях? А используете ли вы их, когда размышляете сами?

Искушение сложностью

Частая ошибка у начинающих системных мыслителей — это выбор для первых проб системной мысли предельно сложных случаев, которые будут трудными и для опытных системных мыслителей. Прежде всего это ситуации, связанные с длинными последовательностями обработки данных в каких-то программах. В IT-проектах часто речь идёт не о физическом мире, а о манипулировании длинными цепочками описаний (цепочка — это когда одно описание описывает не физический мир, а другое описание, то другое описание тоже описывает ещё какое-то описание, и так далее), которые хранятся в базах данных различных приложений, как-то связаны с реализованными в софте алгоритмами и неочевидно связаны с физическим миром.

В наших руководствах чётко говорится о том, что системы физичны. В некоторых школах системной мысли допускаются нефизические системы, но не у нас — обычно это ведёт не к строгим рассуждениям, а неформальным фантазиям, отрыву от физической реальности, отсутствию заземления/grounding. В нашем руководстве будут даны ссылки на исследования по будущим поколениям системного мышления, в которых допускается мысль о том, как думать про части и целые для описаний. При этом привязка к реальности осуществляется через понимание, что об этих частях и целых думает какой-то физический агент. Но это явно сложный случай, это фронтир системного мышления, который пока не попадает в учебники.

Софт и базы данных этого софта — это описания, и не всегда это описания физического мира, часто это ещё и описания других описаний. В случае софта «дотянуться мыслью до физического мира» часто означает ещё и разобраться в инженерии предприятия, которое будет использовать софт, то есть подразумевает какое-то знание менеджмента. Терпения и квалификации распутывать эти длинные цепочки описаний и прихватывать знания прикладной инженерии самых разных систем (включая сложнейшие системы-предприятия, системы-личности) у начинающих мыслителей не хватает, предупреждение об осторожности при работе с айтишными проектами на них не действует. Поэтому они сразу говорят «моя целевая система — это программа, реализующая алгоритм (описанная алгоритмом)», хотя в нашем руководстве есть обсуждения многочисленных ошибок, которые часто появляются, когда делают такие заявления — но эти ошибки становятся понятными чуть позже, сначала же не воспринимаются инженерами-менеджерами, как ошибки.

То же самое относится к мастерству как целевой системе, ибо это такая же «программа, реализующая алгоритм какого-то метода», только не для классического компьютера, а для агента на базе нейросетевого вычислителя

(человека или робота) и инструментов. Иногда первой же целевой системой выбирается какое-то сообщество или даже общество в целом: система, состоящая из множества индивидуальных агентов с сильным интеллектом, причём с неясными границами. Примерно так же сложен случай целевой системы «интернет» или какой-нибудь другой компьютерной сети.

Это как первоклассник музыкальной школы берёт сыграть на рояле в качестве первой пробы услышанный по радио свежайший оркестровый хит — да, прямо все инструменты в этом оркестре, вот давайте попробуем на моём рояле, ответьте мне, как мне это сыграть сегодня к вечеру, особенно вот эту чудесную партию барабана! Преподаватель может лишь печально сказать, что сейчас — никак не сыграть, и особенно партию барабана. Разбирательство со сложными корпоративными айтишными проектами и агентами-людьми — худший вариант для начала работы с системным мышлением. Увы, это замечание мало кого останавливает. Но оно есть, вернитесь и перечитайте. Наверняка где-то у вас есть в большом рабочем проекте небольшой подпроект, где участвуют отдельные системы, для которых вам легко определить их границы в пространстве-времени (мебель в кабинете, физический компьютер с блоком питания, здание), то есть по поводу которых легко договориться, какие именно физические объекты являются этими системами. Вот их и берите для начала.

По определению сложны орг-проекты, ибо в них задействованы люди. Людей очень, очень трудно моделировать, ибо личности в разы сложнее уровня классических киберфизических систем. У нас есть руководство по инженерии личности, где об этом говорится подробней, но работу по его наставлениям надо осваивать уже после освоения работы по наставлениям руководств по рациональной работе, системному мышлению, методологии, системной инженерии.

Мастерство — это часть личности, которая в свою очередь — часть агента. Кроме мастерства как реализованного «аппаратурой» вычислителя, в агенте есть ещё и его организм, собственно аппаратура. Получается, что агент с сильным интеллектом (например, человек) — это программно-аппаратный комплекс, в котором личность — это «софт», а «мастерство» — это одно из приложений/application program в этом софте. Это не так сложно, через пару-тройку месяцев изучения системного мышления это должно быть понятно. Но вот сразу брать систему «мастерство» и «личность» в качестве целевой системы — это сразу получить много-много ошибок. Как на горном велосипеде вместо попыток проехать первые в жизни сто метров по ровной дороге сразу идти на самую сложную трассу самой крутой и дикой горы. Это, конечно, можно делать, но тогда надо быть готовым к синякам и переломам.

Организация — это следующий за личностями системный уровень, то есть ещё более сложный (нужно понимать неустроенности/frustrations, возникающие между людьми и организацией, организацией и окружающими её сообществами, а также обществом в целом). У нас есть руководство по системному менеджменту, оно наставляет в инженерии организаций. Там основная идея в том, что функциональные системы — это выполняющие какие-то методы/практики работы организационные роли, а вот оргзвенья — это подразделения, комиссии, рабочие группы проекта. И минимальным оргзвеном там будет человек или AI-агент. Аналогом «мастерства» там будет «оргвозможность» организации осуществить работы в какой-то роли по какому-то методу. Эта «оргвозможность» как раз будет состоять из скоординированного и поддержанного ресурсами и полномочиями

мастерства агентов, составляющих оргзвено. То есть все сложности с «мастерством» остаются, только они умножаются оттого, что в организации координируется работа самых разных агентов в тот момент, когда они задействуют своё мастерство в работе. «Задействуют в работе» — мы же помним, что «метод/культура/стиль/стратегия работы» отличаются от самой «работы»? Подробности на эту тему отличия, конечно, будут, но они главным образом будут раскрыты в руководстве по методологии, а методологическое мышление вы будете осваивать после системного мышления. Так что **выбирать организацию целевой системой мы не рекомендуем**. Особенно **мы рекомендуем не выбирать организацию целевой системой личностям, считающим себя менеджерами, это у них распространённая ошибка**.

У менеджера, который считает своей целевой системой организацию, эта организация работает отлично, только там беда: выпускаемые ей инженерные системы не работают, как надо. Люди в организации накормлены, станки смазаны, компьютеры жужжат — с организацией всё отлично, но вот с продуктами и сервисами беда, ибо во внимание менеджера она не попадает, так что организация у нас не целевая. Целевая система организации, продукт предприятия (в том числе продукт, который организация меняет своим сервисом) — вот это должно быть целевой системой даже для менеджера, а сама организация — это просто «его система».

Нужно иметь определённые знания и опыт для мышления о системах разной природы с участием классического компьютерного софта и не очень классического «знаниевого софта» для агентов с сильным интеллектом (людей, организаций, AI-агентов, коллективных систем из людей, организаций, AI-агентов). Важно, что «знания и опыт», а не просто «жизненный опыт». Нужно уже иметь достаточно сильный интеллект, ибо поведение людей и AI-агентов удивляет своим разнообразием и даёт неожиданности в проектах. Интеллект нужен, чтобы разбираться в этих неожиданностях, решать многочисленные проблемы. Если у вас в проекте молоток, то он будет вести себя примерно так, как описано в учебниках про молотки. Если у вас Вася или Даша, то они будут вести себя совершенно необязательно так, как написано в учебниках про людей. Более того, они намеренно будут вести себя не так, если будут знать, что вы хотите ими проманипулировать на основании знания из учебника. Вы же и сам такой, почему бы им тоже не быть такими? Если у вас организация из пары тысяч Вась и Даш, то она тоже может вести себя не так, как вы ожидаете — и сильный интеллект (включая владение системным мышлением) вам поможет разобраться.

Но кроме интеллекта вам нужна какая-то знаниевая опора в мышлении об этих сложных объектах, мастерство в методах работы с ними. Наши руководства по системной инженерии, инженерии личности, системного менеджмента (как инженерии организации) вам в этом помогут. Осваивать работу по этим руководствам вы будете только после того, как закончите осваивать «мыслительный минимум» по линии усиления интеллекта (то есть осваивать мышление по наставлениям руководств по рациональной работе, системному мышлению, методологии).

И всё-таки большинство первых систем, о которых хотят думать новички в системном мышлении, оказываются системами с софтом и людьми, ибо мы работаем в мире, в котором окружены софтом и людьми, а с недавнего времени и «нежитью» (софтом AI, который больше похож по поведению на людей, чем на классический

компьютерный софт). Мы, конечно, пытаемся наставить в наших руководствах по рабочему развитию как думать о софте, людях, AI-агентах и созданных из них организациях, но сразу предупреждаем: иметь такие системы как целевые очень сложно в качестве первых примеров системного мышления. Музыканты не начинают осваивать мастерство игры, пробуя сыграть опусы Шопена. Первокласснику не нужно брать решать дифференциальные уравнения, ему сначала нужно тренироваться на арифметике! Но если у вас именно такой рабочий проект, то предупреждаем: будет очень трудно, но будет и несомненная польза, если потратите достаточно времени, чтобы «прорваться».

Если речь идёт о городе (сообщество) или стране (общество), то ситуация ещё более сложна в системном моделировании, чем ситуация с предприятием. В предприятии вы ещё сможете на что-то влиять, ибо там предусмотрены способы выдачи поручений — люди организованы. Если люди не организованы (непонятно, кто начальник и кто подчинённый в каких-то вопросах, нельзя поручить работу), то возможности целенаправленного действия одного человека или даже команды людей малы (если вы, конечно, не командир небольшого вооружённого отряда, финансируемого отнятыми у налогоплательщиков деньгами). Ваше мышление о сообществах и обществах будет непродуктивно — мышление начинающих системных мыслителей будет изобиловать ошибками ввиду чрезвычайной сложности таких ситуаций.

У разных агентов, которых вы хотите инженерно организовать в вашем проекте создания сообщества или общества, будут обнаружены разные предпочтения, они все сами и без вас пытаются что-то сделать для реализации своих «хотелок» разного уровня продуманности, они не ждут ваших действий и заведомо (часто справедливо) считают их непродуманными и не отвечающими SoTA (лучшим известным на сегодня, State-of-The-Art) методам работы, не считают их стильными/культурными, не ждут, пока вы о них всех системно подумаете — они не мечтают, чтобы вы планировали их жизнь и наверняка будут яростно сопротивляться вашим предложениям. Точно так же будете действовать и вы, когда кто-то из осваивающих работу по наставлениям из наших руководств решит пропланировать вашу жизнь, а потом будет настаивать на выполнении этих планов. Вот вы и не планируйте жизнь каких-то малоизвестных вам лично людей, ни в собственной стране, ни в соседней! Разберитесь сначала с «железными» простыми системами, потом с собой как личностью (вы же тоже не любите выполнять составленные вами для себя планы, выполнять выданные себе рекомендации!), потом научитесь организовывать небольшие команды, потом большие организации, а там и силами таких организаций у вас будут возможности что-то существенно поменять в их окружении, то есть в сообществах и обществах.

Вернитесь и перечитайте эти предупреждения: там, где много информационных технологий (включая AI-агентов) и людей (а часто в проекте есть и софт, и люди, и AI — причём что-то из этого даже не замечают, «рыбка воды не замечает», даже вы сами — это люди, и ваша работа поддержана софтом), будьте осторожны с первыми примерами системного мышления. Не берите такие примеры. Не начинайте со своего самого запутанного рабочего проекта, вы только разочаруетесь. Если вы вчера начали изучать суахили, не ввязывайтесь в ситуацию синхронного перевода на бизнес-встрече военных с коммерсантами, кроме неприятностей из этого ничего не получится. Подождите, пока вы не

достигнете какой-то беглости. Тренируйтесь на более простых проектах, где поменьше информационных технологий и поменьше людей.

Конечно, при этом появляются огромные трудности в организации обучения: у человека максимальные сложности в его текущих проектах, и учит он средство уменьшения этой сложности, но как раз в этот эпицентр сложности с его текущим умением ему попадать нельзя, нужно до этого хотя бы немного потренироваться на более простых примерах!

Системное мышление — это не единственное руководство по интеллект-стеку фундаментальных методов мышления из нашей программы, который нужно бы пройти современному инженеру-менеджеру. Всего освоение руководств программы рабочего развития занимает примерно год при выделении двух часов в неделю на пятидневке. Для этого нужно изменить образ жизни и стиль работы, ибо откуда-то эти несколько часов нужно взять, как-то переустроить своё типичное дневное расписание (скорее всего, как личное, так и рабочее).

Это как проходить стажировку, когда уже имеешь некоторый опыт работы: важно, чтобы получить другие рабочие, а тем самым и жизненные возможности. Освоение новых методов мышления вообще-то неблагодарное занятие: если вы занимались этим целый рабочий день с утра до вечера, то вас похвалить будет некому — это не физическая работа с материальными хорошо наблюдаемыми результатами (вроде колки дров или рытья канавы), которую можно существенно продвинуть за пару дней так, что всем будут заметны результаты. Нет, придётся потратить много дней без немедленных наград. Зато это позволит потом претендовать на другие работы и принципиально другой уровень наград.

Для освоения руководства по системному мышлению есть обязательный пререквизит: освоение рациональной работы по соответствующему руководству. Рациональная работа в том числе учит тому, как найти время для дополнительных усилий по освоению мыслительной работы не «по наитию», а по нашим руководствам программы рабочего развития. Этому также учит программа личного развития.

В ходе освоения наших программ развития вы получаете следующие квалификации:

- **Программа личного развития**: освоение мастерства менять себя, любимого, к лучшему.
    - ученик — и там много градусов/степеней/degrees (всё это синонимы, от латинского gradus «шаг, ступень, степень, градус», далее от gradī «шагать, ступать, продвигаться» (восходит к праиндоевр. *ghredh- «идти»). Поэтому раньше степени переводили как градусы. Ученик на выходе должен продемонстрировать, что он умеет полноценно выполнять роль ученика: осваивать какую-то предметную область, тратя на это (если это не «интенсив», а «просто таким образом живу») от двух часов в день. Например, это мастерство начать осваивать какое-то руководство, и не «соскочить» на середине. Тут мы тоже склонны говорить не об обучении быть учеником, а о стажировке у нас в мастерской инженеров-менеджеров: это больше передача культуры постоянного обучения чему-то новому, постоянного личного развития, постоянного инвестирования времени в себя-будущего.
- **Программа рабочего развития**: освоение мастерства менять окружающий мир к лучшему, работая в какой-то организации инженером, менеджером или

кем-то ещё, выполняя работу в коллективном проекте.

- работник — в какой-то организации может работать рационально, удерживая не только свои личные цели, но и цели организации в целом (подстраиваться под ограничение потока работ). Впрочем, работник ещё может и многое другое (скажем, быть собранным, удерживая внимание на важных объектах, а также явно указывать на важные объекты, сообразуясь с их типами — мета-моделировать), а поскольку он стал производительней и точнее в разговоре, то результаты его работы немедленно замечаются окружающими. Это в значительной мере потому, что работник освоил руководство по рациональной работе.
- стратег — рационально выбирает метод и предмет работ в ситуациях неопределённости, обладает системным мышлением (всегда сначала смотрит на систему в момент её эксплуатации как на чёрный ящик в окружении, а потом уже разбирается с тем, как она устроена и что там внутри системы — и не забывает посмотреть на то, что делают создатели системы). Это в значительной мере от того, что освоил руководства по системному мышлению и методологии. Руководство по методологии как раз заканчивается разделом по теории стратегирования, вот отсюда и название степени квалификации. Наши нижние степени рабочей квалификации (работник, стратег) намеренно звучат как продвинутые: единомышленникам разобраться нетрудно, но инженерам-менеджерам, достигшим этих степеней, будет довольно легко этим похвастаться, «в мастерской инженеров-менеджеров у меня рабочая квалификация стратега» (более официально будет «квалификация по рабочему развитию — стратег», но «рабочая квалификация стратега» вроде как звучит помягче и внушительней).
- специалист — разобрался с обобщённым инженерным процессом и разделением труда. Это в значительной мере от того, что освоил руководства инженерного цикла (сейчас это руководства по системной инженерии, инженерии личности, системному менеджменту) как изложение некоторого набора теорий о лучших методах организации работы себя и других. Пока ничего не изменил в плане принятия методов из руководств всерьёз — не изменяет ни своей культуры инженерной работы, ни культуру работы других. Тем не менее, «знает теорию», то есть содержание текста руководства, начитан, эрудирован, имеет хорошо подвешенный язык. Разве что агентности по отношению к миру нет: собственных инициатив по изменению себя и мира к лучшему, которые выходят на уровень реальных изменений, а не просто размышлений.
- практик — освоил руководства инженерного цикла, принял всерьёз для изменения собственной культуры работы: «договорился со всеми», организовал до десяти своих подчинённых. Окружающие замечают, что стажёр поумнел, его производительность выросла, «стал другим человеком».
- мастер — прошёл программу, организует работу предприятия (от десяти человек, «договорил всех», включая не своих подчинённых, но и коллег, и даже людей выше себя по уровням иерархии: дело тут в не «начальственности», а в наставничестве и далее надзоре: умении

научить других людей работать по-новому для них, организовать по-новому, число этих людей, для которых меняется их способ работы, обычно насчитывает десятки человек). Мастера у нас получают право быть наставниками, ибо они уже имеют достаточный собственный опыт в том, как работать с использованием наших методов мышления и картины мира. Тут мы используем другое значение термина «мастер»: не просто тот, кто освоил какое-то мастерство, но как название степени квалификации в мастерстве. Несколько разных значений термина «мастер» раскрываются в руководстве по инженерии личности.
    - реформатор — это просто мастер с масштабами побольше: реформатор изменяет не только культуру работы своего предприятия, но и культуру сообщества, выходящего за пределы предприятия.
    - революционер (менее пафосно — «общественный деятель») — это ещё один рост масштаба: он даёт вклад (impact) в мировую культуру, в мировую цивилизацию, попадает в энциклопедию.
- **Программа исследовательского развития**: освоение интеллект-стека фундаментальных SoTA методов мышления, пока у нас в этой программе только одно руководство по интеллект-стеку методов мышления, но этих руководств планируется побольше.
    - исследователь: создание собственных прикладных или даже фундаментальных моделей мира, теорий, а также их документирование в форме руководств. Если теорию не документировать в форме руководства по деятельности, её много труднее принять всерьёз — и вновь созданная теория никак не повлияет на изменение мира к лучшему.

Моделирование: выбор рабочего проекта

Выберите рабочий проект, которым вы будете заниматься в ходе освоения руководства по системному мышлению. Проверьте, чтобы целевой системой было что-то физическое, что легко показать окружающим (не текст, не информационная модель, не исходный код программы! Если это что-то «информационное», то попробуйте понять, какой объект описывает эта информация, и так дойдёте до физического мира). Целевая система — это не поведение, не путаем бег и бегуна. Целевая система это не характеристика — не путаем «повышение мощности», «мощность» и ту систему, у которой мощность будет характеристикой, а уж «повышение» будет характеризовать то, что вы делаете с этой характеристикой. Если у вас систему использует «пользователь», постарайтесь его конкретизировать для определённого действия с системой (у игры не просто «пользователь», а «игрок»).

	Название проекта	Целевая система	Как называют того, кто будет её использовать?	Что основное делает система в момент использования?	Что основное делаете вы

Пример	Проектирование табуретки	Табуретка для бара	Седок	Поддерживает седоков весом до 100 кг, не опрокидывается, удерживает седоков на высоте барной стойки (стоять и сидеть — голова на одинаковой высоте)	Описываю конструкцию табуретки с точностью, достаточной для изготовления на мебельном заводе
Ваш проект					

Применимость системного мышления: везде и всегда

Последнее препятствие для того, чтобы мыслить системно — это просто отсутствие такого мышления, необращение при рассуждениях к типам системного мышления. Вы просто не ищете в мире целевые системы, не рассматриваете сначала их системное окружение в момент эксплуатации (даже если эксплуатация будет только в будущем), затем не рассматриваете их части-подсистемы, методы их создания, и только потом обращаете внимание на создателей. Автору встречались случаи, когда люди тратили много времени на освоение системного мышления и даже достигали некоторой беглости в его использовании в тот момент, когда им явно указывалось на необходимость каких-то системных рассуждений в сложной ситуации. Но в критических ситуациях собственных рабочих проектов они просто забывали его использовать! Это неудивительно и даже неспецифично для системного мышления: обычно люди знают, как хорошо выполнить то или иное дело, но только мастера реально используют это знание, часто абсолютно автоматически — на то они и мастера.

А не-мастера о правильных приёмах работы и мышления обычно знают, но просто забывают их применить, или им это лень делать, потому как неавтоматическое рассуждение S2 очень трудоёмко по сравнению с интуициями S1.

Вот это чувство потерянности после начала освоения нового мастерства (давайте поглядим на вас через пару недель — вы ещё будете изучать наше руководство? Тратить пару часов в день на работу по его наставлениям?), невозможность реорганизовать свою жизнь с выделением достаточного времени для освоения сложного метода работы (известно же, что иностранные языки в школе и вузе учат годами — но так и не выучивают. Но если заняться как следует, например, в целях эмиграции, или для использования в работе, то всё получается!), неиспользование результатов долгого обучения в рабочих проектах обычно связаны с одной причиной: непониманием, зачем эти новые обширные знания нужны, зачем использовать системное мышление. Живут же люди без этого системного мышления, и неплохо живут! Действительно, живут как в биологии: и бактерии живут, и простейшие, и насекомые, и млекопитающие, и люди. Причём люди живут очень по-разному, но живут!

Резюме тут простое: **если хотите меньше допускать ведущих к неэлегантной**

работе ошибок в сложных коллективных проектах, то освойте системное мышление, а потом используйте его в жизни, всю жизнь, во всех проектах. В какой-то реальной сложной ситуации привычка системно мыслить вас спасёт: вы не сделаете глупых ошибок и не потратите силы и время на переделку уже сделанного даже в тех ситуациях, которые окружающим вас людям будут казаться очень сложными. Вы получите конкурентное преимущество перед более медленными в их мышлении людьми (а иногда и AI-системами). Если в этом месте вы скажете «миллиардеры не знают системного мышления — но они миллиардеры», то вам надо на программу личного развития, про личный успех — это туда. Тут мы обсуждаем рабочее развитие. Без системного мышления сложный софт не заработает, ракета не полетит, в этих проектах не полагаются на удачу, тут полагаются на безошибочное мышление.

Главная ошибка в оценке применимости системного мышления — это принять его за прикладной метод мышления, срабатывающий только для какого-то систем какой-то одной природы, одного вида. Например, систем только одного масштаба и сложности, скажем, только для «железных» инженерных систем, или только для софта, или только для менеджмента. Нет, системное мышление — это набор приёмов мышления из фундаментальных методов мышления. Фундаментальные дисциплины/теории поэтому и называют трансдисциплинами, что они лежат «по ту сторону» от традиционных предметных/прикладных теорий/объяснений/дисциплин, дающих знания методов работы для систем какой-то одной природы, одного масштаба систем, какой-то одной предметной области.

Понятия системного подхода вводятся и используются во многих фундаментальных дисциплинах методов мышления интеллект-стека, начиная с физики, которая вводит само понятие «система». Эти понятия используются для управления вниманием/дают удобную систему типов для важных объектов в условиях наличия в конкретном проекте множества прикладных предметных областей и задействования тем самым множества прикладных методов мышления и действия. В проекте происходит инженерная работа с конкретными видами систем (розами, пароходами, транснациональными холдингами, клиентурами, городами, странами), требующая задействования огромного числа самых разных методов многими организационными ролями агентов-создателей. Создатели могут также создавать (часто не с нуля, но всё одно — менять воплощение) друг друга, образуя сложные графы создателей, где в узлах — создатели и узлы связаны отношением создания: «Иван обучил::создание Машу программированию станков, Маша запрограммировала::создание станок, станок изготовил::создание деталь №3».

Помним, что если даже вы занимаетесь розами, то систему-розу в системе-саду должна вырастить система-садовник, а систему-садовника тоже кто-то должен развить — и систем в рассмотрении у вас сразу стало четыре (если вы их записываете, то удержать во внимании вы легко сможете их все). Прикладные методы/практики/деятельности/культуры работы тут будут для изменения в физическом мире:

- роз (например, розы надо подвязывать — выполняется садовником),
- садов (например, разбивать на местности, определяя где там розы — выполняется садовником, но сады не получится подвязывать),
- садовников (например, их надо учить садоводству — выполняется тем, кто учит садовников, но садовников не получится подвязывать или разбивать на

местности),

- тех, кто учит садовников (например, их тоже надо учить преподаванию, чтобы они умели учить садовников, но мы это даже не упомянули, что кто-то это должен тоже делать — это будет пятая система, и дальше эту цепочку можно продолжать, «учителей для учителей» тоже ведь надо готовить).

Когда это записано, разобраться просто. Системное мышление помогает нарезать мир на объекты, которые удобно обсуждать — но главное, помогает найти важные объекты. Так, в примере про розы мы по факту «автоматически» задали вопрос про пятую систему (того, кто учит агентов, которые будут учить садовников), которая была вне внимания в начале рассуждения. Это огромная ценность: таракана в тарелке супа вы легко заметите, он там лишний, а вот отсутствие соли или кореньев в супе — нет. Суп просто будет невкусный, но чего в нём не хватает, почему невкусный? Заметить то, чего в проекте забыто, а должно бы быть — сила системного мышления в этом. Если в проекте забывают удерживать внимание на целевой системе в её окружении на момент эксплуатации (это трудно, ибо системы-то ещё нет!), то получается на выходе проекта не столько хорошо работающая система, сколько система, которую легко и удобно сделать (если не помнить, что суп при готовке придётся есть, то можно не посолить суп, не положить в него кореньев — это ж «лишняя работа», поэтому о ней легко просто забыть). Системное мышление это обязательно поправит, напомнит о забытом. Девять раз из десяти это будет банальным, а в десятый раз — спасёт проект.

Управление вниманием может быть довольно сложно даже в простом проекте, где один человек играет множество ролей и управляет своим собственным вниманием, распределяя его по объектам нескольких предметных областей, которые могут быть в этом проекте. Поле с пшеницей и агрономия, команда фермерского хозяйства и менеджмент как инженерия предприятия, маркетинг как инженерия клиентуры, финансовая инженерия для организации инвестирования в фермерское хозяйство, инженерия киберфизических систем для агроробота — всё, «простой» проект сразу становится не очень простым. Системное мышление как минимум позволяет эту «непростоту» отследить и организовать мышление так, чтобы справиться с удержанием в голове (и записях в компьютере в помощь голове!) всего проекта в целом, во всём его многообразии.

Вопрос типа «как я могу оценить применимость системного мышления для своей конкретной задачи-проблемы» как раз показывает это непонимание разницы между фундаментальным знанием и прикладным знанием. Давайте заменим «системное мышление» на «управление вниманием» и спросим ещё раз: «как оценить применимость управления вниманием в моей конкретной задаче-проблеме»? Вроде как так не говорят, смысла ведь в таком вопросе нет, ибо управлять вниманием нужно всегда, в любых задачах-проблемах. Вопроса о «применимости» тут просто не возникает. Так же точно не возникает вопроса о применимости логики, ибо фантазировать в рассуждениях на работе нехорошо, логику нужно задействовать во всех рассуждениях. Вот и понятия системного подхода применять (то есть присваивать объектам проекта типы системного мышления) нужно так же: всегда, как логику, как управление вниманием! Собственно, это и есть управление вниманием: понятийное (направленное типами мета-мета-модели) управление вниманием. Вот прямо так и думать в ходе проекта, всё время направляя своё внимание на объекты, указываемые типами из руководства по системному мышлению. Вы как раз это руководство сейчас и изучаете, хотя **до собственно**

понятий системного мышления (система, окружение, эмерджентность, ...) мы ещё не дошли, они будут разъясняться со следующего раздела.

Наше руководство системного мышления наставляет на рабочее системное мышление, поэтому подход в нём инженерный (цель: изменение мира, даже если речь идёт об изменении людей — это всё равно инженерия). Это не руководство по истории системного мышления, поэтому мы не приводим подробные описания общей теории систем, не цитируем Берталанфи, не рассказываем много о теориях систем других мыслителей прошлого века, не рассказываем много о причинах неудачи кибернетики как чуть ли не синонима системной науки. Руководство наставляет в том, о чём надо думать, когда что-то делаешь. Если работать по этому руководству, то ровно так и надо думать в ходе работы, как говорят его наставления! Но руководство не наставляет в том, как именно делать что-то прикладное: какими прикладными методами выполнять работы проекта, какие в проекте занимать роли. Это оставляется на ваше усмотрение.

Руководство по системному мышлению — это не руководство по методологии, системной инженерии, инженерии личности, инженерии организации (менеджменту). Это будут другие руководства, которые вы освоите после того, как освоите системное мышление. В нашем руководстве мы наставляем только о том, как думать о мире в терминах систем. За пределами нашего руководства также остаётся всё не относящееся к понятиям системного подхода из фундаментальных дисциплин методов мышления интеллект-стека. Например, мы не рассказываем подробно теорию принятия решений из рациональности, не даём основы риторики. Руководство также не по прикладным методам работы в самых разных предметных областях (военное дело, медицина, генная инженерия, ракетостроение, создание AI-агентов, программная инженерия, садоводство). Наставления руководства и без этого достаточно объёмны.

Системное мышление оказывается практичным/культурным/деятельным/инженерным/стильным, причём применимым повсеместно: оно усиливает ваш интеллект в любых проектах, а не только в проектах какого-то определённого типа. Это прикладное мышление ортодонтии или бурения в вязких грунтах применимо только для узкого класса задач. Системное мышление нельзя сравнивать в этом плане с прикладными методами мышления, которые позволяют рассуждать об объектах в проектах какого-то определённого класса, в каких-то определённых предметных областях. С системным мышлением вы системно думаете во всех проектах, в том числе и в ситуациях переключения с проекта на проект, формулирования/постановки новой задачи/проблемы, организации нового проекта, разбирательства с новой для вас предметной областью.

Освоение работы по наставлениям руководства по системному мышлению можно сравнить по затратам времени с семестровым вузовским курсом (4 европейских кредита, примерно 30 часов времени обучения в кредите[121]). Это означает, что нужно выделить по 10 часов на 12 недель на чтение материала, выполнение заданий, чтение дополнительной литературы, встречи с наставником. Это не слишком сложно, нагрузка хоббийного уровня: в какой-нибудь школе танцев это два двухчасовых урока в неделю, одна четырёхчасовая вечеринка и ещё два часа

[121] https://ru.wikipedia.org/wiki/Кредит_(образование) и https://ru.wikipedia.org/wiki/Европейская_система_перевода_и_накопления_баллов

на самоподготовку. В любом случае, этот режим требует переустройства своего распорядка дня. Планировать это всё на «в свободное время» (которого как не было, так и не будет) не получится, важно сделать расписание, затем важно его соблюдать. На эту тему можно использовать наставления руководства по рациональной работе. Если выдерживать выделение 10 часов в неделю на освоение системного мышления не удаётся, возможно, вам надо вернуться к руководствам программы не рабочего, а личного развития: там учат учиться.

Избегайте освоения нескольких наших руководств «вперемешку». Вам дана определённая последовательность их прохождения, и она важна. Не надо изучать формулы ньютоновской физики до освоения арифметики. Точно так же не надо пытаться освоить системную инженерию до того, как освоили системное мышление — не будет понятно даже слово «системная»! Руководства составлены именно для последовательного прохождения.

Чтобы учесть особенности человеческого мозга (кривые забывания), а также дать наставления в самых разных контекстах (мозг — это нейросеть, учится на предъявлениях повторяющихся в разных контекстах паттернов), в руководства встроены повторения. А ещё в руководства встроены блоки с ответами на вопросы, которые авторы слышали от инженеров-менеджеров, работавших по предыдущим версиям руководств. Если у вас появился вопрос, то наверняка ответ на него вы встретите или через несколько страниц текущего руководства, или в следующем руководстве. Так, материалы нашего руководства по системной инженерии будут повторяться в руководствах по методологии, системной инженерии, инженерии личности, системному менеджменту. Наставления руководства по рациональной работе в нашем руководстве тоже будут повторены, а также многократно использованы.

После освоения работы по всем этим руководствам в целом станут понятней многие наставления руководства по системному мышлению. Так что рекомендуем не застревать надолго на каком-то непонятном месте нашего руководства, просто идти при знакомстве с руководствами дальше, но потом всё-таки повторно пройти по тексту руководств, начиная с руководства по рациональной работе и заканчивая системным менеджментом. Это должно быть не столько «обучением», сколько «работой», с обязательным выполнением заданий по работе в вашем проекте (возможно, в этот момент у вас уже будет другой рабочий проект, это неважно! Важно, что вы осваиваете методы мышления, методы работы).

Повторим: делайте предлагаемое в заданиях моделирование в рабочее время, оно ведь касается использования системного мышления в рабочих проектах — вот и работайте, думайте над рабочими вопросами в рабочее время, у нас руководство/регламент/стандарт/«описание мыслительного процесса» (все синонимы тут с нюансами), а не «учебник для изучения в отрыве от работы».

Задания: системное творчество

Поставьте отметки о достижении состояния:

1. Написан пост с описанием примеров творчества в ваших проектах.

4. Воплощение и описание системы

Воплощение, описание и документация системы

В системном подходе очень важно быть **адекватным** в мышлении: понимать, говорим ли мы о физической реальности нашего мира, привязаны ли мы к ней, или просто фантазируем о каком-то абстрактном мире идей, который может и не соответствовать реальности. Если мы хотим надёжно менять физический мир в соответствии с нашими замыслами, если мы говорим о человеческой деятельности, то нам нужно как-то привязать все наши рассуждения к физическому миру, в конечном итоге иметь дело с физической реальностью[122].

Адекватность системного мышления обеспечивается тем, что когда мы говорим о системе, то имеем в виду **воплощение системы** (system realization — тот же корень, что real, реальный, буквально речь идёт о существовании в реальности, reality), причём обычно **в момент эксплуатации**, когда система работает/функционирует, взаимодействуя с окружающим миром.

Система понимается всегда как конкретное воплощение системы в физическом мире — индивидуальный, уникальный физический объект. Например, это фирма Apple (все её здания, сооружения и сотрудники фирмы в тот момент времени, когда они в ней работают), топливный насос с серийным номером #12345, установленный на авиадвигателе #5678, перформанс (работа танцоров) танца «Барыня» на сцене Усть-Урюпинского театра вечером 24 октября 2029 года (воплощение системы в будущем нас вполне устроит).

Как узнать, что конкретная воплощённая система существует в физическом мире? Мы будем считать, что в физическом мире присутствуют (в том числе в настоящем, прошлом и будущем — время тут не важно!) только те объекты, которые занимают какое-то место, **объём** какой-то причудливой формы **в нашем пространстве-времени**, принимая идею Эйнштейна про существование физического мира в четырёхмерном пространстве-времени. Есть какой-то кусок пространства-времени, который мы называем **система/system** и выделяем вниманием из его **окружения/environment**, при этом у системы есть **граница/boundary**, проходящая в пространстве-времени. Поскольку тут пространство-время, то неважно, где в пространстве (тут или в соседней галактике) и неважно, где во времени (например, система будет существовать пять минут, но через сто лет от «сейчас»).

Тем самым конкретная система имеет некую протяжённость в пространстве (то есть форму — размер, длину, ширину, высоту, радиус) и во времени (то есть имеется момент, когда она начала существование, и момент, в который она закончит существовать).

А ещё эта система имеет изменяющиеся **состояния/states**. Системы меняются по мере взаимодействия с окружающими их системами, которые тоже находятся в физическом мире, а внутри них меняются взаимодействующие в них подсистемы. Например, «дверь::система открыта», «дверь::система закрыта», «дверь::система выломана» — это всё разные состояния двери. Состояния крайне важны, ибо работы по разным методам могут приводить к разным состояниям систем.

---

[122] В философии эту привязку фактов к реальности описывают как grounding, https://plato.stanford.edu/entries/grounding/

Например, «открыть дверь номер четыре в 3 часа 15 минут 24 мартобря»::работа может перевести дверь номер четыре из состояния «дверь закрыта» или в состояние «дверь открыта» (например, «открыть ключом, повернуть ручку и нажать рукой»::«метод работы») , или «дверь выломана» (например, «как следует двинуть ногой в районе замка»::«метод работы»).

Системное (то есть представляющее различные части мира как системы из системного подхода) рассмотрение мира физично/реально. Понятие «система» пришло из физики[123]. Физика подбирает для выражения свойств объектов окружающего физического мира математические объекты, которые ведут себя похоже в мыслительных экспериментах. Семантика — это трансдисциплина, которая связывает между собой физику с её физическими объектами (предметы), понятия как математические/идеальные объекты и знаки как паттерны на каком-то носителе, которые обозначают предметы и понятия (знаками занимается дисциплина семиотики). Вы уже владеете азами семантики из руководства по рациональной работе, в нашем руководстве по системному мышлению будет раскрыто приложение семантики к системному подходу, а в руководстве по интеллект-стеку из программы исследовательского развития будет дано много больше подробностей про семантику, физику, математику, семиотику и их связь.

Тем самым мы чётко различаем (и это различие — предмет изучения трансдисциплины семантики):

- **воплощение системы** (system realization), занимающее объем в физическом пространстве-времени, как предмет физики.
- **описание системы** (system definition, часто называются ещё и определениями системы, не путайте со «словарными определениями»!) как идеальный/математический/ментальный/абстрактный/понятийный/концептуальный объект. Поведение описаний как идеальных объектов изучается математикой.
- **документ с описанием системы**, как выражение описания системы на физическом носителе. Способы выражения (паттерны на физическом носителе) изучаются семиотикой, «наукой о знаках».

В физическом мире есть какие-то ненулевого размера участки пространства, а в математике — математические точки, в семиотике — паттерны на носителе, знаки/символы. Конечно, есть и распределённые/distributed представления, которыми занимаются в методах познания представлений/representation learning (мы переводим learning в методах AI как познание). Но мы не будем заниматься тем, как устроено интуитивное быстрое мышление S1, которое и использует распределённые представления. Мы будем работать с локальными/символьными/symbolic/знаковыми представлениями, которыми занимается медленное мышление S2. В распределённых представлениях очень трудно найти ошибки, они не локализованы в буквальном смысле слова. А вот в символьных представлениях ошибки найти много легче. Поэтому инженерные задачи надёжней решаются в компактных локальных представлениях с немногими знаками, а не в распределённых представлениях (например, нейросетях — живых в мозгу людей или компьютерных в системах AI).

Поведения математических объектов и соответствующих им физических объектов всегда отличаются, поэтому физика в ходе своего развития заменяет одни

[123] https://ru.wikipedia.org/wiki/Физическая_система

математические представления другими, которые ведут себя более похоже на то, как ведут себя физические объекты. Скажем, сумма углов треугольника вблизи Земли чуть-чуть больше, чем 180° даже не из-за шарообразности земли, но из-за искривления пространства-времени массой Земли. Поэтому нельзя будет реализовать GPS навигацию, если полагаться на геометрию Эвклида в физических расчётах инженерного проекта. Поэтому физика заменила в теории относительности геометрию Эвклида на Риманову геометрию.

Традиционно соответствием описаний как идеальных/математических объектов/понятий, выраженных/presented какими-то паттернами/узорами/шаблонами/повторяемостями на одних физических объектах (документах) и представляемых/represented ими других (описываемых) физических объектов/систем занимается семантика. Впрочем, семантика может заниматься и описанием описаний (скажем, вы можете описать математический язык, на котором описываете физический объект). Но всегда эта цепочка описаний упирается в описание реальных физических объектов, иначе нельзя будет договориться о том, что все эти описания означают, без заземления/grounding с описаниями работать нельзя.

Семантику иногда считают разделом информатики/computer science наряду с алгоритмикой, подробней об этом можно узнать, например, из книги Дэвида Дойча «Начало бесконечности. Объяснения, которые меняют мир»[124], где он рассматривает отношения между физикой, математикой и computer science. Больше наставлений по задействованию методов семантики в мышлении вы можете найти в руководстве по интеллект-стеку методов фундаментального мышления в программе исследовательского развития.

В системном мышлении важно, что все описания на свете — это в конечном итоге описания систем, ибо весь физический мир мы представляем как воплощения разных взаимодействующих систем. Оговорка «в конечном итоге» — это если учесть, что иногда одни описания описывают другие описания.

Отдельно надо сказать, что описания тут могут быть длинным синонимическим рядом — как всегда, с нюансами. Например, это могут быть сведения (описания, чтобы как-то их задействовать для принятия решений), информация (описания, в которых акцент даётся на формальные аспекты презентации и отказ от рассмотрения содержания, например, «обработка информации», теория информации в передаче информации), модели (только важное в каких-то аспектах) и т.д. Как понять, что имелось ввиду? Как всегда: из контекста, при этом помнить, что разные люди используют разные синонимы — иногда, чтобы подчеркнуть эти нюансы, а иногда и просто так, ибо честно считают, что это синонимы, и нюансов нет. В английском «описание системы» у инженеров — system definition, но по-русски «определение системы» будут путать или с математическими «определениями» (один абзац текста), или со словарными «определениями», хотя в английском там нюанс в «задании», «предписании», речь идёт о проекте/design системы главным образом.

Сами описания как ментальные/идеальные/понятийные/математические/абстрактные объекты не имеют места в реальном мире. Нельзя сказать, что математический/ментальный

[124] https://www.litres.ru/devid-doych/nachalo-beskonechnosti-obyasneniya-kotorye-menyaut-mir/

объект «число 300», определяющее высоту в метрах (воплощения) системы «Эйфелева башня», находится где-то в реальном мире и имеет собственную длину-ширину-высоту. Если вы укажете на это «число 300» и скажете, что оно существует и имеет свой объём — то вы укажете не на само число, а на **документ** как **носитель информации**, который своими паттернами/узорами как формами материала (паттернами/узорами частиц краски, или прозрачности материала, или ещё как-то) кодирует это число. Тем самым место в физическом мире занимает не «300» как число, не часть описания Эйфелевой башни (число 300 описывает Эйфелеву башню), а материальный объект кусочка **документации** (system description в отличие от system definition) Эйфелевой башни, т.е. носитель/документ с записанной на нём информацией/описанием. **Документ — это абстрактное/идеальное/математическое/понятийное описание системы, записанное на каком-то физическом носителе информации. Один или несколько документов обычно называют документацией системы.**

В документации знаками на носителе кодируются понятия/концепты/математические/абстрактные объекты, репрезентирующие какие-то (в конечном итоге, потому как иногда бывают описания описаний) различные свойства физических объектов. Вновь и вновь разными словами повторяем эту троицу: знак, понятие, означаемое.

В системном мышлении вам потребуется владение семантикой, что даётся руководством по рациональной работе. Горе вам, если вы плохо понимаете этот материал и будете путать автомобиль как целевую систему и фотографию автомобиля как документ-носитель описания этого автомобиля. Фотография материальна, но материальность её важна только как возможность нести на себе описание физического объекта-автомобиля, материальность носителей информации не должна нас путать — это не материальность воплощения изображённой/описанной/документированной системы!

Типичная ошибка — это считать целевой системой какой-то документ с описанием системы. Если вас не интересует содержание документа (вы — типография, или курьер, или материальный архив: их не волнует содержание, информация, только физическая форма документа), то можете считать целевой системой документ или даже всю документацию. Если вы интересуетесь тем, что описано в документе — то целевая система точно не этот документ, вас волнует описание, а не сам документ. Вот и ищите, что описано в документе, не отвлекайтесь на сам документ!

Экран или бумага фотографии, или даже компьютерная память с файлом фотографии — это носители информации, не сама «фотография» как информация на ней! Порвать бумажную фотографию (документацию на сфотографированного человека::система) — это не порвать человека, изображённого на фотографии (воплощение системы), и даже не порвать изображение человека на этой фотографии (описание системы), ибо оно может сохраниться на сотнях других носителей, бумажных и электронных. Это порвать бумажку, а не изображение и не человека! Наказание за «сожжение флага» — это наказание за запретный образ мыслей, а не за порчу имущества!

Абсолютно такое же действие — выключить экран с фотографией, или стереть файл с фотографией (это не так символично, как порвать бумажную фотографию, но по сути — то же самое. Абстрактные/математические/понятийные/концептуальные

объекты не горят, представление их на носителях обычно легко копировать, и уничтожение документа, а хоть и электронного — это не уничтожение описываемого документом объекта, а часто это и не исчезновение документированного описания, доступного на многих копиях документа). Концепт/понятие/математический объект «число 1» по большому счёту даже не зависит от шрифта и прочих особенностей отображения, он абстрактен, может быть выражен множеством способов, множеством начертаний знаков. И помним, что концептов в физическом мире нет, они абстрактны/математичны. Информационные модели систем (когда-то в прошлом веке их называли чертежами) абстрактны/идеальны/понятийны, документированы они во множестве копий, а описывают они воплощение системы, то есть физический мир. Информационная инженерная модель космодрома, отправляющего людей на ракетах на Марс — это описание возможного/possible (ибо речь идёт о возможном будущем) физического мира.

Описания системы легко отличить от воплощений системы и документов системы — они не занимают места в физическом мире, у них нет объёма и координат в этом мире, они **абстрактны/**«идеальны»/нематериальны/математичны/понятийны как противоположность материальному/физическому/воплощённому. Системы и системные документы (документы о системах) материальны, они занимают место в физическом мире, но мы при этом не путаем системы и документы с описаниями систем.

Людей в конечном итоге интересуют воплощения системы (дома, самолёты, мастерство менеджера, сеанс игры, деньги), а описания и документация системы (информационные модели, листочки бумаги с письменами, изображения на экране) их интересуют ровно постольку, поскольку без них воплощение системы трудно сделать, особенно когда речь идёт о системах, создаваемых многими людьми:

- Документация — это внешняя память, «экзокортекс». Биологическому мозгу не верим, верим записанному на какой-то более стабильный носитель. Документация всегда в экзокортексе, а не неокортексе (коре головного мозга). Для AI-агентов документация может быть и с весами нейросети, но она вполне копируема и стабильна во времени.
- Документация несёт на себе модели (только важное в описании). Это уже результат мышления: про систему сказано не «всё» (что невозможно дальше обсуждать), а только «кое-что, важное» (что можно обсудить, проверить на отсутствие ошибок, принять руководством к действиям по изменению физического мира, а не только изменению описаний: изменения описаний на носителях — это не изменения физического мира, разве что вы типография или изготовитель компьютерных мониторов).
- Проектировщик из коннекционистской/нейросетевой (в мозгу человека или AI-агента) модели готовит символьную модель системы и размещает её на внешнем носителе («экзокортекс») для того, чтобы технолог производства мог подготовить программу изготовления с учётом какого-то конкретного оборудования производственной платформы/завода. Разделение труда: над созданием системы думают разные агенты, поэтому агентам надо как-то передавать результаты своего мышления. Это делается через системную документацию, хотя сегодня чаще всего **документация — это компьютерные модели, а не бумажные документы, и документы в**

таком случае не передают/пересылают, а получают к ним доступ.

Различение для системы воплощения (физично), описания (не физично), документации (физично как носитель информации, это не описанная система) нужны в том числе и для того, чтобы предотвращать распространённые ошибки людей, играющих роли проектировщиков в проектах по созданию систем:

- Когда люди даже и не думают создавать систему, а пишут очередную «концепцию». Нужно проверить, описывает ли концепция что-то существующее в реальности, а не просто описывает создание документации и избегает описания воплощения системы. Скажем, «концепция развития региона»: это утопия? Какие есть инструменты реализации концепции? Ведь «регион» — это абсолютно материальный объект. Кто и что должен сделать (выполнить работы, имея для этого ресурсы и полномочия распоряжения ресурсами), чтобы перевести регион из состояния «было» в «стало» согласно концепции? Точно ли «концепция» описывает физический объект «регион», который будет существовать в будущем?
- Когда разработчики проектной документации считают, что их целевая система — документация. Инженеры из конструкторского бюро или проектного института часто путают воплощение системы в момент, когда она уже сдана в эксплуатацию с информационной моделью такой системы, которую они «нафантазировали» без шанса на её воплощение/реализацию/изготовление. Нет, целевая система — это всегда воплощение системы, без воплощения системы не стоит и заводиться с её описанием (результат мышления, ментальный объект), документированием этого описания (результат вывода описания на внешний стабильный носитель информации). Без намерения сделать систему речь идёт о «художественных описаниях», в них могут быть любые ошибки, и их даже не будет желания проверять. Надо спросить проектировщика, что будет результатом его работы. Если проектировщик, например, здания, ответит «здание», то всё ОК. Если такой проектировщик ответит «информационная модель здания» — то бить тревогу, ибо мысль проектировщика явно не дотягивается до воплощения системы, ограничивается описанием и документацией.
- Разработчики исходного кода программ (это описание воплощения системы) тоже считают, что их обязанности заканчиваются тогда, когда они сдали документацию (исходный код с любым количеством ошибок) в систему хранения версий (носитель — внешняя память компьютера хранения версии). Нет, их целевая система должна быть — работающая программа (меняющая своё состояние! Переменные программы меняют своё состояние в памяти компьютера! Это физический процесс!), а не исходный код на любых его носителях. Более того, надо ещё проверить, нужна просто работающая программа, или работающая программа в комплекте с людьми, которые могут выполнить полезную работу с этой программой — тогда надо будет учесть ещё и «софт в головах людей», который надо тоже разработать, а потом изготовить (кто-то должен озаботиться мастерством людей, которые работают с воплощением программы в момент эксплуатации программы, а не её исходным кодом).
- Менеджер, разработавший план мероприятий какого-то проекта (описание работ, существующее в виде документации проекта в программе проектного управления). Воплощением системы для проекта (набора работ) будет

организация с необходимыми ресурсами и полномочиями, выполняющая работы согласно плану, а не сам этот план. Более того, целевой системой, скорее всего, будет та система, которая получается в ходе этих работ проекта, а не система-организация, проводящая работы! Составитель плана должен прежде всего думать о той системе, которая получится в результате выполнения плана!

- … и таких ошибок «описания вместо воплощения», связанных с неумением или даже нехотением проектировщиков отвечать за изменение физического мира к лучшему, а не только за высказывание идей по поводу таких изменений — множество. На разницу системы, её описания, системной документации всегда нужно обращать внимание. В частности, если речь идёт о **работе аналитика в отличие от работы инженера**, то больше вероятность получить не выходящее в физический мир «описание» (аналитический отчёт, очень редко — какие-то «рекомендации», не входящие даже в аналитический отчёт, но излагаемые отдельно) как конечный результат, но не «изменение физического мира». Инженеры меняют мир, аналитики (по определению!) мир не меняют. Нужно проверять, есть ли ещё в проекте «синтетик» кроме «аналитика», воплощение нужно искать у него. **Если проект в конечном итоге не меняет физический мир, то такой проект можно вообще не делать, ведь ничего в мире не изменится! Ну, или делать в порядке художественного творчества, затраты списывать по статье «развлечение».**

Результат работы проектировщика атомной электростанции — в конечном итоге воплощение работающей атомной электростанции, а не описание атомной станции, даже документированное в форме информационной модели атомной станции в полусотне компьютеров. Результат работы хореографа — это в конечном итоге перформанс танцора (исполняемый в конкретное время в конкретном физическом месте во вселенной), а не описание хореографии (последовательности движений) в виде идей в головах танцоров, или тем более не документ (бумажный или электронный) с описанием хореографии. И это несмотря на то, что проектировщик сам не строит атомные электростанции, а только их описывает, а «хореограф» в его изначальном значении тоже «описатель» танцевального перформанса (от др.-греч. χορεία — хороводная пляска, хоровод + γράφω — записывать, писать. Первоначальное значение хореографии — это отнюдь не сочинение и постановка танцевальных перформансов, а именно искусство записи танцевания как наблюдаемого поведения танцора[125], грубо говоря, это был не «писатель», а «писарь»).

Карта — это не территория. Люди ходят не по карте, а по территории. Карта — это только описание территории, которое может быть представлено в разных документах (электронных или бумажных, или пластиковых), и это верно для всех описаний/документов, не только для географических карт. На географическую карту смотрят, по территории ходят.

Карта коктейлей — это не коктейли, её не пьют. Карта отражает понятийную/ментальную/концептуальную/математическую/абстрактную структуру мира физических коктейлей, документ карты находится в физическом мире, а сама карта в мире абстракций/математическом мире, даже если на документе карты

[125] https://en.wikipedia.org/wiki/Dance_notation

изображены картинки настоящих коктейлей, а сам документ карты вполне физичен. Описания не занимают пространства-времени, они абстрактны/математичны, даже при условии, что описания нельзя явить в физический мир без использования физического носителя информации (ибо информация уже физична, ей нужен носитель). Карта как описание того, что она отражает, не занимает пространства-времени. По ней нельзя «постучать», на неё нельзя «показать пальцем», как нельзя постучать по «числу 300», но можно постучать только по «изображению числа 300» на каком-то носителе — по документу, носителю информации.

Если же говорят, что карта занимает место/объём в пространство-времени, то речь идёт не о самой карте как информационном, абстрактном объекте, а о материальном носителе карты, документе — бумаге и паттернированной краске. Нарисованные на карте объекты не существуют в физическом мире. Постучать по коктейлям на карте коктейлей нельзя, постучать по кускам суши или моря на географической карте нельзя, постучать по атомному реактору в информационной модели атомного реактора нельзя. Постучать и показывать пальцем можно только на участки бумаги (или участки экрана, если это экран, или участки магнитной памяти, если это магнитная память), а не на изображённые на карте объекты. Карта-на-носителе документирует какие-то воплощения систем или другие описания, которые в конечном итоге всё одно должны будут дотянуться как-то логически до физического мира — иначе нельзя будет проверить их нефантазийность.

Карта коктейлей в данном случае — не система, а только документация материальной системы «ассортимент ресторана» (system description/documentation), а информация на ней — описание системы (system definition).

…

А вот сами коктейли, документируемые картой/описываемые информацией на карте — это системы (воплощения системы: **продукты, изделия, предприятия, люди, оборудование**), они занимают место в пространстве-времени, по ним

можно постучать, на них можно показать пальцем. Коктейли даже можно выпить, а вот картинки коктейля не выпьешь. Информационная модель атомной станции электричества не вырабатывает. Описание танцевального перформанса на бумаге или в компьютере не вызывает тех же эмоций у танцора и его зрителей, что само танцевание.

Моделирование: воплощения, описания, документы с описаниями

Для ваших настоящих или прошлых проектов (или знакомых вам проектов ваших друзей) заполните табличку примерами пяти воплощений системы, описаний этой системы и документов с описаниями этой системы. Достаточно одного примера описания и одного документа с этим описанием на каждое воплощение системы, но можно и больше: у одного воплощения может быть много описаний, одно описание может быть документировано на самых разных носителях информации/в самых разных документах. Пишите в таблицу только названия спрашиваемого, не нужно вписывать туда полные тексты и диаграммы!

Номер п/п	Воплощение системы	Описание системы	Документ описания системы
1.			
2.			
3.			
4.			
5.			

Описания

Но зачем нам нужны описания, ментальные/идеальные объекты?! Нас будут интересовать изо всех видов описаний главным образом:

- Символьные/знаковые/языковые описания в отличие от коннекционистских/нейросетевых. Нам нужна коллективная разработка, то есть возможность доступа нескольких агентов к одному описанию как для исправлений, так и для ознакомления. **Символьные описания легко отчуждать (документирование), копировать и пересылать в ходе коллективной разработки**. Описание, находящееся в нейросети агента, отчуждается трудно — там вместо «копирования, пересылки» работает «обучение», совсем другой механизм передачи описаний.
- Описания в рамках теоретической теории понятий (theory theory, объекты и отношения, работа медленного мышления S2), а не прототипной теории понятий (prototype theory, работа с метафорами и аналогиями, concept blending, работа быстрого мышления S1). **Формальность описания нужна для облегчения проверки.**

Документировать описания нам нужно для (коллективно устроенного) мышления: они помогают удерживать внимание на важных объектах в физическом мире (документ/внешняя память удерживает внимание разных агентов-создателей систем на тех объектах, которые мы в нём описали). В том числе эти документированные описания можно использовать и для переноса опыта из одного

проекта по созданию систем в другие проекты, если они описывают не экземпляры системы, а их тип (множество систем какой-то одной природы, вида/сорта систем). Если вы сделали информационную модель табуретки, то это будет информационная модель не одной табуретки, вы можете изготовить десяток табуреток по этой информационной модели — и использовать эти десять табуреток в десяти разных проектах.

Когда говорят «описание», то чаще всего имеют в виду «документированное описание» и даже «документ с описанием». Но остерегайтесь: иногда «описание» вовсе не означает того, что оно документировано, иногда описание есть (понятие, ментальный объект), а документа с этим описанием (физический объект) — нет. **Когда слышите «описание», определяйте по контексту, что имеется в виду: собственно описание, или документ с описанием.** В нашем руководстве мы тоже будем использовать «описание», когда речь будет идти о «документе с описанием». Но вы должны чётко различать описание (ментальный объект), документ с описанием (физический объект, обычно со знаковым описанием), описываемую систему (иногда через цепочку «описаний описаний», но всё-таки в конечном итоге — система, то есть выделенный из мирового пространства-времени какой-то границей четырёхмерный кусок, описываемая вещь).

Мышление в создании/construction каких-то новых систем, в познании/learn и последующем изменении каких-то уже существующих/brownfield систем обычно происходит не для отдельных экземпляров систем, а сразу для множеств систем, то есть для типов систем. Конечно, можно представить ситуацию, когда описывается конкретный экземпляр системы в конкретном экземпляре окружения в конкретное время (например, в записи базы данных поломок насосов на какой-то подстанции или записи в базе данных землетрясений, или записи в истории болезни какого-то пациента, или события в летописи какой-то страны), но это обычно записи времени эксплуатации системы, а не времени разработки. **Разработка (замысливание, проектирование, планирование изготовления) всегда ведётся не для конкретных экземпляров систем, а «в общем случае», то есть для множества/типа/вида/класса подобных систем, даже если речь идёт о создании одной системы (скажем, Эйфелевой башни или новой страны).**

Множество/класс/тип/вид/сорт — это математический/ментальный/понятийный объект. В математике множество из одного объекта не эквивалентно этому одному объекту: свойства множества отличаются от свойств самого объекта в этом множестве.

В системном мышлении то же самое: нас интересуют отдельные конкретные системы, воплощённые в нашем физическом мире, но моделируем их с задействованием присвоения типов/классов. Мышление ухватывает что-то общее во всех ситуациях, мышление происходит не для конкретных систем, о которых мы знаем разные факты. Мышление происходит чаще всего для **классов/типов/множеств/видов/сортов** систем/воплощений_систем/экземпляров_систем. И, конечно, ментальные объекты (множества, типы) легко изменять: для изменения абстрактных объектов не нужно тратить энергию на перетаскивание вещества воплощения системы, нужно тратить энергию только для изменения вещества для воплощения знаков, это неизмеримо меньшее количество.

Информационные технологии работают с описаниями, так что если вы имеете дело

с какими-то компьютерами и компьютерными сетями, то остерегайтесь перепутать документированное описание системы (информационную модель на каком-то материальном носителе) и воплощение системы. Вы легко сможете сообразить, что фотография человека — это не человек, на ней изображённый. Чуть трудней будет сообразить, что табличка с данными температуры ракетного двигателя на листочке бумаги — это описание ракетного двигателя, а сам листочек бумаги тут не важен. Но вот если речь идёт о небольшой базе данных, в которой описывается поведение ракетного двигателя в серии испытаний — вот тут наступает ступор, ибо сама база данных представляется её разработчику какой-то системой, которая будет казаться целью вашей работы, целевой системой, воплощением системы, материальным/физическим объектом, итогом работы. Нет, база данных тут — документированное описание ракетного двигателя, просто в менее привычной форме, документированное не на бумаге. Даже если с этой базой данных рассмотреть программу управления ракетного двигателя и назвать это **цифровым двойником/digital twin**[126], то интересовать вас всё-таки будет поведение **физического двойника**, то есть сам ракетный двигатель. И при разработке такой базы данных и всех программ вокруг неё надо в первую очередь думать именно о двигателе::физический двойник::подсистема! Если не удерживать в голове разработчика этот двигатель, то база данных легко может оторваться своим содержимым от описания реального физического мира — и оказаться бесполезной, а то и вредной. Система тут будет состоять из двух подсистем — цифрового двойника::софт (всё-таки взятый с учётом аппаратуры, софт без вычислителя не бывает) как одна подсистема и физический двойник как другая подсистема. Если вы не понимаете, какую систему вы делаете и заняты только своей подсистемой цифрового двойника (впрочем, и физического двойника тоже), то у вас будут огромные проблемы в проекте. Когда две подсистемы физического и цифрового двойников начнут работать вместе, успешной работы системы в целом не получится — и денег поэтому у её разработчиков не будет (а если их успели выплатить, то деньги даже могут отобрать! Ничего же не работает, за что платить?!).

Умение понять, какая система описывается в физической реальности, если вам показали описание в виде каких-то структур данных, документированных в компьютере — это ключевое умение для всех, кто встречается с современными информационными технологиями. Если у вас CRM (customer relationship management[127]) система, то её базы данных подробно описывают клиентуру как целую систему в физическом мире, а части клиентуры — это отдельные клиенты (возможно, сгруппированные в какие-то сегменты, но это сейчас неважно, пока нам важно разобраться, что клиентура — это система в реальном/физическом мире, а компьютерная CRM-система по отношению к этой клиентуре — примерно то же, что фотография человека по отношению к человеку, документ с описанием с удобными средствами просмотра документа).

Это не означает, что в какой-то момент нас не будет интересовать программная система как физический объект. Будет, как бумажная книга интересует типографа. Но мы не обсуждаем с типографом сюжет книги. Так и с администратором датацентра, который нужен для физического существования компьютерной программы, мы не обсуждаем её «сюжет». Интересуйтесь не самой программой,

---

[126] [126] https://ailev.livejournal.com/1570630.html

[127] https://en.wikipedia.org/wiki/Customer_relationship_management

интересуйтесь теми воплощениями системы, которые описывают данные программы, на изменения которых в физическом мире хотят влиять авторы программы (скажем в CRM программе авторы программы хотят влиять на клиентуру, а не просто на содержимое тамошней базы данных. Если при помощи CRM-системы повлиять на клиентуру не получится, то можно её вообще не использовать в проекте, вообще не разрабатывать, не изготавливать).

Когда найдёте ту систему, которую описывает софт, можно судить о том, насколько авторам программы удалось реализовать их замысел — насколько софт помогает что-то делать с описываемой в его данных системой.

Так, софт покупок в магазине описывает реальные физические продукты, находящиеся на складе в магазине и потенциально приезжающие к вам, приезжающие в физическом мире. Софт влияет на эти продукты.

Если софт не влияет на состояние физического мира, то к нему появляются вопросы. Сама по себе компьютерная программа не нужна, она нужна только для того, чтобы изменять физический мир при помощи заключённых в ней описаний. **Не обольщайтесь физичностью происходящего в типографии, физичностью происходящего в датацентре, физичностью происходящего у курьера. Удерживайте внимание на физичности того, что описывается носителями информации.**

Следите за речью ваших собеседников: если кто-то сказал, что он пишет отчёт и его задача — «отдать отчёт», то проверьте: он и впрямь считает, что описал не работу курьера, который «отдал пакет с бумагами»? Вероятно, речь идёт не об «отдаче отчёта», а об «использовании результатов отчёта» для чего-то, что меняет физический мир. И если этого в разговоре нет, то ваш собеседник или врёт, или лукавит, или существенно заблуждается, или очень плохо выражает суть его работы — и вам нужно задавать дополнительные вопросы, чтобы разобраться в том, что там происходит с его работой.

Для этого нужно обладать кругозором и прикладным мастерством. Если вы изготавливаете ERP-систему (enterprise resource planning, планирование ресурсов предприятия), то вы должны понимать: ERP-система документирует и позволяет удобно просматривать самые разные описания потока ресурсов (ресурсы — это сырьё, полуфабрикаты разной степени обработки и сборки, конечная продукция, «застрявшие» в предприятии, которое что-то выпускает — всё это физические объекты, вещи, занимающие место в пространстве-времени). ERP-система существует как информационная модель, современная «ручка-бумажка», внешняя память для мозгов сотрудников предприятия, она просто отражает состояние этого потока ресурсов, причём модуль планирования ещё отразит в своих выходных данных и будущее состояние потока ресурсов. Это ОК, мир четырёхмерный и будущая физичность нас тоже вполне устраивает.

Единственная цель ERP-системы — учёт ресурсов предприятия с целью планирования максимального прохода/throughput этих ресурсов по предприятию. Чем больше пройдёт через предприятие сырья (реального физического, а не описаний сырья!) и выйдет готовой продукции (реальной физической!), тем больше будет встречный денежный поток (реальные деньги, думайте о них как о кучках золота). Если не брать в расчёт **решений по физическому изменению потока ресурсов** (заказ новых партий сырья, загрузки того или иного оборудования, задействования тех или иных складов, отгрузки по тем или иным контрактам), то

ERP-программу со всеми её базами данных можно бы не делать, физический мир не изменился бы.

Фотография/проект-design/документированное описание и любой другой «информационный объект» как «носитель информации с информацией» — они всегда описывают какую-то систему. ERP-программа делает «фотографию» потока ресурсов предприятия как воплощения системы. Воплощение системы, которое нас интересует — это не воплощение самой ERP-системы как воплощение «фотографии». Оно, скорее всего, банально, как воплощение книги — все они более-менее похожи, пара сотен склеенных страниц с частицами краски на них. Разница только в содержании книг, так и с софтом — воплощения софта похожи, разница только в содержании софта, что отражает софт в физическом мире, что он описывает! **Воплощение системы в случае ERP-софта — это, скорее всего, не сам этот ERP-софт, а поток ресурсов предприятия, физический объект. Если программист ERP-софта считает, что его задача не изменения в потоке ресурсов предприятия, а собственно сам ERP-софт (а в самых тяжёлых случаях этой ошибки — исходный код программ ERP-софта), то проект в беде. Проверяйте, понимают ли разработчики софта, что они меняют своим софтом в окружающем мире — и проверяют ли они, меняется ли это «что» в мире, а не только проверяют ли они свой софт! Да, к вам самим это тоже относится!**

Если вы ничего не знаете про физический объект, который нужно менять вашими описаниями, вы будете безуспешны, никакая системность мышления вам не поможет. В какой-то неуловимый момент ваша работа с описаниями станет совсем уж фантазийной, оторвётся от реальности. Описания не будут соответствовать ни прошлым состояниям мира, ни текущим, ни будущим. Это будут сказочные, утопические описания. В инженерных проектах это недопустимо, на воплощение утопий и сказок нельзя тратить ресурсы!

Системное мышление помогает тем, кто знает свою предметную область, и это знание дотягивается до физического мира. А кто не знает свою предметную область и не дотягивается мыслью до изменений в физическом мире, им уже ничего не поможет. Они будут описывать странное, реализовывать случайные изменения в физическом мире, или вообще не затрагивать физический мир своими фантазиями.

Конечно, выдача фантазий вместо инженерной работы не означает того, что они не будут оплачены. Котята мало что меняют в физическом мире, но всегда накормлены и ухожены. Котятам достаточно менять состояние людей, поднимать им настроение просто своим видом, в современном мире им даже не надо ловить мышей. Люди (и даже роботы, даже AI-агенты), если они не кусаются, тоже будут всегда накормлены, независимо от результата их работы, даже если результаты работы оторваны от реальности, ничего в мире не меняют, просто кому-то поднимут настроение. Но иногда такие утопические работы приносят вред, даже если сами эти работы хорошо оплачиваются. Вред будет хотя бы от того, что будут потрачены впустую ресурсы, которые могли бы быть потрачены на что-то более полезное: не просто поднять кому-то настроение, но вылечить, накормить, согреть. В конечном итоге, если люди сидят голодные, холодные, больные и только и делают, что поднимают друг другу настроение, всё будет плохо. Так что избегайте отрыва от жизни, отрыва от физического мира, отрыва от реальности!

Это не означает, конечно, что в богатом мире довольно много людей не может перейти на предоставление «персональных сервисов» в так называемой «маникюрной экономике»[128]. В руководстве по методологии это будет подробней рассказано в разделе теории стратегирования. Но богатый мир — это когда какие-то люди с их инструментами достаточно произвели жилья, еды, медицинского оборудования и лекарств, средств связи и ещё много чего другого. Голодные люди потратят деньги на еду, на связь, а не на маникюр или коучинг. Голодные люди будут работать, а не говорить «я выгорел». Поэтому в конечном итоге смотрите, что в мире вы собираетесь менять своим проектом, какие ваши амбиции по сравнению с котёнком и его амбициями.

Как связаны понятия «объект в реальном мире» и «воплощение системы»? «Система» — это тип для объектов в реальном/физическом мире. Эти объекты характеризуются тем, что они как-то могут быть выделены из мира (граница системы, причём это граница в пространстве-времени — можно говорить о форме объекта в четырёхмерном мире). Всё, что снаружи от границы — среда/окружение/environment этой системы. Система и воплощение системы — это синонимы. Конечно, внутри системы можно выделить другие системы, они будут называться подсистемами. Конечно, можно выделить границей и какую-то систему, в котором внутри будет находиться система — она будет называться надсистема. Мы будем постепенно раскрывать понятие «система», добавляя и добавляя разные связи между ним и другими понятиями.

Если мы берём какой-то физический объект, то обычно у него уже есть тип из предметной области (тип из мета-модели/middle-ontology), который берётся из учебника предметной области/domain. Например, борт #45667 классифицируется в предметной области как «самолёт», записываем «борт #45667»::самолёт. Само понятие «борт #45667» — это будет понятие модели, соответствующее экземпляру «самолёта» в физическом мире ((вы же не путаете строчку символов ««борт #45667»::самолёт» с реальным физическим самолётом? Строчки не могут летать!). Мы можем записать «самолёт» как тип, а в системном мышлении дополнительно можем записать «самолёт::система», добавив к типу «самолёт» из мета-модели предметной области авиации тип «система» из системного мышления как мета-мета-модели. Значком **::** мы будем показывать тип, который был использован в рассуждении (яблоко::фрукт), то есть показывать **отношение классификации** (принадлежности к множеству). Как это делать, разъясняется в руководстве по рациональной работе, но в системном мышлении мы добавляем, что «система» и другие понятия системного мышления — это и есть типы, которые вы будете присваивать чаще всего. Мир вы будете видеть как набор не просто объектов, но объектов::система (объектов типа «система»).

Можем записать или даже сказать просто «самолёт», без указания типа — при этом чётко отдавать себе отчёт, что это мы просто опустили понятие типа «система» из мета-мета-модели. Так и происходит связь системного мышления и жизни, обеспечивается адекватность его приложения к объектам из рабочих проектов: мы удерживаем в голове (не обязательно даже явно указывать!) присваиваемые объектам предметной области (определяются типами мета-модели из учебника предметной области) типы трансдисциплин интеллект-стека фундаментальных методов мышления (физики, семантики, теории понятий, онтологии и т.д.) — всё,

[128] https://www.ft.com/content/f3cc3767-b0c3-4dd1-983a-6f158799b6c4

что нужно для системного мышления, в данном случае понятие «воплощение системы» как присваиваемый тип. А сами объекты, которым присваиваются типы — они будут в физическом мире, «в жизни».

Возьмём Китай. В системном мышлении это будет «Китай::система», в учебнике какой-то предметной области (например, геополитики) — «Китай::страна». Мы можем записать «Китай::страна::воплощение системы», чтобы показать полную цепочку мета-моделирования. Но можем записать и просто «Китай», понимая, что это «страна» и понимая, что это «система» (удерживая эти типы в голове! Необязательно писать! Но обязательно понимать, какого типа был упомянут «Китай»!). **Важно, что вы всегда «на автомате» должны удерживать во внимании три согласованных между собой объекта:**

- **объект-из-жизни (иногда его ещё нужно найти!), то есть какую-то область пространства-времени в физическом мире, («яблоко в жизни»). Это модель, описание реального объекта.**
- **тип объекта из дисциплины/теории метода работы в этой предметной области («пища», или «счётный объект» — если яблоки используются вместо счётных палочек, например, в детском саду), это мета-модель, модель предметной области, которую узнаёте из материалов каких-то прикладных курсах и прикладного/профессионального развития.**
- **тип объекта из фундаментальных трансдисциплин (например, «система», «физическое тело») методов мышления интеллект-стека. Это мета-мета-модель, изучается при получении образования, в ходе рабочего развития (в том числе освоения наставлений наших руководств).**

И на выходе у вас «яблоко из жизни», которое может быть или пищей, или счётным объектом (пищу едят, счётные объекты в детском саду считают), но ещё вы можете присвоить яблоку понятия типов «система» (и, например, обсудить, как выращивать яблоки) и «физическое тело» (если надо определить, по какой траектории полетит яблоко, если его бросить).

Эта работа с типами — общий принцип для всего мышления, а не только системного. Вы смотрите на мир, и уже много об этом мире знаете — это вам даёт интеллект, который снабжает вас набором типов из дисциплин методов мышления интеллект-стека. **Эти типы сосредотачивают внимание на важном, даже если вы впервые попали в какую-то ситуацию. Вы не можете сказать «я понятия не имею, что я вижу». Вы, если знакомы с системным мышлением, можете сказать «я вижу системы» — и из этого уже многое следует. Скажем, системы имеют свою функцию — какое-то назначение во внешнем мире. И вы сразу уже можете интересоваться функцией тех объектов, которые вы видите, а не тратить время на определение того, чем же там надо было бы поинтересоваться по поводу полностью незнакомых объектов.**

Ищите всегда воплощение системы первым делом, системное мышление — прежде всего про физический мир и изменения в нём, хотя и имеет дело с описаниями. Но описания ничто, а вот описанное ими — всё. **Все описания делаются только ради того, чтобы менять описанное в физическом мире, их может быть много, но сами по себе они никому не нужны, нужны только в связи с тем,**

что они в конечном итоге описывают какую-то систему в физическом мире!

Моделирование: присвоение типа системе по её описаниям

Восстановите, какие системы были описаны в приведённых документах/информационных моделях (впишите имена их типов в предметной области).

Документ с описанием системы	Описываемая документом система
Табличка с названием, временем работы и адресом на дверях магазина	
Софт для личного планирования, содержащий список дел (to do list)	
Заводной глобус из магазина школьных учебных пособий	
Инструкция к заводному глобусу	
Ноты мелодии «Катюша» для бас-гитары	

Как договориться: не обобщать, а конкретизировать

Если один человек упомянул президента США, а другой — Дональда Трампа, то они имели в виду одно и то же лицо? А если другие люди упомянули президента США и Джорджа Вашингтона — они имели в виду то же самое лицо, что и человек, упомянувший президента США в предыдущем предложении?

В инженерии нужна жёсткая логика для подобных рассуждений о том, относятся ли разные описания одному и тому же объекту. Описанный одним человеком насос Р-101 на схеме трубопроводов, и описанный другим человеком насос модели ПДР-15-НШ-12 в монтажной спецификации — это один и тот же насос? А установленный в турбинном зале насос ПДР-15-НШ-12 с серийным номером RKS456/4 — как он соотносится с первыми двумя? Как описать это «в компьютере» так, чтобы и самому не запутаться, и других не запутать?

Ещё Декарт (1596-1650) задавался вопросом: а как вообще понять, что люди говорят об одном и том же объекте, если они видят в нём самые разные свойства (то есть относят его к самым разным классам)? Скажем, один инженер говорит о высокопроизводительной системе, другой — о взрывоопасной, менеджер — о прибыльной, а финансист — о дешёвой? Как тут понять, что речь идёт об одной системе? Ответ Декарта на такие вопросы используют до сих пор[129]: если **места в физическом пространстве-времени/экстенты** у двух объектов совпадают (совпадают формы объектов и их местоположение в пространстве-времени), то это один и тот же объект. Не важно, какие основные или вторичные характеристики, свойства и части увидели разные люди в объекте/системе, или для каких применений этот объект/система им нужна. Не важно, одинаковые или разные имена у тех мест в пространстве-времени, о которых говорят разные люди с разными предметами интереса в той или иной ситуации. Если речь идёт об одном и

[129] Подробней эта история и многие другие положения этого раздела рассказаны в книге Chris Partridge «Business Objects: Re-Engineering for Re-Use», https://disk.yandex.ru/i/2SgjvILB3PqJEZ

том же месте пространства-времени, значит речь идёт о том же самом воплощении системы. Если я говорю о пище, вы говорите о яблоке, она говорит о товаре, он говорит о зелёном физическом теле массой 150 грамм, и все мы показываем на одно и то же место/объем/форму/экстент в пространстве-времени/физическом мире, то речь идёт об одном и том же объекте. Если кто-то показывает в физическом мире/пространстве-времени на бабочку с крыльями и говорит «бабочка», а кто-то другой показывает на изменяющийся во времени и перемещающийся в пространстве объект яйцо-гусеницу-куколку-бабочку-с-крыльями и говорит «бабочка», то у этих двоих есть шанс понять друг друга, даже если они поначалу не могли договориться из-за путаницы в словах/терминах.

Важно, что «ход на понимание» тут на конкретику (воплощение системы, физический мир), а не на «определение» (то есть отнесение рода объектов к их виду — «определение по Аристотелю», «как в словарях», как в учебниках математики). **Выдача определений, равно как и требование определений, обычно затуманивают понимание в сложных ситуациях. Проясняют понимание только примеры воплощений каких-то систем в физическом мире, «заземления»/grounding — все споры о терминах прекращают именно такие примеры.**

Если тут в руководстве по системному мышлению выдать глоссарий, а в нём выдать набор определений, то понимание материала ухудшится: формальность изложения поднимется, а связь с физическим миром станет хуже. В нашем руководстве в его последнем разделе «Вместо глоссария: онтология системного подхода третьего поколения» мы предложили другой способ решения проблемы компактного описания множества связанных понятий системного подхода. Если очень хочется иметь глоссарий, попробуйте вместо чтения этого глоссария почитать этот раздел, причём обратите внимание: он разворачивается в довольно большой текст, если обратиться к прописанной там литературе (десятки довольно объёмных текстов первоисточников).

Если не требовать, чтобы все рассуждения, все описания систем, которые делают люди, в конечном счёте привязывались бы к воплощениям систем, то мы не имели бы возможность проверить, об одном и том же говорят люди, или о разном. Более того, были бы огромные проблемы с проверкой того, говорят ли люди о реальном мире или высказывают благие пожелания, или просто фантазируют, или даже сознательно не хотят доводить свои мысли до реальности. Упор на то, что описание и документирование системы происходит по поводу воплощения системы в физическом мире (а не по поводу чего-то абстрактного, допускающего фантазии) позволяет до некоторой степени игнорировать различия в используемой людьми терминологии. В конечном итоге всегда можно проверить, одно и то же понятие люди обозначают разными терминами, или разные: даже если речь идёт об абстрактных понятиях, всегда можно указать примеры из реального мира, конкретизирующие эти абстрактные понятия.

Это не означает отказа от описаний: описания нужны, чтобы переносить знания про одни места в физическом мире (воплощения системы) на множество таких мест (на классы систем). Описания систем поддерживают абстрактное мышление, но **в конечном итоге всё решает деятельность по изменению мира, то есть деятельность по воплощению/изготовлению/реализации системы в реальном/физическом мире.** Описание помогает удержать внимание на важных соображениях по поводу физического мира, но не более того. Физический мир

главный, системное мышление удерживает на нём внимание, для этого в системном мышлении используется понятие воплощения системы.

Моделирование: заземление абстрактных текстов

Напишите граундинг-эквиваленты/заземления для приведённых текстов, не меньше двух разных. Пример приведён в первой строчке таблицы.

Номер п/п	Абстрактный текст	Заземление
Пример	Воплощение системы должно соответствовать стандартам.	Компьютер должен обеспечивать подключение внешних устройств по стандарту USB 3.2. Мороженое «Пломбир» должно соответствовать ГОСТ 31457-2012
1.	Документация должна быть компактной.	
2.	Наше устройство прочное.	
3.	Объект внимания, который даётся в методе системного мышления.	
4.	Пришло сообщение.	

Отношение состава

Главные отношения в системах (воплощениях систем, помним о физичности!) — это отношения «часть-целое» (part of), они же **отношения состава/сборки** (composition). Инженеры часто говорят об этом как о разбиении (breakdown) системы. В системах главное — это системные разбиения/breakdowns. **Уровни этого разбиения — это и есть системные уровни. Разбиения делаются вниманием, ничего в физическом мире специально не разбивается! Если вы разбили бабочку на крылья, тушку, ножки и усики, то это не означает, что вы оторвали от тушки крылья, ножки и усики: нет, вы просто выделили в живой бабочке эти части своим вниманием!** Более того, вы и саму бабочку выделили вниманием как часть её окружения!

Основной ход в разбиениях — это от части к целому, от системы к окружению. Часть хорошо видна, отлично выделяется, а вот целое в окружении часто приходится поискать. Системное мышление называют поэтому холистичным/ориентированным на целое/holistic. Позже мы узнаем, что это не совсем так — системное мышление ориентировано не только на целое, но и на части, так что оно и редукционистское (сводящее к частям) тоже, а чистый холизм и чистый редукционизм друг без друга — зло, заблуждение, ошибка, тупик. Но всё-таки первая мысль системного мыслителя — в сторону нахождения целого, когда часть известна (то есть любая система вызывает первую мысль в нахождении надсистемы, наружу от границы системы, а не внутрь к подсистемам).

Речь идёт об «истинных частях», то есть физических частях, а не бытовом разговоре про «часть описания системы», «часть характеристик системы» и прочих «частей информационных объектов». Мысли о том, как именно распространить системное мышление на такие «информационные части» сейчас появляются, мы немного этого

коснёмся, но потом. Пока же говорим только о пространственно-временных частях, то есть частях физического мира, частях каких-то систем. Системы, напомним, физичны/реальны.

Крыло и фюзеляж — части самолёта, топливный насос — часть двигателя. Крыло (все молекулы крыла) занимают часть всего объёма самолёта, то есть часть занимаемого им места в физическом мире/пространстве-времени, топливный насос занимает часть двигателя. Все молекулы топливного насоса являются частью молекул двигателя — молекулы же определяются как такие маленькие места в физическом мире. Если речь пойдёт о каких-нибудь нанопокрытиях толщиной в пару молекул, можно повторить рассуждение, перейдя к каким-нибудь кваркам — нюансы с квантовой неопределённостью тут неважны, важен принцип рассуждения про части материального мира.

Если принять, что все системы существуют не просто в физическом пространстве, но в пространстве-времени, то весь разговор о разных состояниях системы или её разных ролях превращается в разговор о частях во времени. Например, яйцо является просто частью бабочки во времени — пока бабочка проходит стадию «яйцо», никакой другой «бабочки» в мире, которая занимает место яйца в физическом мире, нет.

Тем самым с состояниями системы или её ролями (те состояния/периоды времени, когда система выполняет какую-то функцию для её надсистемы, то есть отыгрывает свою роль) можно работать как с отдельными объектами, они могут получать отдельные имена. Бабочка на стадии «яйцо» называется «яйцо». Пётр Сидорович в состоянии болезни называется «пациент». И «пациент» тут просто роль/состояние Петра Сидоровича. Микроскоп, молоток или камень в роли «забивало» для гвоздя — в тот момент, когда они для этого используются (они аффордансы, выполняющие какие-то функции).

Удобно представлять воплощения системы эдакими «червяками» во времени, в которых их место в физическом мире проходит какую-то траекторию во времени/«развёртку во времени»:

При таком подходе **простые/мгновенные события** — это трёхмерные «срезы» системы на какой-то момент времени, эдакие трёхмерные «системы, какие они в моменты остановки их работы», «без времени». До события было одно состояние/роль системы, а после мгновенного события — другое состояние/роль. **Кроме того, сама система появляется в какой-то момент времени, а в**

какой-то момент времени она исчезает, это события появления и исчезновения системы. Спортсменка на фотографии проходит разные события (отрыв от земли, приземление), определяемые её позами в эти моменты времени. Эти позы, как «трёхмерные фотографии» и есть события, разделяющие разные состояния «сальто», «подготовки к сальто», «выравнивание после приземления».

Например, в позном беге[130]событием является «поза бега» — всё тело бегуна в определённый момент времени. «Поза бега» является ключевой для правильного бега, весь бег оказывается основан на событии принятия правильной позы.

Описания очень сложных систем (от микропроцессоров с их диаграммами состояний до предприятий с их регламентами работы) часто опираются на связанную с ними событийную структуру. **Проще всего выяснить что-то про незнакомую систему — это спросить, какие с ней связаны события, то есть какие объекты меняют свои состояния с каких на какие в какие моменты.** В разработке информационных систем предприятия этот метод называется «событийный штурм»[131] (event storming, по аналогии с «мозговым штурмом»). Очень удобно строить онтологию/мета-модель предметной области, задавая вопрос о том, что там меняется в ходе работы, «какие события у вас происходят, какие заметные моменты» — вот моменты изменений и будут тамошними событиями, а дальше вы интересуетесь объектами, которые поменяли одно состояние на другое, и тем самым находите эти объекты. Скажем, всех волнует момент, когда, наконец, прекращается сквозняк. ОК, что в этот момент происходит? Закрывают дверь! ОК, дверь меняет состояние с «дверь открыта» на «дверь закрыта», речь идёт о событии «закрытие двери».

Когда говорят о событиях, то рассматривают их в контексте таких больших отрезков времени, на которых длительностью события можно пренебречь. Если мгновенное/простое событие рассмотреть «в лупу», то мы найдём там **сложное событие,** которое занимает некоторое ненулевое время. Так, между двумя часами «дверь открыта» (и скозняк в комнате, состояние комнаты «комната проветривается») и сутками «дверь закрыта» (перед очередным проветриванием) происходит событие «закрытие двери». Временем этого события можно пренебречь. Но если не пренебрегать, то найдём 30 секунд, когда агент (человек или робот) подходит к двери и выполняет работу по какому-то методу закрытия двери (предложите три разных метода/способа/практики/стиля/культуры выполнения

[130] https://sportpriority.com/blog/gid-novichka/metod-poznogo-bega-chto-eto-i-zachem-on-nuzhen/
[131] https://en.wikipedia.org/wiki/Event_Storming

этой работы). Эта работа переводит дверь из состояния «дверь открыта» в состояние «дверь закрыта». Проект — это выполнение работ по каким-то методам, результатом работ будет перевод каких-то объектов из одного состояния в другое.

Первым делом нас будет интересовать то, как меняют свои состояния объекты в физическом мире — меняют состояние системы. Так, говоря о созревании помидоров, можно выделить сам помидор как целое, и три его **полных части во времени/состояния/темпоральные части** — зелёный помидор, событие покраснения (превращения зелёного помидора в красный) и красный помидор. В большинстве случаев при разговоре про помидор можно пренебречь временем события покраснения помидора и всеми промежуточными при этом состояниями, оно тут просто не принимается в расчёт: предметом интереса у нас будут зелёное и красное состояния помидора, то есть объекты «зелёный помидор» и «красный помидор», а вот «промежуточный помидор» нам не интересен, поэтому мы считаем это просто мгновенным событием.

Вот **диаграмма пространства-времени** (space-time map) из книги Chris Partridge «Business Objects: Re-Engineering for Re-Use»[132], которая это иллюстрирует:

Figure 8.25: Complex event of the tomato changing colour space-time map

Все три измерения пространства на этой диаграмме показывают на одной оси, а время на другой оси. Помидор (экземпляр помидора #91, речь ведь идёт об объектах-индивидах в физическом мире, а не типах) занимает какое-то пространство-время (тип не может занимать место в пространстве-времени, только экземпляры из физического мира могут занимать пространство-время), а внутри него находятся индивиды-состояния зелёного помидора, красного помидора и сложное событие изменения цвета помидора.

Событие «вторая мировая война» тоже длилось много лет, но при рассмотрении «предвоенного мира» и «послевоенного мира» это событие войны считается прошедшим «мгновенно», простым.

В упомянутой книге можно прочесть много любопытного про подобный 4D (4 dimensions, 4 измерения, три пространственных и одно времени) подход к описанию мира, в котором **огромное множество самых разных отношений между объектами сводятся к ключевому для системного мышления отношению части-целого в физическом мире**. «Состояние», «событие», «роль» — все они оказываются просто частями системы, что существенно облегчает мышление, делает его универсальным и компактным. **Системное мышление — это прежде всего мышление про части и целое, отношение**

[132] https://disk.yandex.ru/i/2SgjvILB3PqJEZ (занимающимся моделированием данных в части создания корпоративных информационных систем знание этой книжки обязательно).

состава/композиции/разбиения/is_part_of. Самое тут важное, что при таком подходе резко упрощается рассмотрение самых разных других видов отношений: описание мира в таком подходе становится короче, рассуждения о самых разных явлениях унифицируются. И этот подход хорошо формализуется, что важно для инженерных применений[133].

Что тут можно спутать при рассмотрении истинного/физического отношения «часть-целое»? Удивительно, но естественный язык даёт богатый материал для путаницы.

Например, можно сказать, что шероховатость — это часть богатого опыта восприятия стола. Тут сразу несколько засад: шероховатость — это не физический объект. Богатый опыт восприятия — это не физический объект. Поэтому все молекулы шероховатости не будут входить в число молекул опыта восприятия. Но слово «часть» было использовано, и оно не режет слух! Вот и источник путаницы: **невнимательные (нетренированные в системном мышлении) люди не удерживают отношение часть-целое в материальном мире на несколько уровней. Они путают отношение «часть-целое» с отношением классификации.** Тигры — это часть зверей. Слово «часть» тут не обозначает того, что все молекулы тигров как объект входят во все молекулы зверей, как объекта! Нет, разговор идёт об абстрактных объектах/математических множествах/классах.

Появление абстрактных/идеальных «частей систем» нужно отслеживать и устранять, это ошибки в системном мышлении. Части — это всегда физические части, физические объекты, выделяемые вниманием из целого-окружения этих частей. У автора руководства есть «неоскудевающая рука» и «бесконечная доброта» к инженерам-менеджерам. Нельзя сказать, что это части его, как системы! Если ещё можно сказать, что все молекулы руки автора входят в число молекул автора, то уж про молекулы доброты, которые входят в молекулы автора, как часть этих молекул — этого нельзя сказать. Не путайтесь!

Отдельный источник путаницы — это **сначала перепутать воплощение системы и её описание, а потом говорить о частях описания как о части системы.** Так, усы на моей фотографии можно признать моей частью. Но это не так: все молекулы краски, из которых состоят усы на моей фотографии не входят в число молекул, из которых составлен я в целом! Это путаница между усами и документацией усов, усами и описанием усов! Документация моста — это не часть железнодорожного моста! Молекулы документации (будь это информационные модели на серверах в датацентре, или по старинке пара грузовиков мёртвых деревьев с частицами краски) не входят в число молекул моста. Мост и его части (опоры, дорожное полотно и т.д.) — это одно, а документация на мост — это не часть моста, это просто документация на мост. Документация не входит в состав моста, не взаимодействует с другими частями моста как физический объект, не является истинной частью конструкции моста!

Проверяйте себя, можете ли вы удержаться от бытового использования слов «часть» и «целое», ограничиться для отношения composition только физическим миром, удерживать физичность разбиения на несколько уровней вверх и вниз от той системы, которая находится в центре вашего внимания. Если вы разбили своё описание на три части, то это не системное разбиение! Это просто вы разбили описание на три каких-то части. Эти части ещё

---

[133] https://ailev.livejournal.com/1653296.html

и не взаимодействуют друг с другом, описания ведь пассивны, они не в физическом мире. Это не про воплощение систем, не про системные разбиения, это про другое.

Современные исследования в физике обсуждают, что границы физических объектов и их частей не случайны, они каким-то образом оказываются стабильными: камень как объект в мире лежит в пустыне тысячи лет перед тем, как окончательно превратиться в песок, но всё-таки его форма и размер в физическом мире более-менее стабильны. Это свойство поддерживать свою форму и удерживать состояние в каких-то приемлемых для устойчивого существования параметрах начинается с организационного/эволюционного уровня «вещество» (камень в пустыне, капля масла в воде — если их потыкать чем-то не слишком крепким, то они сохранятся), но по мере усложнения уровня организации до более-менее живых объектов, удержание формы в борьбе с энтропией как физически неизбежным разрушением оказывается всё более и более удивительным. Если потыкать даже твёрдым предметом живое существо и поранить его, то с большой вероятностью небольшая рана заживёт. Если вы попробуете потыкать твёрдым предметом разумное существо, то вам уже может и не удастся этого сделать. Если потыкать даже большим твёрдым предметом какой-нибудь завод, то он будет восстановлен в удивительно короткие сроки!

Даже граница системы становится из «мыслимой»/абстрактной/идеальной какой-то вполне материальной, тоже занимающей объём: мембраны клеток, корпуса кораблей, стены домов — они тоже в ходе усложнения организации жизни (и технологической жизни), по мере эволюции (и техно-эволюции) становятся полноценными системами! Подробней о том, что границы систем в физическом мире проходят не совсем случайным образом, а отражают фундаментальные законы природы, определяющие стабильность систем перед лицом разрушительного времени (физики тут упоминают неизбежный рост энтропии, преодолеть который может только жизнь, обучающаяся его преодолевать в ходе эволюции), можно почитать в работах Fields, Glazebrook, Levin и других авторов[134].

Вопросы. Воплощение и описание систем

В какой момент появилась система «Ваза»?

> --- В момент создания эскиза художника ### Нет, это появилось описание вазы. Ваза ещё не занимает какое-то физическое пространство-время в мире. Карта — это не территория, эскиз вазы — не ваза.

+++ По завершении формовки на гончарном круге ### Да, ваза заняла какое-то пространство-время в мире. Далее все технологические операции пойдут с этой вазой.

---

[134] 2021, C.Fields et. al, "A free energy principle for generic quantum systems", https://arxiv.org/abs/2112.15242, 2020, Fields, Glazenbrook, "Information flow in context-dependent hierarchical Bayesian inference" https://chrisfieldsresearch.com/contextual-pre.pdf, 2018, Fields, Glazenbrook, "A mosaic of Chu spaces and Channel Theory with applications to Object Identification and Mereological Complexity", 2018, https://arxiv.org/abs/1803.08874.

--- По завершении росписи ### Нет, в этот момент ваза уже есть — у неё уже есть какое-то место в физическом мире, и не говорится, что «ваза должна быть расписная и обожжённая, иначе это не ваза». Ваза вообще может быть стеклянной.

--- По завершении обжига ### Нет, обжигается уже ваза — она уже имеет форму, присутствует в мире, в ходе обжига она просто приобретает свойство водоупорности. И ваза вообще может быть металлической.

--- В момент установки в зале дворца ### Нет, к этому моменту ваза уже занимает какое-то пространство в мире, и речь идёт просто об изменении её положения в пространстве в момент установки, но не о её появлении.

Когда возникла система «Ваза с драконами»?

--- В момент создания эскиза художника ### Нет, это появилось описание вазы. Ваза ещё не занимает какое-то пространство-время в мире.

--- По завершении формовки на гончарном круге ### Нет, уже существует ваза, занимающая какое-то пространство-время в мире, но драконов на ней ещё нет.

+++ По завершении росписи ### Да, ваза с драконами появляется в этот момент — место в пространстве-времени теперь занято именно «вазой с драконами». До этого момента была просто «ваза».

--- По завершении обжига ### Нет, к этому моменту ваза с драконами уже существует (занимает какое-то место в пространстве-времени), просто меняются какие-то её свойства — она становится водонепроницаемой.

--- В момент установки в зале дворца ### Нет, к этому моменту ваза с драконами уже появилась, она просто поменяла своё местонахождение.

Когда закончилось существование системы «Ваза с драконами в парадном зале дворца»?

+++ Когда её украл дворецкий ### Да, с этого момента объект «ваза с драконами» существует, но объект «ваза с драконами в парадном зале дворца» исчез.

--- Когда украденную дворецким вазу разбила у себя дома жена дворецкого ### Нет, она разбила другой объект — «вазу с

драконами», уже украденную из дворца, но не «вазу с драконами в парадном зале дворца».

--- Когда осколки разбитой вазы вместе с прочим мусором попали в подсыпку дорожной насыпи ### Нет, в этот момент исчез объект «разбитая ваза», а «ваза с драконами в парадном зале дворца» исчезла задолго до того.

Когда возникла система «стол семьи Ивановых»?

--- В момент изготовления последней детали комплекта стола на фабриках ИКЕЯ ### Нет, стол ещё не имеет формы стола, не занимает пространства-времени как «стол». Это только «набор деталей стола».

--- В момент сбора комплекта деталей в одну упаковку в упаковочном цехе фабрики ИКЕЯ ### Нет, в пространстве-времени ещё нет «стола», есть только «упаковка с деталями стола».

--- В момент снятия упаковки со стеллажа И.И.Ивановым в магазине ИКЕЯ ### Это до сих пор не стол, это только упаковка с деталями — в пространстве-времени нет объёма/места, занимаемого этим столом.

--- В момент оплаты стола И.И.Ивановым на кассе магазина ИКЕЯ ### Это до сих пор не стол, это только упаковка с деталями (хотя это уже будет упаковка семьи Ивановых, независимо от того, где она находится).

--- В момент вноса упаковки в квартиру Ивановых ### Это до сих пор не стол, это только упаковка с деталями (хотя это уже упаковка семьи Ивановых, независимо от того, где она находится).

+++ В момент окончания сбора в квартире Ивановых стола из деталей, находящихся в упаковке ### Да, в пространстве-времени появилось место, соответствующее объекту «стол» — и этот стол семьи Ивановых. Даже если бы этот стол собрали на улице, или прямо в магазине — момент появления стола семьи Ивановых именно этот.

--- В момент установки стола на место на кухне Ивановых ### Нет, к этому моменту стол уже существует и принадлежит семье Ивановых — он занимает какое-то место в пространстве-времени, и перемещение его в пространстве ничего тут не меняет.

Что из перечисленного является частью четырехмерного объекта «стол»?

--- Инструкция по сборке. ### Нет, это описание стола.

+++ Этикетка, приклеенная снизу. ### Да, это составная физическая часть стола.

--- Чертежи стола, находящиеся на заводе-изготовителе. ### Нет, это описания стола.

+++ Надпись шариковой ручкой на столешнице стоящего в гостиной стола. ### Да, частицы пасты шариковой ручки в надписи являются физической частью стола — они часть того места в пространстве-времени, которое занимает стол. Содержание надписи не имеет значения, имеют значение стол и краска как информационные носители, вопрос про носитель информации как часть стола.

--- Отклеившаяся от стола этикетка, которую выкинули в мусор. ### Нет, отклеившаяся от стола этикетка больше не является физической частью стола — она не будет частью того места в пространстве-времени, которое занимает стол.

Выберите из списка ниже названия физических объектов:

--- Атомный реактор ВВЭР (водо-водяной энергетический реактор) ### Нет, речь тут идёт о классе/типе объектов: он не занимает какого-то конкретного места в пространстве и времени, так что он не физичен — физичны только конкретные экземпляры атомных реакторов ВВЭР.

+++ Атомный реактор первого блока Калининской АЭС ### Да, мы можем указать на конкретное пространство и время, которое занимает в мире этот экземпляр реактора.

--- Техническая документация атомного реактора первого блока Калининской АЭС ### Тут речь идёт обо всех комплектах документации (не конкретный комплект), то есть это класс таких комплектов — нельзя указать место в пространстве-времени, где находится «документация» (пространство-время может занимать только конкретный носитель информации, «медиа»: конкретные экземпляры листов бумаги, ячейки флеш-памяти, домены магнитного диска). Это класс описаний конкретного физического объекта-реактора как выделенного места в пространстве и времени, но это класс описаний — не физический объект.

+++ Первый комплект технической документации первого блока Калининской АЭС ### Да, речь идёт о конкретном пространстве и времени, где находится первый комплект (т.е. экземпляр, физический объект) документации. Тут речь идёт главным

образом о носителе информации, а не о том, какая информация на этом носителе, не об описании первого блока Калининской АЭС.

Продуктом компании является макетная плата, поставляемая в коробке с набором документации. Что из нижеперечисленного относится к воплощению системы?

--- Принципиальная электрическая схема платы ### Нет, это описание из какого-то документа. Описание — это не воплощение!

--- Схема разводки платы ### Нет, это описание из какого-то документа, описание не может быть воплощением.

--- Руководство по использованию платы ### Нет, это описание макетной платы (даже если оно документировано, оно не может быть воплощением).

--- Описание портов платы ### Нет, это описание каких-то важных характеристик макетной платы, описание не может быть воплощением (даже если описание документировать).

+++ Ничего из приведённых других вариантов ответа ### Да, воплощением системы является по факту именно макетная плата, в списке других ответов её нет.

Отверстия

Объект «отверстие» в языке определяется как нечто несуществующее, «дырка». В бублике дырка — то место, где нет теста. Но в инженерном мире дырка/отверстие/полость вполне себе существует как отдельный конкретный объект в физическом мире: её можно сделать (просверлить), её можно облицевать каким-нибудь покрытием. Скважина — это отверстие в земле, нефтяники на сленге её часто называют «дыркой»: она ценна именно тем, что в скважине ничего нет, поэтому по ней можно качать нефть или газ. «Проходка» (например, кабельная проходка[135]) — это устройство с отверстием, через которое можно пропустить какой-то другой инженерный объект (в кабельной проходке — кабель, в кровельной проходке это может быть кабель, труба, антенна и т.д.).

Если помнить, что отверстие занимает определённый объём, определённое место в пространстве-времени, то дальше ему можно дать имя (инженеры так и делают), и обсуждать какие-то технологические операции с ним — изготовление, учёт, проверку, ремонт, «настройку». Более того, это хороший критерий для определения того, стоит ли такой «дырке» давать название: если есть какая-то операция (работа по какому-то методу, приводящая к смене состояния) с такой «дыркой», то для учёта и проверки состояния «дырки» после такой операции лучше бы у «дырки» было отдельное имя.

Отверстие/дырка/полость занимает место в пространстве-времени, следовательно, присутствует в физическом мире. «Рабочая полость» в компрессоре — это тоже «дырка», рабочее пространство в трубопроводе — это тоже «дырка». Легко о них

[135] https://ru.wikipedia.org/wiki/Кабельная_проходка

думать, представляя заполненным это пространство молекулами газа или жидкости, или даже вакуумом: физическое вещество тут не важно. Важно, как об этом думать, а думать нужно о местах физического мира, областях/объёмах пространства/времени.

Антракт — это часть (во времени) от концерта или спектакля, когда отсутствует представление. Рассуждать об антракте можно так же, как и об инженерных отверстиях, но это будет не пространственная часть, а часть во времени/темпоральная часть спектакля или концерта как материального объекта, воплощения системы. «Кульминация спектакля» тоже может быть представлена как вполне материальная часть: это темпоральная (во времени) часть спектакля, в которой достигается максимальный эмоциональный накал. С другой стороны, можно считать, что эта часть занимает незначительное время, и тогда «кульминация спектакля» будет событием — а темпоральные части будут важны до и после этого события.

Как же мыслить про антракт или кульминацию как темпоральные части спектакля? Это будут все предметы спектакля (все молекулы входящих в состав спектакля предметов — артистов, реквизита, зрителей) в какой-то отрезок времени.

Так же можно обходиться и со странными объектами, которые нужно учитывать поимённо (ибо с ними нужно вести какие-то действия), но которые трудно выделить как отдельные от физического объекта — например, сварные швы. Сварной шов нужно запроектировать, потом сделать, потом его регулярно нужно проверять. Это означает, что у экземпляра сварного шва должно быть индивидуальное имя, это конкретный физический объект, занимающий место в физическом мире. Если понимать, что сварной шов — это просто место в пространстве (и времени!), то никаких проблем в мышлении о таком объекте не появляется: это такая же часть системы как собственно труба, или шестерёнка, или отверстие. У них могут быть свои части (хотя в шве есть молекулы, а в отверстии молекул нет, но есть место в пространстве, которое могли бы занимать молекулы!), они входят как части в другие материальные объекты, у них могут быть описания.

Это в какой-то мере противоречит понятию «устойчивой к действию разрушительного времени конфигурации материи» как системы в физике. Так, если смотреть на пламя свечи, то исследователь Karl Friston замечает невозможность указать границу системы, ибо там идёт какой-то беспорядочный обмен молекулами с окружением. Но если взять инженера, то для него «пламя» вполне себе объект, состояние которого можно измерить (скажем, измерить температуру в центре пламени и на его условной границе). И это пламя инженер будет определять по его пространственной форме в окружении и необходимости выполнять какие-то операции с этой формой: у инженера будет тут всё то же самое рассуждение, что с «отверстием». И пламя для него будет вполне устойчивым, а уж сварной шов с его расплывчатой границей между собственно швом и материалом на молекулярном уровне — и подавно будет устойчив.

Можно и дальше разбираться с подобными сменами состояний и объектами, которые меняют состояние в ходе мгновенного/простого события или сложного события, занимающего некоторое время — в зависимости от рассмотрения. Так, в 3D-печати ракетных корпусов[136] можно легко представить себе корпус ракеты, состоящий и сплошного «сварного шва» — по факту речь идёт о непрерывном

[136] https://www.relativityspace.com/

наплавлении металла. Важно понимать, что **масштаб времени рассмотрения**/масштаб темпоральной части системы тоже имеет значение. В большом масштабе времени «вот корпуса не было — вот он есть», а в малом масштабе времени «мы добавили один виток наплавки на корпус, и так тысячи и тысячи раз». И важны не столько какие-то «объекты», воспринимаемые как «привычные твёрдые объекты», сколько занимаемое пространство-время в каких-то границах системы. Во времени такими границами системы (или состояний системы) являются **события**, в пространстве — **форма/место**. **События и формы/места — это границы системы или состояний системы.**

Поведение системы: процессы/изменения

Состояния системы меняются с ходом времени, и рассмотрение частей и целых материальных объектов даёт возможность говорить об изменениях — то есть обсуждать изменения/действия/**процессы**/поведения/процедуры/activities просто как наборы взаимодействующих систем/«воплощений_систем»/вещей-в-физическом-мире в качестве частей целого изменения/процесса. Процесс/поведение называется обычно глаголом или отглагольным существительным. Тут много нюансов, подробней об этом будет в руководстве по методологии, где нас будет волновать описание методов работы, то есть описание поведения систем-создателей (описание работ по каким-то методам) в сложных графах создателей и развития одних систем другими системами. Скажем, создание и развитие системы AI-агента командой инженеров компании-изготовителя, а эту команду инженеров создаёт какая-то команда менеджеров по развитию этой компании. В ходе эксплуатации как-то себя ведёт созданный инженерами AI-агент. Как-то себя ведёт в ходе своей эксплуатации (то есть в ходе создания AI-агента) команда инженеров. Как-то ведёт себя в ходе своей эксплуатации (то есть в ходе создания команды инженеров) команда менеджеров развития компании. И это явно не конец цепочки, ибо команду менеджеров тоже кто-то создавал, и это тоже система-создатель. Методы создания и развития систем и обсуждает методология, подробно создание::поведение и развитие::поведение разбираются в руководстве по методологии.

Тут же мы рассмотрим способ, каким можно думать о поведении системы в неразрывности системы и её поведения (ибо не бывает поведения в отрыве от того, что как-то себя ведёт, равно как чего-то, что ведёт себя «никак»). Думать о молотке легко, легко его представить, легко описать. Но вот как описать поведение молотка?

Поведение/behavior — это всегда изменения каких-то состояний систем (прохождение событий), а изменений бывает множество вариантов.

Наиболее значимыми для нас различимыми вариантами поведения (в случае систем-создателей говорят также о деятельности) являются:

- **функции** (для каких угодно систем, для создателей чаще говорят о методах/практиках/стилях/культурах и ещё много синонимов с нюансами). Это поведение системы как функционального объекта. Как думать о функциях/методах наставляет руководство по методологии.
- **Работы**. Это поведение систем-создателей как аффордансов/конструктивных объектов (ресурсы, которые могли бы выполнить работу) согласно их функциям, а систем-создателей как конструктивных объектов, выполняющих роль функционального объекта. Как

думать о работах наставляет руководство по рациональной работе, а ещё про операционный менеджмент (как управлять работами) рассказывает руководство по системному менеджменту.

Самый общий тип изменений, относимый к какой-то одной системе, мы будем называть **поведением** этой системы по отношению к изменяемым системам. Поведение — это взаимодействие системы с её окружением, то есть «внешнее поведение». Мы тут не рассматриваем то, как взаимоменяются/взаимоизменяются подсистемы, которые тоже как-то себя ведут, совместно давая внешнее поведение системы, в состав которой они входят. Внешнее поведение электронных часов — показ времени, для этого там внутри работают батарейка, какие-то микросхемы и экран, защищаемые от внешних воздействий корпусом часов.

Конечно, как-то ведёт себя (то есть меняется) и окружение. Скажем, агент, поглядевший на часы, меняет своё поведение. Более того, агент (например, человек), меняет своё поведение, даже чтобы поглядеть на часы (наблюдение — это активное действие, его надо запланировать, затем исполнить, хотя бы повернуть голову так, чтобы глаза увидели часы). Поведение системы само может быть ответом на внешние причины, а не «инициированным самой системой». Например, часы могут показать время после того, как на них нажали кнопку — часы реагируют на внешние воздействия.

Конечно, терминов для описания изменений в языке существует множество, нюансов самых разных поведений множество. Например, «если ничего не происходит, это формально изменение/поведение, или нет? Пробел ведь тоже символ?».

Иногда все термины, означающее поведение (глаголы, отглагольные существительные) считаются синонимичными, иногда их значение отличается и вводится какая-то специализация или даже классификация.

В текущем подразделе руководства мы не будем вводить разные типы изменений/поведения/процессов. Оставим различения сервисов, функций систем, и «методов работы»/деятельностей/«видов труда» систем-создателей на более позднее рассмотрение (разница там больше терминологическая, понятие же имеется в виду одно и то же — функция/сервис/метод работы, в его отличии от собственно работы). Заметим только, что слово «процесс» используется и как общее название для «всяких типов изменения систем» (в физике), и как синоним для методов (в process management, процессном управлении). И в этом плане использования в менеджменте «процессы»/«методы работы»::поведения отличаются от «работ»::поведения. Метод и работа — это разные виды поведения (два способа рассмотрения поведения, с акцентами на разные его аспекты):

- метод/способ как паттерн работы с (функциональными) предметами метода (инженерное содержательное рассмотрение),
- работа — задействование ресурсов-конструктивов для выполнения метода (экономическое рассмотрение).

Иногда пытаются подчеркнуть это различие процессов/методов и работ, добавляя к процессу эпитет «рабочий», и тогда «рабочий процесс» означает метод работы, а работа — это таки работа, не «рабочий процесс», другой тип. Подробней, как мы уже говорили, будет в руководстве по методологии. Просто внимательней к слову «процесс»: оно может означать разное, как, впрочем, и любые другие слова, которые люди используют для описания не только поведения, но и чего угодно.

Уточняйте, что имеется в виду, заземляйте!

Сами «изменения»::метод::поведение (точнее тут будет «изменение состояния каких-то объектов»::метод) плохо представляются людьми и AI-агентами, чтобы их детально обсуждать. Скажем, легко представить гвоздь, микроскоп, молоток и подробно обсуждать гвоздь. Но как представить «забивание::метод гвоздя::«объект, меняющий своё состояние в ходе забивания»»? Возникает в мозгу какая-то гифка/мини-фильм, где плотник забивает гвоздь. Вроде как «забивание гвоздя» это и означает. Но как это обсуждать подробно, как это рассказывать другим, как вводить типы для поведений/процессов/изменений?

С этим поможет 4D экстенсионализм, то есть представления о существовании вещей, которые занимают место/форму/экстент в 4D пространстве-времени. Подход с четырьмя измерениями даёт возможность сократить число отношений за счёт того, что огромное число отношений оказывается отношениями «часть-целое», и тогда изменения/процессы (изменяющиеся состояния какого-то объекта) становятся просто темпоральными/временны́ми частями.

«Забивание гвоздя» легко представить просто как перечисление физических объектов/систем, участвующих/взаимодействующих/взаимоизменяющих свои характеристики в период времени, соответствующий этому забиванию — т.е. «забивание гвоздя включает участие в нём гвоздя, молотка, доски, плотника». Распишем типы: “забивание гвоздя”::процесс/изменение/поведение включает::“указание на множество” “участие”::отношение “гвоздя”::система, “молотка”::система, “доски”::система, “плотника”::система.

Отношение **участия** (participation) в изменениях/действиях/процессах/процедурах/activities — это просто **специализация отношения состава** (composition, part_of/whole_to — в зависимости от направления чтения, «состоит из»/«разбито на» — тоже в зависимости от направления чтения).

Очень трудно обнаружить в жизни «процесс забивания гвоздя», сами процессы/изменения как объекты оказываются невидимы «одним кадром». Чтобы представить изменения, нужно представлять сразу «фильм»/гифку, какой-то сложный шаблон/паттерн/ритм, разворачивающийся во времени, то есть обнаружить и представить «повторы», «похожести во времени». Если перейти к темпоральным частям, то очень легко обнаружить взаимно изменяющиеся гвоздь, молоток, доску, плотника. Чуть сложней обнаружить их, если роль молотка исполняет камень или микроскоп, роль доски играет стена, роль плотника играете вы сами (и поэтому забываете включить в процесс себя), и лишь только гвоздь остаётся гвоздём — похожесть всех этих процессов «забивания гвоздя», безусловно, есть, но нужно чётко понимать, как её обсуждать. Так и обсуждать: «процессы» обсуждаются через перечисление материальных предметов, взаимоменяющихся в ходе их взаимодействия. Это обращение к перечислению «вещей»/физических объектов вместо собственно «поведения» как особого не сводимого к вещам объекта, остаётся надёжным приёмом для поиска работ, действий, методов/практик/деятельностей/труда, процессов, функций, сервисов и прочего поведения в физическом мире.

Ищите в мире не обозначаемые глаголами и отглагольными существительными действия, ищите обозначаемые существительными объекты физического мира (системы), которые как-то себя ведут во

взаимодействии, проявляют в мир своё (внешнее) поведение.

.

Во многих графических языках моделирования **стрелочки с ромбиками на конце** как раз означают отношение состава, причём целое там, где ромбик, а часть — где ромбика нет. Жёлтый «шеврон вбок» — это стилизованная стрелка, означает, что что-то меняется во времени, им обозначен «процесс». А голубые кружочки означают физические предметы, участвующие в этом процессе. Голубые кружочки — это части процесса как целого. В пределе можно считать, что «процесс» — это просто такая составная вещь, которая состоит из участвующих вещей — это если отношение участия трактовать просто как классическое отношение часть-целое.

Так, конкретное исполнение (экземпляр, процесс, работа) танцевания, выполняемого танцором в какой-то момент времени начинает существовать, а в какой-то момент времени прекращает существование — процессы не вечны, как и любые физические объекты. Если вы перечислите все объекты этого «танцевания::процесс танцора::система», то опишете танцевальный перформанс (который даже можно считать отдельной системой, которую надо будет создать, чтобы в нём рассматривать танцевание танцора). Танцевальный перформанс как система включает в себя на время его исполнения

- всех четырёхмерных танцоров, поведение которых — танцевание
- поддерживающий танцоров фрагмент четырёхмерного танцпола (паркет, на который опираются танцоры. Это важно, попробуйте прокрутиться на одной ноге два оборота на асфальте и на паркете — покрытие пола для танцев очень важно!), и ещё
- объем воздуха с колебаниями в нём, ибо в этих колебаниях — музыка для танцевания.

Мы тут сознательно отказываемся от термина «танец», поскольку он очень многозначен, имеет множество словарных значений, и поэтому путает:

- Танец в самом общем значении «поведение агента». Скажем, книжка Peter Senge называется «Танец перемен», и это просто указание на то, что всё в организациях взаимодействует-танцует и тем самым происходят перемены.
- Танец как танцевание, метод работы танцора (танцор — это роль агента, функциональный/ролевой объект). Это как раз то, что мы рассматриваем в этом разделе. Танцор танцует — это и есть танец. Слово используется как указание на метод работы агента в роли танцора.

- Танец как работа, которая выполняется конкретным агентом в конкретное время, «перформанс»/«исполнение»/«выход». Танец Васи в детском саду №6, который он будет танцевать 4 мартобря 2029 года с 14:05 до 14:07.
- Танец как вариант танцевального стиля/культуры/практики/метода танцевания в отличии этих методов/культур/практик между разными вариантами танца. Танго или вальс отличаются, и это танцы («танец танго», «танец вальс»).

Так что мы будем говорить не просто «танец», а танцевание, танцевальный перформанс, танцевальный стиль — и тем самым снимать неоднозначность. Похожая путаница будет и со многими другими поведениями/изменениями (например, «жизнь», «отдых», «эволюция»), подразумевающими какие-то изменения каких-то объектов (экземпляры и ресурсное рассмотрение работ, типы и паттерны/стили/методы/культуры/практики работ как минимум). Хотелось бы иметь эти различения, чтобы:

- мочь обсуждать разные аспекты ситуации с изменениями, для этого иметь разные понятия для разных типов поведения, чтобы обсуждать самые разные особенности самых разных изменений физического мира и изменений описаний этого мира, а также изменений описаний описаний.
- Не путать системы и их поведения. Хотя мы тут и говорим, что в рамках 4D экстенсионализма поведение системы — это сама система в ходе её изменений, которые вызываются взаимодействиями как подсистем внутри системы (внутреннее поведение), так и системы с какими-то системами в её внешнем окружении (внешнее поведение), всё-таки трудно думать о четырёхмерных системах-процессах, легко запутаться, и мы крайне не рекомендуем это делать. Думать лучше о системах и их поведениях, считая поведение отдельным типом от типа «система» и различая типы поведения для одной и той же системы.

Поведение — это не система

Плохо считать поведение и систему одним и тем же объектом — вам будет крайне трудно описывать поведение поведения: «состояния танца в ходе танца по методу танца» — запутаетесь сразу. Если вы выбрали какой-то глагол или отглагольное существительное в качестве системы («танец», «жизнь», «отдых»), то это будет порождать и многозначные толкования, и трудность описания «состояний процесса» (а не состояний системы) — и поэтому мы будем считать такой выбор ошибкой, как бы вы ни ссылались на работы по 4D экстенсионализму и не декларировали, что «системы и процессы занимают одно и то же место в пространстве-времени, поэтому это одно и то же». Формально да, одно и тоже. Но из-за особенностей языка и коммуникации вы такие процессы-системы не сможете внятно обсуждать, понятно описывать/моделировать, поэтому такие выборы «процесс-это-система» будут ошибкой.

Мы будем различать системы и поведение систем, но для уточнения границ системы и поведения систем всё-таки использовать 4D экстенсионализм, для описания поведения просто перечисляя взаимодействующие с изменением их состояний вещи/физические тела. Метод::поведение устранения сквозняка при проветривании мы опишем через изменения состояний двери: сначала дверь будет в состоянии «дверь открыта», затем состояние «дверь закрыта». Если нужно

понять, что там было, будем перечислять предметы, участвующие в изменениях состояния (нога и дверь, ключ и дверь, порыв ветра и дверь). Разные участвующие (отношение participate) в «поведении» предметы задают разные виды/классы метода/процесса/практики/стиля::поведение прекращения сквозняка путём «закрывания двери»::«класс метода прекращения сквозняка». Например, «прекращение сквозняка»::метод::поведение состоит из более мелких методов «обнаружить причину», «устранить причину, закрыть дверь», «проверить, что сквозняка больше нет», «объявить присутствующим о том, что сквозняка больше нет».

Использование терминологии «над-» и «под-» для методов/практик может быть использовано двояко:

- Как темпоральная часть ситуации задействования какого-то метода работы (тут ровно такое — и мы использовали слова «более мелкий метод» вместо слова «подметод», когда описывали приёмы/шаги закрытия двери)
- Как альтернативные классы/типы методов::тип (например, прекращение сквозняка при проветривании путём закрытия двери, против прекращения сквозняка путём закрытия окна — и там перечисляться уже будут другие предметы, в том числе не дверь, а окно). В нашем руководстве мы предпочитаем тут говорить «вид» или «класс» (отсылка к родо-видовым отношениям), например, «вид труда», «вид деятельности», «вид культуры», чтобы показать разные типы/классы, классифицируемые типом метод/труд/культура::тип. Хотя в данном случае можно говорить и об отношении специализации (метод предотвращения сквозняка специализируется до прекращения сквозняка закрытием двери и прекращения сквозняка закрытием окна).

Так что когда говорим о «подметодах»/«подпрактиках», то внимательней относитесь к тому, к какой иерархии идёт отсылка. Отношения классификации, специализации и часть-целое — это очень разные отношения, будьте внимательны.

Вот все эти рассмотрения состояний систем нужны для детального (часто — коллективного) обсуждения какого-то поведения как перехода системы из одного состояния в другое.

Так, инженер, который думает о танцевании::поведении как требующем набор предметов, существующих в их взаимоизменениях («танцевальный перформанс»), может подумать ещё и об источнике колебаний воздуха, и включить в состав перформанса плеер (смартфон в роли плеера) и аудиосистему (роль аудиосистемы могут играть или стационарные усилители и аудиоколонки, или портативная акустика, или живая рок-группа со всей своей аппаратурой).

А ещё вам придётся добавить к танцевальному перформансу зрителей (ибо танцоры любят сказать, что «если нет зрителей — то это «просто движения», а не танец» Обратите внимание, что танец::стиль или танец::перформанс, или танец::работа — непонятно, танцоры сами с огромным трудом говорят о танцах. Это одна из причин, по которой трудно учиться танцевать: вам просто плохо объясняют, что надо делать!).

Заметим, что добавка зрителей к перформансу (буквально: зрители как подсистема::«физическая часть системы» перформанса::система) — неочевидный мыслительный ход. Например, если два танцора танцуют друг с другом, то этого достаточно для танца, они «зрители» друг для друга (зрители — это роли, а не

какие-то отдельные объекты). Тем самым зрители и танцоры — участники (отношение participation, специализация отношения «часть-целое») танцевального перформанса, и тут возникает довольно богатый набор объектов для обсуждения ситуации перформанса. Мы можем рассуждать о танцевании артистов более точно, привязываясь к физическим объектам, физическому пространству-времени.

Танцевание танцора — это процесс/действие/activity, но когда мы добавляем к танцору другие объекты и это становится «танцевальным перформансом»::системой (набором подсистем, а не «процессом-танцеванием», ибо там много чего происходит в ходе перформанса, кроме собственно танцевания::поведения танцора::система), думать и о танцоре, и о перформансе с его участием можно уже примерно так же, как о «станке», «автомобиле», «отверстии». Уточнять, что такое «танцевание» (конкретизировать/заземлять/grounding) можно, давая примеры воплощений этого поведения/процесса у танцора в разных перформансах, а не обращаясь к словарным определениям термина «танцевание» или «художественный перформанс». Обращаться к определениям — это приводит к длительным и бесплодным «спорам о терминах», избегайте таких споров. Уточняйте значения через заземление.

Итак, даже о поведениях можно думать более-менее одинаково, как о целых объектах (процессах), состоящих из объектов-частей (подпроцессов), и это сильно экономит мышление.

По танцевальному перформансу (внутри которого идёт «танцевание танцоров/артистов»::поведение примерно так же как внутри автомобиля идут «работа мотора»::поведение, «вращение колёс»::поведение) тем самым, как и по любому другому материальному/физическому воплощению системы, можно условно «постучать», его можно условно «положить в тачку», на него можно условно «показать пальцем» — просто эти действия происходят с набором физических объектов, участвующих в перформансе как его части.

Условность «постучать» или «погрузить в тачку» или «показать пальцем место в пространстве-времени» заключается в том, что физические объекты в системах могут быть недоступными, слишком маленькими, слишком горячими для этих действий, но это неважно. Речь в этих метафорах идёт просто о том, что мы говорим о физическом мире, обсуждаем что-то воплощённое (занимающее конкретное пространство-время, имеющее форму и координаты в пространстве-времени), а не абстрактное/идеальное/информационное/математическое/ментальное.

Когда же мы говорим не о танцевальном перформансе в целом, а о танцевании танцора в составе этого перформанса, то стучать можно по танцору, а само танцевание — это будет изменение/процесс/поведение, а не изменяющаяся система. Изменяться будет танцор и что-то вовне танцора как системы (всегда идут взаимодействия, взаимоизменения) — состояния воздуха, несущего музыку танцору как входную информацию, состояние зрителей как результат изменений, вызываемых танцеванием. Но если захочется рассматривать все взаимоизменяющиеся объекты вместе — соберите из них новую систему, например, соберите из танцоров, помещения, музыки, плеера и колонок, а также зрителей «танцевальный перформанс»::система. Чтобы обсуждать подробно «вращение колёс» как поведение колёс в автомобиле, добавьте дорогу, добавьте двигатель, вращающий колёса, добавьте автомобиль с его весом — получите «автомобильную

поездку» примерно по той же логике, что и «танцевальный перформанс». Поездку уже можно как-то считать системой, внутри которой идут самые разные процессы. А вот по «вращению колёс» как собственно поведению постучать нельзя, если вы не добавите предметов. Будете стучать просто по колесу::системе, которая как-то себя ведёт (вращается в данном случае). Запоминаем: поведение всегда системы, а само поведение — не система, а именно поведение системы (даже не свойство, а просто поведение системы, смена состояний системы)! Указав на систему, вы указываете на систему вместе с её поведением, а вот никак не ведущей себя системы не бывает, ибо готовые системы все работают, функционируют, оказывают сервисы, влияют на окружение, изменяются внутри себя — то есть **системы всегда участвуют в процессах, всегда как-то себя ведут вовне, как-то изменяют своё окружение**.

Условность с представлением таких систем, как «поездка», «игровая сессия», «сеанс доступа» в виде набора каких-то подсистем в тех или иных состояниях ещё и в том, что трудно представить, как вы «стучите» по всем этим взаимодействующим подсистемам одновременно, да ещё и в ходе их непрерывных изменений при взаимодействии. Какой-нибудь «сеанс связи» — это взаимодействие серверов и линий связи в разных странах, и ещё физическая конфигурация там непрерывно меняется: когда рвётся какой-нибудь физический кабель, сеанс связи продолжает существовать, а взаимодействие серверов просто начинает осуществляться через другой физический кабель, или даже через лазерный луч!

И всё-таки это удобное мышление про такие «раскиданные в пространстве» системы, только нужно понимать, что все эти самые разные меняющиеся физические объекты присутствуют в физическом мире, это не описания::«ментальный объект» и не документация::«описание на физическом носителе», а также не собственно изменения/поведения/процессы. Процессы как изменения систем вполне физичны, находящиеся в процессах системы как отдельные изменяющиеся объекты просто участвуют в них (участвуют в процессах, участвуют в изменениях, «осуществляют изменения», например, «осуществляют сервис::поведение»).

Скажем, «танцевальный перформанс» (как и любое «мероприятие») мы легко назовём «системой», и ведущий там процесс/изменение будет — танцевание танцоров, а ещё провайдинг музыки (музицирование музыкантов, или игра колонок, к которым подключили плеер, причём через Bluetooth — важно всегда помнить, что есть альтернативы, которые можно рассматривать отдельно). Танцевание — нет, не назовём системой, оставим поведением танцора. Системой, осуществляющей какое-то поведение (работу, функцию, сервис, практику) танцевания останется танцор, а танцевание будет поведением (работой, сервисом, практикой, методом, способом, функцией — особенности использования разных терминов с отличающимися в системном мышлении их значениями обсудим позже) танцоров, а не перформанса. Перформанс не танцует! Он развлекает, радует, приносит деньги, что угодно, но танцуют в нём танцоры, они внутри перформанса, его часть. **Части системы обладают одними свойствами/характеристиками и поведением, а целое — другими свойствами/характеристиками и поведением.** Это появление новых свойств и нового поведения у системы по сравнению с частями называют **эмерджентностью/emergence**.

Системы всегда во время их существования (когда они готовы и используются/эксплуатируются, а не когда их создают и развивают) производят

какие-то изменения в окружающем мире (то есть как-то себя внешне ведут, участвуют в процессе) и меняются в ходе этих изменений сами. Поэтому танцор — танцует в ходе его работы, отдыхающий — отдыхает, живой агент — живёт. Как об этом говорить? Чтобы говорить точно, надо как-то собрать надсистему для изменяющихся объектов — и обсуждать поведение какой-то системы в составе её надсистемы.

Какие тут ошибки? Например, сказать, что танцор танцует в ходе/в составе танца, а не в ходе/составе танцевального перформанса. Вы не сможете внятно сказать, из чего состоит «танец» (это перформанс в целом? Только танцевание танцора, то есть смена в каком-то ритме поз танцора?), вас все будут понимать по-разному. Про «танцевальный перформанс» и танцора с его танцеванием в ходе этого перформанса вы легко договоритесь, избегая многозначности определения танца как какой-то отдельной системы. Сказать, что отдыхающий отдыхает в ходе отдыха, живущий живёт в ходе жизни — это тоже будет невнятно, ибо интуитивно непонятно, какие вещи вместе составляют «отдых» и какие вещи вместе составляют «жизнь». Поэтому ситуации отдыхания и проживания придётся тщательно специфицировать, осторожней с переводом процессов в системы по линии 4D экстенсионализма, вам придётся предъявлять надсистемы для систем, о поведении которых вы говорите, и ещё могут спросить про смену состояния смены состояний (то есть смену состояний поведения — очень непросто придётся отвечать на этот вопрос, но если система есть — будет и её поведение, если поведение назвали системой, то спросят о её поведении, «поведении поведения»).

По предприятию::система в ходе его деятельности::поведение можно «постучать». А вот по «деятельности предприятия» — вряд ли постучите, надо будет предъявить все объекты, затрагиваемые предприятием во внешнем мире в ходе изменений — и тогда наверняка эта «деятельность предприятия» будет каким-то совсем другим объектом, у него будут другие свойства (как «танцевание танцора» стало совсем другим, когда включилось в «танцевальный перформанс»). Хотя деятельность предприятия много сложней танцевания, но по большому счёту не так уж от танцевания и отличается, там взаимодействуют в ходе этой деятельности люди, оборудование, здания и сооружения, сырьё и полуфабрикаты. Вот по ним и можно постучать. Предприятие существует в нашем мире. Несмотря на то, что в нём непрерывно идут процессы, и оно участвует в процессах во внешнем окружении, можно с ним работать как с «вещью», хотя и состоящей из очень многих других вещей, участвующих в его деятельности::поведение.

Системы какого-то предприятия (**оргроли**, выполняющие **методы/оргпроцессы**[137] **работы**, противопоставляемые **службам/оргзвеньям**, выполняющим какие-то **оргсервисы как работы**) при понимании этого предприятия как совокупности физических объектов (здания, материалы, оборудование и даже люди), играющих какую-то в нём роль, становятся вполне «физичными», неабстрактными, имеющими пространственно-временную протяжённость. Все составляющие предприятие физические объекты легко представить, их легко обнаружить по их описаниям, их легко описывать. Для начала нужно просто перечислить входящие в оргпроцесс физические объекты — и сразу станет понятно, одинаково ли вы понимаете этот оргпроцесс с другими людьми на предприятии. Но нельзя «подводить под определение», нельзя определять

[137] Мы вслед за ISO 29148 рекомендуем говорить не «бизнес-процессы» (business process), а «организационные процессы» (organizational processes), оргпроцессы.

оргпроцессы через задание для них каких-то классов операций/процедур, это только приведёт к спору о терминах. И нельзя считать регламенты оргпроцессов ими самими: регламенты только описывают процесс! «Постучать» по регламенту-описанию (который представлен/presented на экране, или бумаге, или даже в компьютерной памяти) нельзя, но можно постучать по регламенту-на-носителе, по **документу регламента,** процессной документации. И это постучать по совсем другому объекту, чем оргпроцесс/работа, определяемые регламентом. Постучать по человеку и постучать по фотографии человека — это стучать по разным предметам! Постучать по документу можно, а вот постучать по процессу, который описан документом — нет, будете стучать по системе, которая осуществляет процесс.

Мы часто будем приводить в качестве примера поведения системы танцевание танцора — танцор (как будет показано дальше в руководстве) не так тривиально разбирается на части, как насос или велосипед. Но танцевальные перформансы с танцеванием танцоров всё ещё много проще предприятия с его работой работников, поэтому думать о танцах проще, чем о предприятии. И совсем недаром одна из классических (год выпуска — 1999) книг Peter Senge по системному мышлению для предприятий называется «Танец перемен»[138].

Ещё один пример — интеллект, как система, реализованная мозгом или аппаратурой компьютера. Его поведение — мышление. По мышлению не «постучишь», а вот по интеллекту (мы понимаем, что это часть материальной системы-вычислителя) — вполне, даже если мы не знаем точно, какие именно части (нейроны и иные клетки мозга — какие именно, транзисторы классического компьютера) выполняют мышление::поведение.

Если речь идёт об одном объекте-экземпляре, который надо сделать и для этого сделать организацию создания — это хороший выбор системы. Мероприятие (в нашем случае — «танцевальный перформанс» это такое микро-мероприятие, а танцевальная вечеринка — просто мероприятие, танцевальный фестиваль — большое мероприятие, их «делают», «проводят») — это вполне нормальный выбор системы. Понятно, как об этом думать. Но всё одно придётся определять, что такого делает мероприятие со своим окружением и что входит в состав мероприятия, а что входит в состав окружения.

Если это рутинный «рабочий процесс»/метод, или «физический процесс»/функция («течёт вода», «убирается помещение», «работает фирма» — и это проходит «от забора и до обеда, конца-края не видно»), то есть речь идёт не о чём-то создаваемом, а о методе/культуре/«рабочем процессе»/деятельности, то это плохой выбор системы, договариваться по поводу такой «системы» будет крайне трудно, формально будет всё верно, а проекту от такого выбора системы — сплошной вред.

Ещё раз: мероприятие — ОК как создаваемая и развиваемая система (а также игровая сессия, сделка и прочие нестандартные не очень похожие на деталь или завод объекты, и для них понятно, как их описать и как о них думать как о системах), а вот метод работы или работа — не система, а поведение какой-то системы (метод осваиваем, методу обучаем, с методом проводим «постановку метода работы», но не «создаём экземпляр метода» или «выпускаем метод мелкой серией». Сравните: создать пять игровых сессий хорошо звучит, это не «создать пять методов ведения игровых сессий», понятно что метод обсуждаем отдельно, игровые сессии —

[138] https://www.ozon.ru/context/detail/id/1378492/

отдельно. Игровая сессия тут — это состояния игрового сервера с игровым миром и игрока, которые удерживаются совместно во взаимосвязи, понятно как о них рассуждать как о материальных объектах, а не «поведениях». Мы позже в руководстве будем обсуждать подобные примеры «неклассических/нежелезных систем»).

Что делать, когда непонятно, хорошо это или плохо? Например, «давайте организуем сбор урожая» — это создать систему «сбор урожая», или организуем рабочий процесс сбора урожая, как надо? Жёстких критериев нет, рецепта (на уровне общих наставлений, без вникания в детали конкретного проекта) не бывает. Онтологические выборы (какие объекты каких типов выделять в мире) — они всегда вероятностны, иногда и через год надо возвращаться и что-то переопределять в проекте: что там хорошо выбрано как система, а что плохо выбрано. В ходе многочисленных обсуждений целевой системы нашими инженерами-менеджерами они переопределяют выбор целевой системы чуть ли не еженедельно, по пять-шесть раз, и каждый раз считают, «вот этот вариант — самый лучший!». Но когда «попадают» — всё волшебным образом начинает объясняться окружающим за пару минут, договорённости в масштабах проекта достигаются мгновенно. Интересно, что в речи название целевой системы обычно появляется очень рано, но называемый объект никто не считает целевой системой, а иногда нужен месяц или даже пара месяцев, чтобы сообразить, что это и есть целевая система. Но в тот момент, когда делается удачный выбор — наступает огромное облегчение, «пазл сошёлся».

Если же просто провёл формальное формирование системы как «все вещи, участвующие в изменениях», то формально всё будет ОК, а договорённостей в проекте не будет, ибо «твой формализм понимаю — но он не решает мои непонятки с тем, как думать о процессах как вещах, а ещё я не могу договариваться с людьми дальше по графу создателей, тебя я понял, а меня уже никто не понимает». Будьте аккуратны: системное мышление — это про то, как договорить всех вокруг себя, а не про использование ментальных хаков типа «давай процесс сделаем системой».

Не злоупотребляйте тем, что процессы/изменения физичны, не объявляйте их системами. То, что существует мыслительный трюк представления какой-то динамической системы-процесса в физическом мире через перечисление участвующих предметов, не должно приводить к тому, что вы будете везде искать эти процессы и тут же объявлять их системами.

Если ребёнку в руки попал молоток, то все предметы в доме превращаются в гвозди. Если вы увидели в какой-то момент танцующего танцора и предприятие, ведущее деятельность, то это вы увидели танцора и предприятие, а не само танцевание и деятельность. Банан — это банан, а не созревание банана! Молоток — это вещь, а не всегда «забивание гвоздя»! Вам придётся собрать много других предметов, участвующих в изменении, и сделать из них другую систему (надсистему к забивающему молотку), чтобы справиться с упаковыванием процесса в какую-то систему (как это сделали с танцевальным перформансом, собрав надсистему к танцору), и название станет другим, и свойства этой системы станут другими, и — внезапно! — у этой системы тоже будет какое-то поведение, отличающееся от поведения частей этой системы, она сама себя будет как-то вести в окружающем мире (танцевальный перформанс зачем-то делают? Вот это «зачем-то» и будет функция::поведение танцевального перформанса. Если бы танцевальный

перформанс не производил каких-то изменений в мире, его бы не собирали из многих предметов, включая живых людей или роботов, не трудились бы над его созданием).

Да, если вы рассуждаете о «танцоре»::система, то это правильный пример воплощения системы, которая осуществляет «танцевание»::поведение и входит в состав «танцевального перформанса»::система — это хорошо поддерживается мышлением без дополнительных рассуждений, это общепринято. Но если вы вместо «буханки хлеба»::продукт/система/воплощение системы вдруг начинаете выделять «поедание хлеба»::«воплощение системы», то тут что-то не так, вы явно перемудрили, перепутали систему и поведение этой системы. Хлеб будет воплощением системы или едок хлеба? Вероятней всего, едок хлеба, хотя хлеб тоже будет участвовать. Но можно представить себе систему «обед» с едоком и едой, где будет и «поедание хлеба»::процесс, осуществляемый едоком::система, и в котором будет участвовать «буханка хлеба». Всё-таки поведение (работы, сервисы, функции, процессы, процедуры, практики, деятельности, культуры и стили поведения, виды труда, методы, стратегии — тут очень много разных терминов, обозначающих разные рассмотрения поведения) находят у каких-то объектов-систем, его само не делают главным героем в мышлении, а приписывают вещам. Есть вещи, они как-то себя ведут, то есть изменяют состояния других объектов в их окружении, а ещё они могут изменять и своё состояние.

Плохо все вообще объекты (объект — это нечто, на что направлено внимание агента) представлять, как поведение, внутри которого спрятаны какие-то взаимодействующие вещи, хотя в традиции системного мышления и были школы, где системы объявлялись процессами (или процессы объявлялись системами, что то же самое: процессы и системы приравнивались друг к другу, «поведение первично, а что именно себя ведёт-изменяется — неважно», и обычно там идёт ссылка на Гераклита). Эти школы не стали ведущими, такое мышление оказалось неудобным, оно не выжило в конкуренции с другими вариантами понимания природы системы. Сейчас общепринято, что системы участвуют в процессах-из-физики (то есть системы изменяются сами и изменяют своё окружение), а не сами по себе процессы-из-физики и есть системы, которые состоят из других процессов-из-физики — и так на много уровней.

Вопросы. Процессы как физические объекты

Какие утверждения верны для объекта, являющегося в четырехмерном пространстве-времени процессом создания чертежа стола?
--- Этот объект обладает протяженностью только во времени, в пространстве он ничем не представлен; ### Нет, участвующие в создании чертежа физические объекты (люди, инструменты и т.д.) занимают место в пространстве.

--- Этот объект обладает протяженностью во времени и представлен в пространстве самим чертежом; ### Нет, кроме самого чертежа в процессе его создания участвуют и другие предметы — люди и инструменты, которые занимают место в пространстве.

+++ Этот объект обладает протяженностью во времени и совпадает в период рисования со всеми объектами, задействованными в рисовании

(лист, карандаш, линейка, рисующий инженер). ### Да, все задействованные (участвующие) в рисовании чертежа стола объекты входят в этот процесс через отношение участия/participation (это подвид отношения состава/composition).
--- Этот объект обладает протяженностью во времени и представлен в пространстве рисующим инженером ### Нет, в процесс создания чертежа входят и другие объекты, занимающие место в пространстве.

--- Этот объект обладает протяженностью во времени и совпадает в период рисования со всеми объектами, задействованными в рисовании, а также со всеми созданными по этому чертежу столами до конца их жизненного цикла. ### Нет, созданные по чертежам столы не входят в процесс создания чертежа. Создаваемый чертёж — описание этих столов.

--- этот объект — система ### Нет, это не система, это процесс. Создание стола ведётся системами-агентами (людьми), задействующими в ходе этого процесса и другие объекты.

Компьютерные программы как системы

Программа::система («системная» вроде ОС или СУБД, программное приложение/application) в момент их работы — это вещь, физический объект, она занимает место в пространстве-времени, она материальна. По работающей программе (части компьютера, которая меняет своё состояние в ходе работы программы) можно «постучать», можно ткнуть в неё пальцем. Работающая программа — физическая часть компьютера, которая проводит вычисления этой программы в ходе её работы. Программа описывается её алгоритмом (описание порядка вычислений), документированном в виде программного кода на каком-то языке (машинном в памяти компьютера в момент исполнения программы, языке программирования в момент написания программы). Но алгоритм не меняет своего состояния, когда он «работает», ибо это описание. А вот компьютер, интерпретирующий алгоритм, меняет своё состояние в ходе исполнения этого алгоритма — и та его часть, которая занята выполнением алгоритма, и есть программа::система. Структурированные данные меняют своё состояние в ходе выполнения программы. Алгоритм тоже может меняться в ходе выполнения программы (часть алгоритма может вычислить другую часть алгоритма как свои данные, а потом эта другая часть алгоритма тоже будет вычислена). В любом случае, программа (алгоритм и данные в памяти, «оживляемые» вычислителем с каким-то процессором — если речь идёт о классической компьютерной архитектуре, но тут могут быть варианты с разделением процессора и памяти, не всегда они чётко разделимы) в ходе её работы — это объект, который меняет свои состояния и тем самым как-то себя ведёт, это объект четырёхмерного пространства-времени, его можно считать системой.

У программы как физического объекта в момент работы есть разные состояния, которые представляют собой физические состояния оперативной памяти и регистров процессора, отражающие текущее состояние алгоритма и изменяющихся в ходе выполнения программы данных. Компьютер занят физическими процессами/изменениями/взаимодействиями своих составных частей (памяти, процессора) в ходе вычисления данных по алгоритму как описанию программы. Эти

процессы в компьютере занимают какое-то место в физическом мире: пространство, в котором расположены взаимодействующие части компьютера, и время, во время которого программа (то есть части компьютера в её составе) проводит вычисления:

Ещё раз подчеркнём: программу следует считать воплощением системы только в тот момент, когда она реально запущена на исполнение и работает, делает то, ради чего она была написана. Это довольно контринтуитивно, но **исходный код программы — это не программа (алгоритм и его данные), но только описание программы; исходный код программы в системе управления версиями или просто в файле на носителе — это только документация программы. А ещё есть данные, программа их получит после начала работы и выдаст на выход после окончания работы. Не путайте «описание на носителе», фотографию человека с самим человеком. Программа — это то, что отражает состояние данных в момент её исполнения, это то, что меняет свои состояния во времени исполнения (а не в момент создания).** Исходный код не меняет своих состояний в ходе исполнения, в нём нет изменяющихся данных, это просто описание алгоритма и структур данных.

Поэтому программисты, которые считают, что их инженерная работа закончена в момент написания исходного кода (алгоритма и структур данных) — эти программисты глубоко неправы, и это типичная ошибка. Из признания этой ошибки появилось целые движения DevOps[139], SRE[140] и platform engineering[141] — программисты признали, что они должны быть озабочены не только разработкой кода программы как алгоритма и структур данных (время ***Dev***elopment), но и быть операторами работающей на серверах программы (время ***Op***erations). Подробнее об этом в руководстве по системной инженерии.

Исходный код — это описание программы (оно делается «в классах», как любое проектирование, один исходный код описывает множество возможных экземпляров программы). Перед использованием/эксплуатацией программы её нужно изготовить, используя исходный код в качестве «описания с детальностью, достаточной для изготовления»: откомпилировать, собрать, разместить в оперативной памяти нужного компьютера (возможно, перед этим оформив в какой-то контейнер) и передать на неё управление. Сейчас и исходный код на языке становится промежуточным внутренним представлением, как и машинный код: описание программы делается на естественном языке, а потом текст программы пишет и отдаёт его на выполнение AI-агент, чаще всего называемый Copilot. Нам не

---

[139] https://en.wikipedia.org/wiki/DevOps, есть похожий вариант этого движения, SRE и более современный вариант platform engineering

[140] https://en.wikipedia.org/wiki/Site_Reliability_Engineering

[141] https://platformengineering.org/blog/what-is-platform-engineering, https://thenewstack.io/how-is-platform-engineering-different-from-devops-and-sre/

так важны сейчас эти особенности создания программы из каких-то описаний. Нам важней сам факт различения

- исходного кода::описания программы (алгоритма и структур данных),
- Файла::документ с описанием программы («информационный объект»),
- работающей программы::система.

Тем самым **программа как система ведёт процесс изменения как своих состояний, так и состояний окружения (например, устройств ввода-вывода), производя операции над данными, записанными на вычислитель/computer как на носитель данных.** Выполнение::процесс::поведение программы::система::«предмет выполнения::метод» — это работа::поведение этой программы на каком-то работающем экземпляре компьютера (или работающих экземплярах компьютеров — например, клиентском и в облаке), в тот момент, когда часть этого компьютера (или компьютеров) работает в роли программы, задействуя метод «выполнение», то есть меняя в ходе работы физическое состояние тех своих частей (мест в пространстве-времени), которые являются носителями информации об алгоритме и данных программы. Понятно, что от исходного кода до вот так работающей на компьютере программы обычно долгий путь, который раньше по большей части проделывался руками программистов и операторов компьютера, а сегодня проделывается самими компьютерами (используется инструментарий компиляторов, линкеров/мейкеров, инструменты DevOps).

Ошибка, которую делают программисты, считая свой исходный код программой, ровно того же сорта, которую проектировщики и конструкторы делают, считая своей системой разрабатываемые ими информационные модели (а раньше — чертежи) и другую проектную и конструкторскую документацию. Карта не территория, меню не едят, на чертежах не летают, исходный код не хранит значений своих переменных (не хранит изменяющиеся данные) в ходе исполнения[142].

Итак, первая ошибка — это считать, что исходный код программы и есть программа. Нет! Исходный код — описание программы!

Вторая ошибка — это считать программу::система (даже не путая её с исходным кодом) той системой, которую нужно создать и развивать. Если спросить разработчика программы, какую систему он с коллегами делает, обычный ответ — «моей целевой системой является программа». Это ошибка: считать программу целевой в ситуации, когда программа содержит только описание целевой системы в своих данных (например, это часть системы управления/цифрового двойника, не имеющая отдельного смысла от управляемой

[142] Тут есть нюанс, связанный с фон-неймановской архитектурой: программа может быть рассмотрена и как данные на носителе, и как исполняемый объект. То же относится к «программе в мозге»: лежит ли в мозге в его нейронах только описание, или же нейронная сеть сейчас именно «вычисляет» что-то, то есть там работает программа-как-процесс в физическом мире — это тот самый вопрос про «материальность мысли». Это нюансы, мы их тут не рассматриваем. Совет тут — рассмотреть вероятность, с какой речь идёт о «просто документированном описании программы/мысли» или «исполняющейся программе/думающейся мысли», а вероятность эту брать исходя не из «истинности», а из целей какого-то действия. И обсуждать нужно наиболее вероятную ситуацию, полезную для осуществления вами задуманного. Но это нюанс, он непринципиален в большинстве ситуаций. Мы помним, что у нас тут в системном мышлении не математика со 100% формальными утверждениями, плохо приложимыми к жизни, а вероятностные рассуждения, «как в жизни».

системы/физического двойника). Например, мы уже рассматривали, что ERP-система как программа в своих данных содержит описание потока комплектующих через предприятие, она нужна только для того, чтобы как-то совладать с этим потоком, оптимизировать его.

Как только разработчик забывает, что его программа только цифровой двойник физического двойника, и что он разрабатывает целую систему из цифрового двойника и физического двойника, то проект будет провален. Это как если бы разработчика самолёта не интересовал самолёт в целом, но только его система управления. Нет, работоспособность ERP-системы проверяется не по тому, что она сама работает, а по тому, как она управляет потоком комплектующих через предприятие, важно поведение физического двойника. Работоспособность системы управления самолётом (цифровой двойник) определяется не сама по себе, но по работоспособности самолёта (физический двойник). Важно поведение физического двойника, и создание и развитие системы из подсистем цифрового и физического двойника. И даже в более слабых вариантах **цифровой тени** и **цифровой модели** это тоже выполняется: цифровой двойник — это автоматическая система управления, цифровая тень — автоматизированная система управления — она получает данные с датчиков физического двойника, но вот дальше просто показывает состояние оператору, и тот «руками» управляет системой. Цифровая модель и данные получает «вручную», и выдаёт параметры управления для «ручного» управления. Скажем, навигатор в автомобиле с автопилотом — цифровой двойник, навигатор со спутниковым датчиком и голосом говорящий, куда рулить — цифровая тень, а если навигатору надо говорить, где вы находитесь и он будет говорить, куда дальше ехать — а рулить надо вам, то это цифровая модель. Разговор о софте необходимо сегодня вести с учётом того, что этот софт или сам по себе агент, или цифровой двойник, или цифровая тень, или цифровая модель какой-то другой системы — физического двойника (ну, или софт — часть сети софтов, **digital twin network**, и всё равно нужно думать о физическом двойнике, который управляется этой сетью софтов).

Третья ошибка «программы как системы, которую мы разрабатываем» — считать саму программу разрабатываемой системой, а не какой-то подсистемой в ситуации, когда речь идёт о поддержке работы какого-то оргзвена с людьми или другими программами. Если ваша программа выдачи зарплаты чудесно работает, а вы забыли научить работников предприятия или AI-агентов (или даже другие программы в рамках автоматизации) мастерству работы с этой программой, то это означает, что программа не работает: зарплата-то не выдаётся! Если оплата ведётся только «за программу», то можно потом долго спорить, кто оплачивает обучение сотрудников или AI-агентов, или оплачивает перепрограммирование всех связанных программ (которое может быть много дороже разработки самой обсуждаемой программы выдачи зарплаты).

В случае ошибки с цифровым двойником программа и её данные (например, в базах данных) понимается как книга: нас чаще всего волнует не книга, и даже не содержание книги (сам книжный текст), а описанное текстом книги. Волнуют описанные в книге миры, не волнует сама красота слога (ну, разве что «особых ценителей», если речь идёт о беллетристике, письме ради письма). С программами то же самое: волнует описанный программой фрагмент окружающего мира, не волнует сама программа (волнует не больше, чем книга и её текст). Если у вас программа заказа химчистки, то вас волнует одежда, которая из грязной станет

чистой, деньги, которые с вас спишут, варианты транспортировки вещей на чистку и назад. Программа волнует программиста, но это его личное дело: всех остальных волнует, как эта программа справится с собственно ситуацией химчистки, да и программиста тоже должно это волновать. Если программа будет шикарна как софт, но химчистка с ней будет невозможна — то это провал проекта! **Софт коварен, если его представлять целевой системой, он чаще всего работает с описаниями целевых систем, ищите за ними физический мир, изменения в котором имеют значение!**

В случае ошибки с «в нашей ответственности только программа, про людей и организацию ничего не знаем» происходит включение в целевую (или даже «нашу») систему только софта, но не оргзвена той службы/service provider, которая выполняет какой-то осмысленный сервис (работы, меняющие что-то вовне этой службы), поддерживаемого при помощи софта. В корпоративной разработке софта клиенты ожидают по итогам этой разработки не столько корректную работу компьютера с приложением::программа, сколько корректную работу той части организации, которую должен этот компьютер поддержать: **люди в организации должны вместе с программой сработать по какому-то организационному алгоритму, «рабочему процессу», выполнить регламент работы, в котором учтено использование программы — то есть в них должно сработать «мастерство», а программа при этом будет инструментом этого мастерства.** Софт без людей не работает, нужно предъявлять работу софта-инструмента с людьми (хотя сейчас всё чаще и чаще люди заменяются другим софтом, в том числе софтом с AI), но на контроле всё-таки не изменения состояний самой программы (поведение программы как результата разработки, «софт выдачи зарплат работает»), а изменение состояний чего-то вовне программы («зарплаты выдаются», для чего «выдающие зарплату и получающие зарплату люди работают с софтом выдачи зарплаты», то есть и «люди работают», и «софт работает»). У вас не примут работающую дрель. У вас примут работника, который будет вашей дрелью сверлить дырки с стенах: нужны дырки, а не дрель! А там работник хоть пальцем пусть сверлит, если будут дырки! Если дрель есть, но дырок по каким-то причинам нет — за дрель не заплатят, ибо не за что платить. То же самое про софт: нужны результаты работы софта как инструмента, а для этого нужно, чтобы работник продемонстрировал результат этой работы с инструментом.

Такой совместный сервис людей и компьютеров называется обычно workflow («поток работ»::поведение), но не менее часто называют такой поток работ и «оргпроцесс», и «рабочий процесс», обращая внимание не на собственно экземпляры работы, а на метод/способ работы и заведомо включая операции программы в общие операции оргпроцесса. Программа-в-разработке — это чаще всего только часть целевого оргпроцесса-в-разработке. И сдавать готовую программу-в-работе не удастся саму по себе, её работа никому не нужна. Нужен оргпроцесс-в-работе, в котором программа только часть, а ещё там участвуют и другие программы, и люди, и иногда ещё и какое-то другое оборудование, и все они должны правильно взаимоизменяться. Если «приложение учёта деловых поездок» работает, но в командировку никто отправиться не может, потому как никто не знает, как пользоваться программой, то за работу программы платить не будут. Платить будут за деловые поездки, которые получатся только тогда, когда сработают вместе люди и «приложение учёта деловых поездок». За работоспособность только софта платить никто не будет! Софт обычно только часть той системы, которая создаётся и развивается проектом разработки!

Про процессы (в том числе оргпроцессы, рабочие процессы) люди думают трудно. Поэтому программа тут будет частью оргзвена (подразделения, рабочей группы проекта, службы, провайдера сервиса), которая будет выполнять нужные изменения в конечном итоге (через цепочки многочисленных изменений описаний и описаний описаний) в физическом мире. Сама по себе программа будет личным делом программиста или небольшой команды программистов, удерживать её в эпицентре внимания как полноценное воплощение какой-то целевой системы — опрометчиво, это обычно только часть какой-то системы, типа как мотор в электромобиле или система управления стиральной машиной. Мотор — важная часть электромобиля, но нужен обычно не мотор, а электромобиль. Если в электромобиле хорошо работает мотор, а сам автомобиль от этого хорошо работающего мотора не едет, ибо мотор плохо согласован с его конструкцией, то за мотор никто не заплатит, заплатят только за работающий электромобиль в сборе. То же самое про систему управления стиральной машиной: нужна не она, а работающая стиральная машина, разработчикам заплатят только за неё, а разработчикам системы управления достанется только часть этих денег в той мере, в какой они справляются с управлением стиральной машины, а не в той мере, в какой система управления работоспособна сама по себе.

Дело ещё осложняется тем, что программы/софт/приложения как информационные объекты (носители информации с описаниями алгоритмов и данными в базах данных) могут выстраиваться в длинные цепочки, в которых на базе одних описаний программы (или даже программы-внутри-подразделений-с-людьми) меняют другие описания, дальше третьи — и только в конце такой длинной цепочки «описанных описаний» наступает желаемое изменение внешнего физического мира. Вы можете долго описывать себя, где взять ваши деньги, ваши намерения, записывать промежуточные результаты принятия решений всевозможными другими программами и людьми в компьютер и/или телефон, но только в конце длинной цепочки заказа авиабилетов и проверки билетов, а также проверки наличия самолёта на стоянке, вас физически пропустят на самолёт: сотни самых разных программ и людей, часть из которых вам известна, а часть даже неизвестна, срабатывают, чтобы получить в физическом мире ваше состояние «я сел в самолёт, который готов к перелёту». Если этого результата не случится, все эти программы и люди сработали зря. **Поэтому не ленитесь раскручивать все эти «описания описаний», все длинные цепочки программ и программ-с-людьми — пока не дойдёте до изменений в физическом мире, ради которых всё делается. И убедитесь, что в физическом мире будет всё в порядке. Да, это трудно, но системное мышление заставляет развернуть ваше внимание в эту сторону. Это важно: выход обсуждений работы софта в обсуждение изменений физического мира — это имеет значение, если надёжных и ожидаемых изменений в физическом мире не будет, то проект разработки софта можно и не делать вообще, всё одно ведь в физическом мире ничего не изменится!**

Например, ровно эта связь между объектами софта и физического мира обсуждается в подходе DDD (domain-driven design), где требуется, чтобы объекты программы напрямую отражали объекты физического мира в предметной области работы компании (впрочем, там требуют и отражения объектов ментального

мира — мира описаний, но это для нас не так важно сейчас)[143].

Целевая система в проектах разработки софта

Как же найти изо всех возможных систем целевую систему, которую изготавливают программисты в конечном итоге, ту систему, изготовление которой будет считаться успехом проекта («все удовлетворились, никого не обидели»)? Ибо даже работающая программа оказывается часто работающей с описаниями и документированием описаний какой-то другой системы, нежели реальной системой, изменяющей физический мир — и надо мыслью тянуться дальше по цепочке, пока не найдём систему, которая таки меняет что-то в физическом мире, а не только в описаниях!

Для начала надо просто пройтись по списку ошибок из предыдущего подраздела. Хотя этих ошибок может быть много больше, но и этот короткий список обычно сильно продвигает понимание. Скажем, предположить, что ваша программа — это цифровой двойник, и тогда вам надо найти физического двойника, а система будет состоять из цифрового и физического двойников. Соответственно изменить и проект: сдача программы будет не неубедительной для заказчика демонстрацией работы самой программы с её базой данных как цифрового двойника (но именно «демонстрацию работы только программы, и ничего сверх этого» хотели бы разработчики: «бумага всё стерпит», оторванная от жизни софтинка вольна фантазировать свои результаты, успешность фантазий показывать очень легко), но демонстрацией надлежащей работы физического двойника под управлением цифрового двойника. Продемонстрировать изменения физического двойника под управлением цифрового двойника много труднее, чем продемонстрировать оторванного от реальности цифрового двойника. Но только это и будет реальным успехом проекта, заплатят за это. Другое дело, что когда речь идёт о ракете или мосте, или даже о пациенте в больнице, то физический двойник очевиден. Но как его найти, если речь идёт о паутине объектов и их отношений в каком-нибудь предприятии? Что там может быть физическим двойником, работоспособность которого в паре с цифровым двойником надо продемонстрировать?

Первый же **мыслительный ход, которым вы найдёте физического двойника — это смотреть не на алгоритм (описанный исходным кодом), а на данные этой программы (часто они лежат в какой-нибудь базе данных). Эти данные описывают какую-то систему (включая процессы этой системы, то есть изменения состояний системы) — чаще всего эта система и будет физическим двойником.** Именно эта описываемая данными вашей программы система и будет воплощением той системы, которым будут озабочены все окружающие разработку люди! Этот мыслительный ход срабатывает не в 100% случаев, но может быть хорошим стартом поиска целевой или «нашей» системы, когда вокруг много программистов. В более слабой версии этот же ход хорош и для информационных объектов (отчётов, моделей, баз данных и т.д.): что описывают эти объекты, какую систему? Интересует она, а уж сам информационный объект главный только для команды разработчиков, остальных он интересует только по сопричастности к происходящему с описываемой этим объектом системой.

Если речь идёт о программе учёта командировок, то целевая система — то, что описывается её данными, и будет в результате работы программы изменяться

[143] https://en.wikipedia.org/wiki/Domain-driven_design

(создаваться, уничтожаться). Это деловая поездка. Да, деловая поездка — это вроде как «езда и проживание командированного», но она собирается из разных объектов до надсистемы, в которой эта езда происходит (вроде как «танцевальный перформанс» из «танцевания танцора») и сразу становится понятно, как её описать и учесть в компьютере. Для этого просто нужно учесть все входящие в деловую поездку (отношение участия/части-целого/состава) объекты — людей в роли командированных, билеты как описания поездок («проездные документы», с разными нюансами отношения к бумажным и электронным носителям для этих документов), часто сюда относят и гостиницы, и их описания — платёжные документы по проживанию, и багаж с его оплатой и документированными описаниями «багажные квитанции», и т.д.

И тут надо не промахнуться, ибо «командировкой»/«деловой поездкой» называют разные процессы::поведения или мероприятия::система (и вам надо определиться ещё — вы это считаете процессами, или системами?):

- Посылающий в командировку (условно: «начальник») называет «командировкой» работу, которую надо выполнить где-то в другом городе. Часть этой работы — сначала агенту надо попасть в этот город, но это мало интересует.
- «Командированный» агент считает «командировкой» не столько результат работы (это для него просто «результат работы»), сколько «поездку»: полёт на самолёте, поездка на автобусе, проживание в гостинице, питание.
- «Бухгалтер» как агент считает «командировкой» не столько всю «поездку», сколько «подотчётную часть поездки»: не любой проезд, а только тот, который надо будет компенсировать. Скажем, до города проезд будет оплачен, а разъезды по городу — нет, а вот «командированный» считает это всё «командировкой».

Поскольку разные агенты считают командировку разными процессами или разными системами — надо просто дать им разные имена (скажем, «удалённая работа», «деловая поездка», «подотчётная часть деловой поездки», а от слова «командировка» отказаться, как от запутывающего) и научить разных агентов использовать эти имена, чтобы не путаться в разговоре. И вот дальше задаться вопросом: софт, который помогает «оформить командировку» — он помогает создать какую из этих трёх систем?! Конечно, это будут совершенно разные софты, или части одного софта, работающие как цифровые двойники (или тени, или модели) для разных целевых систем: одни помогают начальнику с «удалённой работой», другие — командированному с «деловой поездкой», третьи — бухгалтеру с «подотчётной частью деловой поездки». Есть подозрение, что для всех так называемых административных процессов есть не меньше трёх вот таких рассмотрений, поэтому-то и трудно разрабатывать софт для администрирования предприятий: все путаются из-за неразличения разных систем под одинаковыми именами! Если вы будете находить при этом разные объекты (пары: системы и их поведения) в физическом мире (4D экстенсионализм) и давать разные имена — путаница прекращается, и все участники проекта быстро договариваются между собой.

Если речь идёт о программе для станка с ЧПУ, то целевая система — это описываемая данными этой программы деталь. Программа же описывает работы станка, а станок нужен для получения детали. Цель — деталь, деньги от неё, а всё остальное только цепочка, приводящая к этой цели. Если программа работает

хорошо, а деталь получается плохо — денег не будет. Так что с программами (включая **мастерство** как «программа для людей и AI-агентов» и тем более **личность** как совокупность всех видов мастерства одного агента) как целевыми системами нужно быть осторожными. Подробней про это в руководстве будет, когда будем обсуждать целевые системы проекта.

Ещё в мире софта (а весь мир потихоньку становится миром софта, особенно если учитывать нейросетевой софт современных систем AI и софт в головах людей: личность — это софт на аппаратуре организма) могут быть такие странные воплощения системы, как сессии — учебные сессии, или игровые сессии. Например, игровая сессия состоит из игрока в состоянии игры, игрового сервера, который помнит состояние мира игры, работающей на телефоне игровой программы, которая управляется игроком с одной стороны и сервером игры (через интернет) с другой стороны. Все эти объекты, взаимно меняющие состояния — это и есть игровая сессия. Все сотрудники игровой компании будут счастливы с таким определением целевой системы, ибо игровые компании продают эти игровые сессии, увеличивают их суммарное время! Обратите внимание, что ни одна из упомянутых тут других систем не будет достаточна для полноценного описания происходящего в компании, торгующей компьютерными играми на своих серверах: ни сам игрок в состоянии игры, ни серверный софт, ни софт в телефоне, ни софт платёжной системы. Оплата-то идёт обычно как раз за игровые сессии, с небольшими вариациями, и пока вы не опишете работающий объект «за что платят» — разговор с бизнесменами компании клеиться не будет!

Часто происходит путаница, когда софт представляет собой «маркетплейс». Мало того, что софт этого «маркетплейса» считают целевой системой, так ещё и игнорируют данные для этого софта, всех обслуживающих этот маркетплейс людей, клиентов этого маркетплейса, всех возможных посредников и провайдеров, которые работают с этим маркетплейсом, чтобы получить в итоге такое воплощение системы, за которое кто-то будет согласен заплатить. Никто не заплатит за «маркетплейс», это просто одна из частей, которая позволит зарабатывать: маркетплейс изготавливает то, что нужно окружающим его людям, и целевой системой будет то, что нужно этим людям, а изготовлено оно хоть лаптем, хоть маркетплейсом — это неважно.

Например, маркетплейс может изготавливать «доставку». «Доставка» — это не процесс «доставления», а какой-то товар (вещь!), находящийся в пространстве прямо на пороге того места, куда эта самая доставка предназначена. Вот эта «доставка»::система в момент её использования, то есть «клиент взял в руки купленный товар»::«процесс использования» (повторим: не доставка::процесс, а конечное состояние физического мира: доставка::система как «товар передан»::состояние и проведённые уже процессы транспортировки товара до порога владельца и передачи товара во владение) и есть целевая система, за это платят, это воплощено в физическом мире, это изготавливается, а всё остальное — это просто цепочки описаний в «маркетплейсе» и других системах, создающих эту целевую. «Доставка» будет наносить непоправимую пользу клиентам, а всё остальное будет наносить непоправимую пользу лишь самым разным разработчикам — они получат деньги от клиентов за вовремя полученную «доставку». Обратите внимание, что мы не написали тут «вовремя выполненная доставка», ибо это означает «доставка::работа::поведение», «выполняется» обычно работа. Правда, есть и словоупотребление «выполнено» с системами,

например «художник выполнил картину к праздникам» означает, что он выполнил все работы по созданию картины к праздникам. Так что «выполнил доставку вовремя» тоже можно было бы понять как «выполнил все работы по созданию доставки вовремя». Говорить можно очень по-разному, если вы следите за типами. Если не следите за типами — запутаете и себя, и других.

Обычно при **выявлении/discovering целевой системы** какие-то интересные предположения делаются за 30-40 минут обсуждения. Но через неделю-другую приходится уточнять эти предположения, быстро определить целевую систему обычно не получается. Скажем, «маркетплейсы» как провайдеры сервиса по изготовлению «доставки» из «товара» почему-то не получаются, если рассматривать только «доставку» как целевую систему, создаваемую «маркетплейсом», ибо транспортировка идёт в обе стороны: это «поставка против платежа» (сделка заканчивается, когда произошёл и денежный расчёт, и поставка товара). «Маркетплейс» нужен только тогда, когда он гарантирует финансово и поставщику, и покупателю именно вот эту «поставку против платежа». Если не гарантирует, то «просто Гугл» отлично справляется с задачей взаимного поиска покупателей и продавцов, и достаточного конкурентного преимущества перед «просто Гуглом» нет, поэтому и «маркетплейс» остаётся утопической идеей. Так что нужно очень внимательно относиться к тому, что является целевой системой в проекте: успех проекта непосредственно связан с выбором целевой системы.

Можно ли считать, что тут перечислены все типовые случаи целевых систем, связанных с софтом? Нет! Мир крайне разнообразен, целевых систем неисчислимое количество разных типов, каждый день появляются новые бизнес-модели, с этими бизнес-моделями появляются новые типы целевых систем. Физический мир меняется всё более и более изощрённо, а системное мышление позволяет не запутаться в этом море описаний и таки **обратить внимание на главное: что меняется в физическом мире, что является воплощением системы, что является целью всего большого (а не только вашей маленькой части) проекта, за что люди будут согласны заплатить.**

Все эти рассуждения легко переносятся из мира алгоритмов в мир данных. Ещё полвека назад не было даже персональных компьютеров (первый появился в 1980 году), а лет двадцать назад ещё считалось, что мир захватят сложные алгоритмы, которые будут хитро перерабатывать относительно простые данные в базах данных. Сегодня оказалось, что современное программное обеспечение сдвигается в сторону работы простыми алгоритмами над сложными данными. Даже в алгоритмах AI сработал этот принцип: самые простые нейросетевые алгоритмы над сложными данными весов нейросетей добиваются огромных успехов, разве что нужно очень много вычислительной мощности, которая позволяет выполнить эти простые алгоритмы (горький урок Sutton'a[144]).

Поскольку сложность из алгоритмов перемещается в данные, то системным подходом начинают интересоваться не только инженеры-программисты, но и инженеры данных. Никогда не нужно забывать, что данные — это в конечном итоге описания каких-то систем. Даже в больших языковых моделях их данные — это описание/представление/presentation нашего материального мира как системы, только в сжатом виде модели мира — результат процесса познания/learning. Но в момент их обработки какой-то программой они сами становятся частью системы

[144] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

этой программы, они меняют своё состояние, они могут пропасть, они ведут себя как «вещь». То есть данные для обработки их программой тоже нужно «изготовить» из первичных описаний. И когда мы интересуемся, как получить из данных полезный результат, то как и в случае программ, мы должны научиться их изготавливать из исходных данных — и мы по аналогии с DevOps будем говорить о DataOps[145].

Системное мышление тем самым нужно как программистам, так и специалистам по обработке данных, а с учётом того, что мастерство и личность — это тоже программы, ещё и специалистам по искусственному интеллекту, а также специалистам по работе с людьми. В силу углубления разделения труда можно упустить тот момент, что самые разные профессии заняты одним и тем же, могут быть поддержаны одними и теми же способами размышления.

Системное мышление поможет инженерам-программистам, инженерам по данным, инженерам-администраторам датацентров, менеджерам всех этих многочисленных ролей договориться между собой, а также поможет найти и обучить работать со своим софтом и данными всех тех сотрудников, которых затрагиваются этим софтом и данными. Программные системы (в момент работы программ), системы данных (в момент использования данных), личности и мастерство в их составе — всё это системы, в мире софта системное мышление применимо не меньше, чем в мире железных систем, или живых систем, или организационных систем. **Нужно не выпячивать именно программистскую часть (программную инженерию и инженерию данных), всё время помнить про подчинённую роль софта и примат физического мира, который описывается и изменяется софтом в компьютерах и головах, в том числе данными этого софта — и всё в проекте создания информационных систем тогда будет в порядке.**

Мастерство как система

Для того, чтобы клиент смог получить результат оргпроцесса, выполняемого каким-то оргзвеном в своей оргроли (например, внутренний клиент сможет получить зарплату::деньги::система, начисленную::оргпроцесс::поведение и выданную::оргпроцесс::поведение бухгалтерским отделом::оргзвено как казначейством::оргроль), приложение/программу/софт::система нужно настроить, добыть для него исходные данные, научить работать с использованием этого софта как инструмента сотрудников (вот буквально как ребёнка учат молотком забивать гвозди — гвозди не мимо нужного места, а молотком бить не по пальцам, вот так и учат использовать софт), а затем **проверить не столько софт, сколько работу всего оргзвена/службы/провайдера в целом, включая приданный этому оргзвену софт.** Софт при этом вроде как входит в состав подразделения примерно так же, как молоток входит в состав оргзвена каких-нибудь кровельщиков, которые с его использованием будут изготавливать крышу здания.

В современном мире может быть и наоборот, оргзвеном выступать софт, а в его состав будут включены какие-то люди (операторы, команда разработчиков), но требоваться будет всё равно работа не только одного софта, но и входящих в его состав людей.

Повторим: никого не волнует отличная работа программы начисления зарплаты, волнует само начисление зарплаты (можно представить его как деление какой-то

[145] https://en.wikipedia.org/wiki/Dataops

кучки физического золота на предназначенные для раздачи сотрудникам части) — и если начисления зарплаты не произойдёт, то программистам команды создателей софта трудно будет объяснить не получившим свои деньги сотрудникам, что с их программой всё в порядке, а неправы все остальные сотрудники отдела бухгалтерии, неправильно заполняющие поля этой программы и нажимающие не те кнопки. Сотрудникам всё равно, софт внутри бухгалтерии или бухгалтерия внутри софта, их волнует результат работы софта в конечном итоге!

Вы приглашаете к себе экскаватор рыть траншею, и приданного к экскаватору экскаваторщика (человека, который играет роль экскаваторщика с его экскаватором)? Или приглашаете экскаваторщика как подразделение из одного человека, в состав которого входит заодно и экскаватор? Я думаю, что вам проще считать, что есть подразделение-экскаватор, а в его состав входит ещё и управляющий человек, «софт управления экскаватором, исполняемый живым компьютером». Важен не сам экскаваторщик, а его экскаватор, результат ожидаем от экскаватора! В операционной нам обычно важен хирург и анестезиолог, а вот в кабинете компьютерной томографии — прежде всего томограф, а что там сменные бригады операторов при нём — это где-то внутри томографа как подразделения, состоящего прежде всего из томографа и нескольких человек к нему в придачу.

Если у вас в компании AI-агенты или мощный серверный софт, который даёт основные результаты, то обычно непонятно, приписываете ли вы этот софт (включая AI-агентов) к людям, или наоборот — людей к софту. **Люди в этом плане — тоже «универсальные роботы, у которых внутри специализированный софт по работе с другим софтом или оборудованием».**

В проектах по разработке программ/софта/приложений/цифровых двойников очень часто есть необходимость разрабатывать не только тот софт (традиционный или нейросетевой AI-софт), который существует в компьютерах, но и тот нейросетевой софт, который существует в мозгах людей как вычислителях. Этот софт/программу, реализующий алгоритм/теорию/дисциплину какого-то метода работы как физический объект мы будем называть **мастерством** выполнения какого-то «ролевого поведения»::«метода работы». «Ролевое поведение» — это поведение **агента** (который состоит из **организма** и/или **аппаратуры** в случае живых агентов-людей или не очень живых AI-агентов, а также **личности** и/или **программного обеспечения/программных средств** как совокупности всего софта/программ, работающего на этом организме/аппаратуре), взятое в части поведения одного только мастерства из состава личности, которое помогает личности играть одну какую-то роль. Грубо говоря:

- Человек::агент = организм + личность (личность как совокупность самых разных видов мастерства, «весь софт, все программы для мозга»)
- AI-агент = аппаратура + личность (личность как совокупность самых разных видов мастерства)
- Мастерство::система = обычная или нейросетевая программа::система, которая работает на базе какой-то аппаратуры AI-агента или организма человека, реализует алгоритм/теорию какого-то метода/способа работы.
- Метод::поведение — обобщённый алгоритм/теория, по которому работает мастерство, а также инструменты, которые нужны для того, чтобы метод мог быть выполненным (скажем, для метода рытья траншей землекопу кроме мастерства рытья траншей нужно иметь или лопату, или экскаватор —

инструменты, делающие возможным применение мастерства).

В проекте разработки корпоративного софта обязательно будут:

- учёт **аппаратуры** для этого софта («железо» или «виртуальная машина» — контейнеризация софта, учёт того, что часть софта исполняется на сервере в датацентре, а часть может быть и на телефоне с крошечным экраном, и т.д. — везде своё, но аппаратура должна быть!). Программа работает в составе **программно-аппаратного комплекса**, а не сама по себе!
- собственно разработка целевого **софта**, её все видят и признают.
- «изготовление **мастерства** в мозгах людей» (обучение работе с программой: нужно «изготовить» те части самых разных людей в самых разных ролях, которые смогут делать что-то полезное с программой — изготовить мастерство работы с программой). Заметим, что надо будет как-то учесть и организмы этих людей, те самые «мозги людей» — если людей вам не дали, то и мастерство вы в них не изготовите, всё то же, что с аппаратурой для «программно-аппаратных комплексов».
- Изготовление **мастерства** в AI-агентах (это только-только появляется, но тут будут свои особенности: нынешних AI-агентов и не запрограммируешь как софт, и не обучишь как людей, а мастерство работы с вашей программой от них потребуется)
- Изготовление исходных **данных** (data engineering) и разноска их по разным **носителям**, с которыми будет работать программа. А уж сама программа будет формировать выходные данные, и их тоже надо знать, куда и как положить.

Не пропускайте эти части! Они физичны, их тоже нужно «проектировать», а затем «изготавливать»:

- сам софт «разрабатывают» в ходе **программной инженерии/software engineering**,
- для изготовления мастерства в составе личности людей и AI-агентов их надо **учить** (и разрабатывать способы, которыми это будет происходить, и способы проверки научения — подробно об этом в руководстве по инженерии личности),
- дополнительный классический софт **настраивать** в части использования его интерфейсов (и проверять, всё ли настроилось).

А что же с людьми, которые работают с программами? Очень часто воплощением целевой системы проекта будет какая-нибудь часть организации, которая должна выполнить какое-то дело — выдать кредит, начислить зарплату, изготовить деталь. И если окажется, что программа работает «правильно» с точки зрения её разработчиков, но люди в организации не могут с ней работать даже неважно по каким причинам, то не будет засчитана за правильную и работа программы. Что толку в работе программы по начислению зарплаты, если люди с ней не смогут сработать? Если программисты хотят получить деньги за свою часть работы, они должны быть уверены, что кто-то работает и с людьми, а сдаваться заказчику будет совместная система из обученных людей и настроенных программ (а также данных к этой программе, что может быть отдельной проблемой и может требовать отдельных исполнителей). Сдаваться будет начисленная зарплата, изготовленная качественная деталь — и дальше по цепочке работа оргзвена, ведущая к этим важным целевым результатам, а внутри этой работы оргзвена (подразделения, проектной группы) будет сдаваться софт.

Для обучения людей проводите примерно такие же рассуждения, что и про компьютерные программы. Учебные книги, видео, компьютерные курсы, а также руководства/регламенты/стандарты/«описания рабочих процессов» (например, наше руководство по системной инженерии) — это только документированные описания того поведения, которое хотелось бы получить после обучения. Поэтому целью обучения (в том числе обучения работать по методам, описанным в руководстве/регламенте/стандарте) будет мастерство выполнять работу согласно методу.

Мастерство::софт::система — это какой-то работающий (система в момент эксплуатации всегда как-то себя ведёт!) физический кусок обученного мозга как нейросети. Мастерство как система в момент попадания её в целевое окружение включается по аналогии с компьютерной программой, которая включается в момент попадания её в окружение её вызова в компьютере и производит те вычисления/мышление, на умение делать которые было направлено обучение.

Мозг — это вычислитель, мастерство работает как часть вычислителя. Это неважно, что речь идёт о нейросетевом вычислителе. Неважно, какими конкретно частями мозга реализуется мастерство. Главное, что мастерство реализуется мозгом, а не

космосом, внеземными силами, вселившимися духами и прочими не очень правдоподобными причинами. Прочтённые книги, просмотренные видео лишь задают описание того, что потребуется от мозга потом, в ситуации использования мастерства. Прочтённые книги, просмотренные видео и выполненные упражнения лишь обучают нейросетку, повышая вероятность протекания в мозге физического процесса (со сменой состояний!) мышления в соответствии с результатами обучения в ситуации использования мастерства.

...

Единственное что, так это не нужно говорить о нейронах и их обучении, это явно не тот уровень деталей (не тот системный уровень, не тот уровень размера систем), который обсуждается при обучении. Но представление результата обучения как обученного для работы в целевых условиях участка мозга «мастерство» заставляет продумать и то, как выглядит это обучение в физическом мире (участок мозга должен в момент обучения быть где-то расположен, должны подводиться данные и выводиться результат мышления::вычисления), и как выглядит в физическом мире работа результата обучения как системы (должно быть соответствующее физическое окружение мастерства, чтобы оно было уместно. Умение делать сальто с места неуместно под водой, но уместно в цирке).

Скажем, в момент использования мастерства человек может находиться в машиностроительном цеху, среди жуткого грохота окружающих машин. Включится ли мастерство (софт/программа нейросети/обученный участок мозга) без дополнительного напоминания? Выдаст ли в этом окружении безошибочно и без пропусков то мышление, которому его учили? Сколько времени это займёт? Что происходит до этого момента, что будет происходить после этого момента? Переход к физической системе (участку мозга, который готов действовать/мыслить) оказывается продуктивным и для ситуации обучения. В обучении много описаний, но не они оказываются главными. Главным тут оказывается результирующий

физический объект: система с предписанными ей свойствами и известным поведением в момент использования.

Что является целевой системой проекта строительства моста? Мост, конечно! Но для строительства моста нужно создать организацию проекта, которая проведёт его создание, потом провести работу этой организации сначала в части проектирования (обычно в городе, в проектном институте), затем в части стройки (несколько тысяч человек в чистом поле!), и только потом получить удовольствие от моста. Это всё **цепочка создателей: создать создателя, потом создатель создаёт целевую систему.** В общем случае речь идёт о разветвлённых графах создателей, где создатели — это узлы графа, а рёбра — направленные отношения создания и развития (изменение состояния) создателями других систем этого графа. Часто ли нам нужно думать о создателе создателя создателя? Да, часто. Легко представить какой-то проект создания софта в проекте строительства моста. Зачем существует этот проект, зачем нужна команда разработчиков софта? Чтобы создать и развивать софт. Зачем нужен весь этот софт? Чтобы создать и развивать (как минимум — поддерживать) мост. Зачем нужен весь этот мост? Чтобы поддерживать (менять состояние) транспортного потока. Если нам нужен транспортный поток через реку, то мы создадим мост, для чего создадим софт, для чего создадим команду программистов, и так далее (чтобы создать команду программистов тоже нужно, чтобы кто-то был этим озабочен, провёл все нужные работы).

Системное мышление требует отследить всю цепочку от транспортного потока через мост до используемого софта. Если не отследить, то софт оказывается не нужен — хотя разработчики очень хотят за него получить деньги, особенно за ненужные никому «фичи»/возможности этого софта. Если возможности софта не будут использованы, то они могут и не работать как надо, но деньги за них заплатят, если удастся утаить связь этих возможностей софта с реализацией целевого моста. И если окажется, что целевому мосту от этого софта с его возможностями ни холодно, ни жарко (это никак не ускорит его строительство или никак не повлияет на транспортный поток), то это будет означать, что софт можно этот и не писать, проектом этого софта не заморачиваться, деньги за него не платить. **Системное мышление быстро проверит, не занимаетесь ли вы никому не нужной работой.**

Системное мышление заставляет об этом всём думать сразу, а не в момент неподписания актов приёмки-сдачи по проекту «создания программного обеспечения». **Обычно сами по себе программы как системы никому не нужны, они нужны только как части каких-то других систем — и нужно убеждаться, что разработка идёт этих других систем в целом прежде всего, а программ — только как части этих систем или как части систем, которые делают эти целевые системы — разбираться нужно со всеми цепочками описаний описаний, ведущими к реальности, к физическому миру.** Не путайте фотографию и изображаемый на фотографии предмет, книгу и живой мир, который описан в книге, софт и реальный мир, описываемый и меняемый софтом.

Вопросы на физичность отношения состава

Джон написал несколько примеров для руководства по системному мышлению, вы должны проверить их и отметить правильные.

+++ Пример отношения состава: «Документация состоит из трёх томов, каждый том состоит из трёх тетрадей». ### Да, всё в порядке: все молекулы тетрадей входят в число молекул томов, а все молекулы томов входят в число молекул документации. Осмысленные операции возможны и с документацией, и с томами, и с тетрадями, сшитыми в тома.

--- Пример отношения состава: «Описание состоит из рассказа по трём темам, каждая из которых содержит три сюжета» ### Нет, речь идёт не об отношении состава/композиции для физических объектов. Описание не занимает места в пространстве (но место в пространстве занимает носитель информации со знаками, в которых выражено описание), объекты описания «тема» и «сюжет» не занимают места в пространстве. Поэтому нельзя точно сказать, что там часть чего, в физическом мире это отношение не проследить. Системные рассуждения для этих «частей и целых» вряд ли будут работать во всей их полноте. В современном системном мышлении есть ход на конструктивистское описание (в котором от отношений переходим к операциям конструирования частей-целых, подробней будет это рассказано в последнем разделе нашего руководства), там можно решить проблему с составом описания, но этот «состав» будет отсылать не к «отношениям композиции», а к «операциям композиции».

+++ В спектакле есть три акта, каждый из которых состоит из трёх явлений. ### Да, всё в порядке: спектакль — это 4D объект (из актёров, декораций, зрителей, развёрнутый в некотором промежутке времени, аналогичный разобранному в руководстве «танцевальному перформансу»), и три акта — это крупное деление спектакля на темпоральные (то есть во времени) части, отделяемые друг от друга антрактами (когда спектакль вообще не идёт, и все зрители в буфете, антракты не считаем входящими в спектакль), а явления — это тоже истинные физические части четырёхмерного акта, которые отличаются приходом («явление») или уходом какого-то персонажа пьесы. То есть это всё истинные части четырёхмерного физического объекта. То, что спектакля нет во время антрактов, ничего не означает. Между молекулами воды тоже есть расстояние, но это не мешает каплю воды считать одним объектом, воспринимать её как целый физический объект, хоть и состоящий из (взаимодействующих! Как-то связанных друг с другом!) разделённых некоторым пространством более мелких физических частей. Разбирать на части ни воду, ни спектакль не надо, их части выделяем вниманием!

+++ Крепление состоит из трёх отверстий с резьбой М3, разнесённых по треугольнику на 6 см друг от друга. ### Да, это «крепление» имеет чёткие координаты в пространстве, понятны операции как его делать, понятно, что с ним будут делать после изготовления. «Состоит из» тут отношение композиции, части «крепления» — отдельные отверстия с резьбой М3.

--- Пентапризматор состоит из трёх модулей, каждый из которых состоит на учёте в пентарегистре ### Нет, пентапризматор может ещё иметь частями три модуля (состоит из трёх модулей), но вот добавочка «каждый модуль состоит на учёте» предполагает «состоит» явно не в значении термина для отношения состава/композиции/части-целого в физических объектах.

--- Люди клана Среднезубых состояли из воинов, торговцев и юристов, а воины в свою очередь состояли из лучников, кавалеристов и артиллеристов. ### Нет, тут описывается не деление клана Среднезубых как какого-то целого объекта на физические его части. Тут приводится классификация входящих в множество

Среднезубых объектов-людей (типа классификации Линнея для животных), это не отношение состава (например, нельзя сказать, что «воины» — это какой-то целый объект с его особыми свойствами и поведением, как нельзя сказать, что «мыши» — это какой-то целый объект среди «млекопитающих». Если говорить о таксоне/классификаторе, то это ментальный объект, а не физический).

--- Треугольник состоит из трёх сторон, а каждая сторона состоит из двух точек-вершин треугольника, соединённых отрезком прямой. ### Нет, треугольник не физический объект, его нельзя разбить на истинные части, рассказывается о «множествах точек», и это математические точки, а не точки физического мира. В современном системном мышлении есть ход на конструктивистское описание (в котором от отношений переходим к операциям конструирования частей-целых), там можно решить проблему с составом описания, но этот «состав» будет отсылать не к «отношениям композиции», а к «операциям композиции».

Задания: системы в физическом мире

Выполните задания:

1. Убедитесь, что в вашем рабочем проекте в конечном итоге создаётся физическая система, а не просто какое-то описание. Какова ваша целевая система? Что изменится в физическом мире, если ваш проект закрыть прямо сейчас, чего в физическом мире не будет существовать без результатов вашего проекта? Что делает целевая система в физическом мире, какой процесс она в нём осуществляет, какая «непоправимая польза» наносится миру этим его изменением со стороны этой системы? Чего не будет хорошего, если целевая система не будет осуществлять этого поведения? Напишите пост с ответами на эти вопросы в порядке мышления письмом. Посмотрите, что из этого поста можно добавить в материалы вашего рабочего проекта, чтобы проект лучше достигал своих целей, чтобы упростить объяснения этих целей и важность их достижения всем людям, которым приходится объяснять эти цели.

5. Методы/функции/практики/деятельности/культуры/методы работы и роли

Машинка типов

Системное мышление существенно задействует ваше мастерство по самым разным фундаментальным дисциплинам методов мышления интеллект-стека (физике, теории понятий, семантике, логике, онтологии, этике и т.д.). В вашей голове и теле к моменту попыток работы по нашему руководству по системному мышлению должна уже хорошо работать **машинка типов.** Да, и в теле тоже, ибо **онтологический дребезг как сигнал ошибки в типе** замечается как ощущение в теле, это ведь один из интерфейсов к вычислениям, ведущимся в мозге. Эту машинку типов изготавливают в инженерах-менеджерах по ходу освоения рациональной работы. Более подробно обучение мастерству как инженерное «изготовление мастерства в агенте на основе сухой или мокрой нейросети» рассматривается в руководстве по инженерии личности.

Если машинка типов работает в начале освоения мышления по наставлениям руководства по рациональной работе работает у 10% инженеров-менеджеров, то

на выходе — почти у всех, хотя и не очень бегло. Руководство наставляет выделять объекты по заданному их типу своим вниманием, а затем быть собранными: удерживать эти объекты во внимании. Это удержание вниманием не «ментально», ибо надёжности биологического мозга не верим, даже если этот мозг тренировать на «хорошее внимание», а технически — внимание удерживается путём записи, желательно в компьютере.

А дальше? Как сказал один из инженеров-менеджеров, который осваивал системное мышление в группе стажёров своим одногруппникам в те времена, когда мышление по руководству по рациональной работе ещё не было в обязательных пререквизитах к освоению системного мышления: «я вообще не понимаю, что вы вычитываете из руководства по системному мышлению, если вы не освоили мышление по руководству по рациональной работе!».

Машинка типов отвечает за присвоение типов и контроль типов в каких-то описаниях, документированных в текстах как на естественном языке с низким уровнем формальности, так и в моделях из объектов и отношений, документированных в текстах (или даже диаграммах) на языке с высоким уровнем формальности. Главное, чтобы у инженера-менеджера как будущего мастера было не просто рациональное понимание факта, что у объекта всегда есть какой-то тип, но его машинка типов работала с достаточной беглостью в рабочих проектах.

Что стоит за машинкой типов? Мозг поддерживает работу с разными теориями понятий/concepts[146]. Одна из простейших теорий понятий, объясняющих результаты работы мозга в режиме «быстрое мышление S1 по Канеману» — это теория прототипов, когда мы для каждого объекта представляем какой-то объект-прототип, и дальше действуем по аналогии. В этой теории понятий всё мышление — это непрерывная обработка метафор, поиск похожестей. Другая теория понятий — это существенно более сложная theory theory[147], в которой типизированные понятия связаны между собой отношениями (которые тоже можно выразить понятиями). Она соответствует медленному мышлению S2, формальному мышлению, логическим построениям. И тексты типа «сепулька это зверь. Есть ли позвоночник у сепульки?» разбирает именно такой механизм мышления. Если вы знаете, что у зверей есть позвоночник, то указание для сепульки типа «зверь» вполне достаточно, чтобы сообразить — да, позвоночник у сепульки есть. Если вы в уме не отнесли сепульку к типу, или просто это отнесение сепульки к типу не удержали во внимании до второй фразы в этом коротком тексте, то вы будете в недоумении — как отвечать про наличие или отсутствие позвоночника в абсолютно неизвестном объекте?!

Если вы хотите системно мыслить, то вы должны иметь встроенную в мозг и тело машинку типов. Режим медленного мышления S2 присутствует в системном мышлении практически всегда: мы системно мыслим с существенным задействованием теоретической теории (theory theory) понятий. Если вы не помните о theory theory — вам надо вернуться к руководству по рациональной работе.

Скажем, вам сообщили: «Лидер — это роль. Какая роль у лидера?». Если у вас нет в голове машинки типов, то вы начнёте что-то сочинять бытовое метафорическое про лидера, полагаясь на быстрое мышление S1 и прототипную теорию понятий с метафорами и образами: что он ведёт за собой людей, что лидер должен быть образцом для действий других людей, и т.д. (проверьте себя, ответьте на вопрос и

[146] https://iep.utm.edu/concepts/
[147] https://iep.utm.edu/th-th-co/

запишите ответ).

Если машинка типов у вас в голове есть, то вы должны прийти в недоумение: «лидер — это роль». В этот момент лидер у вас помечен типом «роль». Следующая фраза — «какая роль у роли»? Ибо если лидер помечен у вас в сознании как роль, то вы удерживаете это и до второй фразы, и проверяете на соответствие типу. Вопрос «какая роль у роли» должен страшно напрячь, ибо его осмысленность мала. **Машинка типов, которая встроена в голову, вскрывает все нелепости в тексте, она нужна для критики в ходе творчества.**

Если вам сказали, что инженер — это роль Васи, то нехорошо спрашивать инженера, как ему сегодня спалось. Инженер — это роль, всё равно что «папа» в игре в «дочки-матери» или Принц Гамлет в театральной пьесе. Спалось хорошо агенту (человеку или даже AI, которому нужно «спать» для дообучения), который играет роль инженера в рабочем проекте как ролевой игре, или «папы», или Принца Гамлета. Актёра/актора::агент и нужно спрашивать про «хорошо спалось». Такой вопрос инженеру::роль — это сбой в машинке типов, мозг такое должен запрещать «на автомате», поднимать тревогу! В принципе, если какой-то человек считает, что он не играет роль короля или инженера, или дерева в сказочной постановке, а сам и есть король, или инженер или дерево — он сошёл с ума. Людей и AI нужно чётко отделять от их ролей.

А как узнавать, когда термином «инженер» называют роль, когда должность, когда актёра-в-роли::агент? Из контекста, но для этого надо быть собранным/внимательным и удерживать в голове то, что «инженер» — это всё-таки роль, а не играющий эту роль агент. Спит Вася, а не инженер. Вася «отыгрывает роль»::поведение инженера::роль (то есть выполняет::работает «метод работы»/«вид инженерии/деятельности», **роли агентов «выполняют методы»/«работают по методам»::поведение**) как раз, когда не спит! Без разбирательства с типами в подобных ситуациях можно запутаться.

В среднем машинка типов надёжно «на автомате» работает в голове без специального обучения у примерно 10% имеющих высшее техническое образование (это наша статистика по группам стажировки по рациональной работе, всего один из десяти человек), а то и меньше. Почему так мало? Ведь все люди задействуют режим работы мозга S2, медленное логическое мышление? Это неизвестно. Что нужно было делать, чтобы она надёжно заработала? Есть разные гипотезы, одна из них — чтение большого количества нехудожественной литературы и разбирательство с формальными моделями, требующими удержания в голове длинных цепочек рассуждений с задействованием типов. Косвенно эта догадка подтверждается экспериментами с большими языковыми моделями, которые обучаются сначала на программном коде (это формальные языки со строгим контролем типов), чтобы у них экспоненциально выросла способность логического рассуждения[148].

Если вы не попали в те 10% населения, у которых машинка типов «просто есть, не знаю откуда», то вам с системным мышлением без освоения мышления по руководству по рациональной работе будет запредельно трудно: вы будете просто не понимать материал.

Системное мышление всё основано на theory theory, и логика там

[148] https://arxiv.org/abs/2401.00812

просвечивает через кажущиеся абсолютно художественными и гуманитарными тексты. Но нет, эти тексты прошиты типами: системное мышление заставляет оперировать типами существенно чаще, чем это встречается в обычной жизни. Эта работа с типами и даёт силу системному мышлению. Если вы по умолчанию используете в качестве инструмента вашего мышления теорию прототипов или образцов вместо теоретической теории/theory theory, то вы не справитесь с пониманием материала нашего руководства, не сможете овладеть системным мышлением.

Системное мышление даёт мета-мета-модель как набор типов. Типы системного мышления («система» с её разными подтипами, «окружение» и т.д.) указывают на важнейшие объекты внимания в рабочем проекте, помогают моделировать, составлять системные описания мира. И помним, что системные описания надо как-то документировать, они письменные, а не «в голове».

В нашем руководстве явное присвоение типа происходит с помощью нотации **::**, например, самолёт::система. Вы к настоящему моменту уже много раз встречали её в наших текстах.

Так что для освоения системного мышления есть пререквизит: хорошо и бегло работающая «машинка типов». Без неё системного мышления у вас не будет, надёжной и результативной коммуникации в сложных проектах тоже не будет[149].

Ролевые объекты/роли и «игра по роли»/функция/назначение в окружении

Одна и та же система может иметь множество описаний:

- Функциональное/ролевое: чайник нужен для кипячения воды. Он подогревает заливаемую в него воду до кипения, затем громко пищит, чтобы сообщить о том, что вода закипела.
- Конструктивное: чайник состоит из корпуса, нагревательного элемента, сигнализатора кипения (датчик, контроллер, динамик), шнура с вилкой для подключения к электросети.
- Размещения: чайник стоит на окне в кухне, воткнут в розетку слева от окна.
- Стоимостное: чайник стоит 5 тысяч рублей сразу и потом вы заплатите ещё за электроэнергию в зависимости от вашего тарифа, 1.2Квт за час работы.
- … и таких описаний может быть ещё множество. Художественное описание, чтобы оценить вписываемость в интерьер, описание возможных угроз (ожоги, удары током и т.д.) по линии безопасности, инструкции как проводить техобслуживание (удалять накипь) и ремонты, и т.д.

Помним, что в 4D экстенсионализме если вы описываете какие-то объекты и эти описания относятся к одному и тому же месту в пространстве-времени, то это один и тот же объект. Если вы описываете «тёплый предмет на окне в кухне», ваш друг «оранжевый предмет на окне в кухне», а я описываю «чайник на окне в кухне», и вы понимаете, что кроме этого чайника ничего на окне в кухне не стоит, то это просто речь идёт об одном и том же тёплом оранжевом чайнике на окне в кухне.

---

[149] См. об этом доклад Прапион Медведевой на 6й конференции ШСМ, «Машинка типов: зачем она нужна», https://www.youtube.com/watch?v=2T7P64GARzY&t=28850s (с 8:00:50)., и об этом же её доклад на 10й конференции, там цифры не изменились: https://www.youtube.com/watch?v=ULhN5QdbeR0&t=2213s (с 21:26).

Все эти описания вроде как имеют равный статус, но первые четыре — общепринятые, как самые важные, а первое из них — первое среди равных. Системное мышление рассматривает системы прежде всего как имеющие какое-то **назначение в своём окружении в момент использования/эксплуатации/функционирования**, то есть в готовом (а не в момент создания) виде выдающие для кого-то важное/интересное поведение в окружающую их среду. Это описание принято называть **функциональным** (какую функцию выполняет система в её окружении в момент её работы, упор на функцию::поведение) или **ролевым** (какую роль играет система в окружении, упор на ролевой объект::вещь, то есть объект, который как-то себя ведёт). Кому-то важное — для разных агентов из разных проектов для их собственного функционирования что-то может быть важным, а что-то неважным. Одному агенту от чайника надо, чтобы быстро кипятил воду, а другому всё равно, что там со скоростью кипячения, но важно, чтобы был оранжевым: этот другой агент занимается не чаепитиями, а интерьером. Системное мышление должно дать возможность договорить всех по этим важным вопросам, чтобы создаваемая система оказалась успешной. При этом «важное поведение» — это «важно нужное» или «важно вредное» поведение, главное тут не в собственно позитивности или негативности поведения, а в том, насколько оно оказывается важным для затрагиваемых системой или затрагивающих систему агентов. Это агенты-создатели системы, но также и агенты, которые заинтересованы в системах в окружении системы в ходе эксплуатации, но ещё могут быть и агенты, заинтересованные в проектах создания систем, находящихся в окружении надсистемы. Все эти интересы надо будет удовлетворить. Так что функциональных описаний (одна и та же система может играть разные роли в разных надсистемах, которые выделяют разные агенты в своих проектах, связанных с целевой системой) тоже может оказаться множество, придётся согласовывать и их. А поскольку мы тут помянули и агентов-создателей, и агентов-пользователей, и системы в окружении, то обычно функционально надо описывать и всех их. Системное мышление требует этого описания как первого.

Сначала надо понять, что делает система в момент эксплуатации, какую пользу или вред она приносит своим поведением: какая роль::вещь у системы и какая функция::поведение этой вещи (поведение роли в момент её работы). Роль — это сама система, в её физическом виде, ибо ролевой объект физичен, когда его роль выполняется конструктивным/материальным объектом в составе системы. Конструктивный/материальный/материализованный/materialized объект важен во времени создания/construction создателями/constructors, он представляет собой «материал» роли, вещество и поля. Впрочем, и ролевой/функциональный объект физичен, хотя его реже называют «физическим объектом. Физический объект существует и устойчив в меру существования во времени материала в какой-то форме (NESS, незначительные флуктуации для удержания устойчивости тут не влияют), и этот материал берётся независимо от его функции. А вот ролевая/функциональная система физична, ибо она сохраняется до тех пор, пока её функция::поведение выполняется какими-то конструктивными объектами. Тем самым мы рассматриваем систему минимально дважды (а потом мы узнаем, что минимально надо рассматривать систему четырежды, а обычно — много большее число раз):

- **Как роль:** функциональный/«ролевой объект»::«физический объект» (физичность обеспечивается воплощением конструктивными/материальными объектами) выполняющую какую-то функцию::поведение. Функциональная

вещь непрерывно существует, пока сохраняется роль (выполняется одна и та же функция), а материал может меняться. Чайник как функциональный объект есть, пока какой-то конструктив нагревает воду. Если не нагревает — не чайник.

- **Как конструктив:** конструктивную/«материальную вещь»::«физический объект», которая непрерывно существует, пока сохраняется её физический **материал**, а вот роль при этом могут меняться. Камень — это или пресс-папье::роль, или гвоздезабиватель::роль, но это просто камень, «материал», «конструктив».

Связаны эти объекты отношением воплощения: набор конструктивных объектов воплощает какую-то функциональную систему (роль).

В разных версиях системного мышления немного по-разному относятся к физичности функциональной вещи/объекта. Скажем, в ISO 15926 считают функциональный объект одним из подтипов (отношение специализации, «подмножество») физического объекта.

Four kinds of **physical_object** are recognized, discriminated on their basis of continuity. They are defined as subtypes of **physical_object** as shown in Figure 47:

Figure 47 — Types of physical object

Более того, в ISO 15926-2:2003 пространственное место (spatial_location) — это тоже физический объект, в силу 3D экстенсионализма. Как функциональный объект определяется на базе устойчивости назначения/поведения, как конструктивный/материальный объект определяется на базе устойчивости наличия материала, так и пространственный объект определяется на базе устойчивости места. И отдельно выделяется ещё один материальный объект, который в инженерии важен, но который трудно определить через постоянство материала, места и назначения: поток/stream, в потоке важно постоянство его пути/flow_path.

«Функциональный физический объект» и «материальный физический объект», связаны отношением «темпоральная целая часть»:

- слово «темпоральная» тут как раз обозначает то, что функциональный физический объект существует временно, только когда его как роль играет какой-то физический объект. Например, «крановщик» существует только в

тот момент, когда эту профессиональную/деятельностную роль играет какой-то человек или даже AI-агент, а «забивало» существует только тогда, когда эту роль играет камень или микроскоп.
- Слово «целая» — это когда в какой-то момент один конструктивный объект целиком выполняет роль функционального объекта (крановщик целиком состоит из играющего его роль человека, забивало целиком состоит из камня или микроскопа, а иногда из молотка, но тоже целиком).

В других школах мысли говорят о том, что функциональный объект — это не вот прямо «физический объект», а «физичен, когда воплощён конструктивным физическим объектом». А как надо? В разных инженерных школах, в разных школах философской логики даются разные варианты.

Ряд инженерных школ для конструктивного объекта, рассматриваемого в момент его игры по роли вводит понятие аффорданс/affordance/«подходяшка». Так, «Вася, обученный на курсах крановщиков» — это «подходяшка» для игры роли «крановщик», а «микроскоп» как материальный объект — «подходяшка» для роли «инструмент с функцией рассматривания мелких предметов», но в некоторых ситуациях это «подходяшка» для роли «забивало гвоздей».

Будьте внимательны к предлагаемой онтологии в её нюансах и обилию терминов для одних и тех же понятий, но общее везде одно: 4D экстенсионализм позволит разобраться. В любом случае, мы не можем обсуждать системы, пока не поймём:

- Функцию: что делает система (какая её функция, какого поведения мы от неё ожидаем)
- Конструкцию: какой материал выполняет нужное поведение
- Размещение: в каком месте этот материал выполняет поведение
- Стоимость: во сколько обойдётся создание и эксплуатация системы (совокупная стоимость владения).
- … и дальше тут может быть ещё много разных описаний той же системы, каждый раз уточняющих предыдущие и требующие согласования: все они должны непротиворечиво описывать одно и то же место в пространстве-времени, чтобы это были описания одной и той же системы.

Системы — это воплощения системы, физические объекты. По сути, говорится о том, что все системы::«физические объекты, которые мы выделили своим вниманием из их физического окружения» демонстрируют какое-то поведение, меняя состояние внешнего окружения целевым образом.

Системы представляют собой роли в окружении, которые выполняются конструктивными объектами, выбранными из материальных объектов-аффордансов и играющими роль этой системы в окружении, причём это происходит в каком-то месте. И неважно, какими это терминами выражено: слова могут использоваться самые разные. Так, термин «функция», обозначающий поведение рассматриваемой в качестве ролевого/функционального объекта/вещи в момент её использования/эксплуатации, может заменяться в случае системы-создателя с сильным (способным планировать свои действия в изменяющемся окружении) интеллектом словами метод/деятельность/инженерия/практика/культура/стиль/стратегия, да и тут возможны нюансы, связанные с выделением специализаций или классификаций (скажем, «вид труда», «вид инженерии», «субкультура»,

«подстиль», если хотят подчеркнуть узкий характер высказывания — и помним, что приставки «под-» и «над-», как и «мета-», могут отсылать к разным типам отношений, всегда уточняйте).

Итого мы ввели в предыдущем тексте подраздела следующие важные объекты внимания (**онтика функционального рассмотрения системы**):

- Система как физический объект.
- Окружение системы.
- Время использования/эксплуатации/функционирования системы.
- Роль (система в какой-то роли), функциональный/ролевой объект, которая или физический объект, или просто «физична» в момент эксплуатации системы — в зависимости от школы мысли, мы берём вариант, когда это физический объект, как в ISO 15926-2.
- Конструктивный/материальный объект времени создания, физический объект.
- Аффорданс как выбранный из окружающего мира подходящий материальный объект (возможно, сложно устроенный, например атомная электростанция как подходящая для производства электроэнергии), выполняющий роль.
- Функция/назначение/«поведение системы как роли» --это поведение по приведению окружения системы в ожидаемое состояние во время работы/использования/функционирования системы. Это общий термин для всех видов систем, но если агент «умный» (агент в узком смысле слова), то термин будет «метод» (и ещё десяток других — деятельность/труд/стиль/культура/практика работы и т.д.).
- Состояние системы (речь идёт о мелких «рабочих» изменениях состояния, которые не ведут к разрушению системы) и состояние окружения (в том числе ожидаемое состояние как результат выполнения функции системы) в момент использования/эксплуатации/«выполнения назначения».

Конечно, никакими определениями эти сложные отношения не выразишь, требуется довольно много текстов и примеров.

Повторим: разные школы инженерной и системой мысли имеют нюансы использования всех этих терминов, нюансы в определении понятий. Использование типов и заземление помогают разобраться.

Функция — это обычно поведение с акцентом на «назначение как ожидаемый результат поведения в момент использования: что там произошло с окружением системы», но функционирование — это «работа функции», акцент на время работы/использования и саму систему, производящую изменения в окружении. И тут ещё нужно следить, когда речь идёт об экземпляре системы, а когда о типе систем. Так, «функция самолёта — перевозка грузов и пассажиров по воздуху» — это высказывание не про конкретный самолёт, а про «множество бортов с разными серийными номерами, разных производителей», при этом общее высказывание про все экземпляры типа «самолёт» (борта с серийными номерами), а не про сам тип как абстрактный объект.

Можно погрязнуть в этой онтологической работе, но вы не сможете построить свои тексты на естественном языке, чтобы они были онтологически точны: вас перестанут понимать, текст будет как текст на языке программирования. Всё-таки в коммуникации с агентами на базе языковых моделей (людьми и AI-агентами) надо задействовать механизмы понимания естественного языка, включающие и S1 с её

прототипной теорией понятий, и S2 со строгой логикой, но не переусердствовать: общение на языках программирования и моделирования возможно, но очень трудно. Формализовать полностью нужно только то, что очень важно: это очень трудозатратно! В какой-то мере, это объясняет малую распространённость системного мышления: оно явно требует работать с типами, формализовать много больше обычного, поэтому трудоёмко даже при его натренированности. Но зато системное мышление даёт отличные результаты ровно за счёт хода на формализацию онтологии ситуации, формализацию системных описаний.

Поэтому в наших текстах мы будем по возможности соблюдать строгую типизацию, но не переходить окончательно на формальный математический язык системного моделирования. Так что вам нужно будет в любом случае как-то разбираться с неточностями в указании типов объектов и понимании типов объектов. Вам поможет заземление/grounding: всегда представляйте: намерение собеседника, в том числе намерение автора руководства/регламента/стандарта (что он вообще хочет выразить, о каких ситуациях говорит? Конкретных «экземплярах» или типах ситуаций?), а также пытайтесь делать конкретизацию с примерами из реальной жизни, то есть «заземлять» высказывания. И, конечно, если есть возможность, то переспрашивайте, уточняйте типы.

Если какой-то подраздел руководства оказывается для вас плохо понимаемым, то помните: никакие определения понятий и уточнения терминов не помогут, помогут многочисленные употребления терминов в самых разных ситуациях, это будет существенно уточнять понимание. С высокой вероятностью, вы получите примеры этих многочисленных употреблений терминов в самых разных ситуациях прямо в тексте нашего руководства, ибо изложение наших наставлений сознательно включает повторы изложения (принцип spaced repetition). Так что просто продолжайте осваивать работу по руководству дальше, не застревайте надолго, а затем ещё раз перепройдите всю цепочку руководств программы рабочего развития, это поможет больше, чем застревание на маленьком непонятном кусочке какого-то подраздела. Мы тут всё повторим, ещё не раз, в какой-то раз понимание обязательно появится! А ещё есть чат поддержки: https://t.me/systemsthinking_course, а ещё есть клуб: https://systemsworld.club/.

Продолжим разбираться с типами. Скажем, используемый (а не разрабатываемый и не изготавливаемый: время использования системы, а не время её создания) физический столярный молоток[150] в своём окружении обычно (но не всегда) играет роль гвоздезабивального устройства. Для простоты мы назовём это «забивало» или «забивало гвоздей», но иногда вы можете увидеть очень общие слова («устройство», «система», «инструмент») и далее указание на функцию («забивание гвоздей»). Так и получаются канцеляриты типа «гвоздезабивательной системы» или «гвоздезабивального устройства», «гвоздящего инструмента».

[150] https://dnipro-m.ua/ru/news/vidy-molotkov-i-ikh-naznachenie/

«Забивалом»::роль::вещь, проявляющее в момент использования функцию::поведение «забивание», может быть камень, может быть микроскоп — они вполне могут быть аффордансами для «забивало». «Забивало» это роль, роль могут играть разные актёры.

Чаще всего роль «забивала» играет (столярный или даже слесарный) молоток. Можно даже перепутать конструктивный/материальный объект «молоток» с его ролью «забивало» и назвать «молотком» саму роль, тогда конструктивный/материальный объект «микроскоп», (первичное назначение которого в его традиционной роли — помогать смотреть на мелкие предметы) будет обозван «молотком», если выбран в качестве аффорданса для названного «молотком» «забивала».

Эта путаница между конструктивом-в-роли и ролью-конструктива вызывает смешные шутки в ситуациях, когда все привыкли, что какой-то конструктив играет/воплощает какую-то роль, и вдруг надо поименовать роль — а все уже забыли даже название роли! «Кто у нас сегодня будет Васей» — говорят в ситуации, когда Вася долго играл, например, роль «крановщик» — и вдруг Вася не пришёл, а ищут не конструктив для «крановщика», а «конструктив для конструктива», Васи. Роль же просто «подразумевают». Это смешно, но уже не так смешно, когда вместо «молотка» ищут «другой молоток» и тем самым теряют возможность заменить один аффорданс на другой. Скажем, забивало, роль которого играет молоток, на забивало, роль которого играет нейлер (nailer, автоматический электрический гвоздезабиватель).

Будем всё-таки различать функциональное/ролевое именование системы по её функции в окружении и конструктивное/материальное имя системы как материального объекта «без предполагаемого окружения», которое только ещё

прикидывают в качестве аффорданса/подходяшки для его роли. Имён не жалко, у одного объекта может быть много имён. Главное — это понимать, что если два разных имени относятся к одному и тому же месту в пространстве-времени, то это один и тот же предмет. Если «микроскоп» и «гвоздезабивальное устройство» в какой-то момент относятся к одному и тому же месту в пространстве-времени, то это один и тот же предмет. Если «молоток» и «микроскоп» относятся в какой-то момент к одному и тому же месту в пространстве-времени, то это один и тот же предмет — и можете гадать, это микроскоп, которым колотят гвозди, или молоток, которым что-то пытаются рассмотреть.

Ролевое/функциональное имя системы всегда указывает на назначение/функцию как поведение системы в роли. Если роль системы заключается в забивании гвоздей, то имя ролевого объекта — «гвоздезабивальное устройство», «гвоздезабивальник» или что-то подобное. И для молотка тут рассмотрение вроде бы заканчивается, ибо обычно из контекста понятно, в чём состоит забивание гвоздя чем-то типа молотка. Но нет, «гвоздезабивальник»::система как ролевой объект может демонстрировать удивительно разное поведение:

- Вгонять гвоздь ударом бойка на ручном приводе, механически
- Вгонять гвоздь нейлером на электрической тяге,
- Вжимать гвоздь прессом,
- … придумайте ещё десяток вариантов.

Чтобы изменить способ/метод работы системы «гвоздезабивальник» (смотрите, сколько разных функциональных имён для роли системы тут появилось, более и менее канцеляритных, но каждый раз означающих функцию «забивание гвоздя») как простой физической системы, требуется пересобрать её из других конструктивных/материальных объектов в качестве аффордансов. Нейлер и молоток устроены совсем разным образом в части используемых конструктивов.

От целевой системы через метод работы к роли создателя, но не наоборот

У всяких простых систем, например, представленных конструктивом «молоток», крайне низкая **вменяемость**::«характеристика интеллекта» (подробней об этом будет в руководстве по интеллект-стеку программы исследовательского развития). У агентов-людей и AI интеллект посильней/универсальней и вменяемость много выше (их не нужно «аппаратно» пересобирать в части изменения конструктива, их можно научить какому-то поведению, дав объяснения — «перепрошить аппаратуру на новый алгоритм», а не «переделать аппаратуру по-новому»). А ещё такие агенты могут использовать разные аффордансы для включения в свой состав как системы, такие аффордансы для выполнения создателями каких-то методов работы по их алгоритмам/теориям в составе системы-создателя называют **инструментами/tools** (в случае измерительных инструментов — **instruments**).

Напомним, что у «агента» два значения:

1. Широкое — что угодно автономное, которое находится в неравновесном устойчивом состоянии (от атома и молекулы до кошки, от кошки до городской агломерации с парой миллионов жителей и многочисленных зданий и сооружений, набитых домашней утварью и промышленным оборудованием).
2. Узкое — это агент в широком значении, но только такой, который способен к планированию действий, приводящих к резкому ухудшению своего состояния

для того, чтобы затем улучшить своё состояние (скажем, потратить все ресурсы на плановое создание лука и стрел, чтобы потом успешней охотиться). Обычно этим словом «агент» называют людей и/или технические системы с достаточно сильным искусственным интеллектом (AI-агенты). И тут постоянные споры о том, достаточен ли интеллект AI-систем, чтобы называть их агентами в узком смысле, или они агенты только в широком смысле.

Дальше мы будем говорить главным образом об агентах в узком смысле: людях и AI. А поскольку речь сейчас пойдёт главным образом о терминологии, а не понятиях, то «сила интеллекта» тут будет даже не так важна. Вы сами поймёте, проще говорить о функции забивания гвоздей для молотка или культуре забивания гвоздей молотком. Скорее всего, в первом случае ролью будет молоток, а вот во втором случае ролью будет какой-то агент, в состав которого входит молоток. Но и для агента мы можем сказать, что его роль — забивание гвоздей (молотком ли, микроскопом или электрическим нейлером, это уж на его усмотрение). Как узнать, что говорить, как узнать, как понимать говоримое? Из языкового чутья. Просто помним, что «функция» для «агентов» чаще всего звучит как «метод/способ работы»/«вид труда/инженерии» и т.д. Но мышление про роль и конструктив — те же! Роль и её функция — «зачем этот агент нужен как система», конструктив — «какая система-агент как материал/конструктив взята аффордансом на роль».

Нужно всегда уточнять, какой именно «метод/способ работы» выбран для выполнения своей роли агентом с сильным интеллектом: в силу универсальности агента могут быть неожиданности. Когда мы говорим «неожиданности» — это означает обычно «неприятный сюрприз», расхождение с ожиданиями. Скажем, неопытный работник в роли плотника будет метод работы «скрепление досок» выполнять без различения шурупов, гвоздей, клея — в том числе, например, забивая шуруп молотком (и в первые секунды после выполнения этой работы даже будет казаться, что доски скреплены шурупом надёжно!). При этом микроскоп в качестве аффорданса для «гвоздезабивальника» использовать вряд ли кто будет, а вот людей и AI — запросто, даже если они не умеют выполнять работу по какому-то методу. Пациент/больной::роль на приёме у врача::роль:

— Здравствуйте, доктор::роль, у меня болит живот.

— А что вы вчера ели?

— Голубя ел.

— А вы голого голубя ели?

— Да, поймал, ощипал и голого голубя ел.

— А вы голову голубя ели?

— Да, голову голубя ел.

— А вы голую голову голубя ели?

— Ну, голую голову голубя ел. А в чём, собственно, дело?

— Хм—м—м... Честно говоря, я логопед::роль, но чем смог — помог!

Разобраться в ролях невозможно, если не рассматривать их функции::поведение. И повторим: поведение по способу добиваться изменений в окружающей среде для разных агентов будет называться очень по-разному: функции, практики, оргпроцессы, методы, способы, виды труда и т.д.

Если «тот, кого называют врач/доктор» (то есть агент, который взят как конструктив для выполнения функции/метода) имеет мастерство в логопедии::метод, а не гастроэнтерологии::метод, то результат ситуации будет примерно такой, как в анекдоте. Не забываем проставлять типы, при этом роли определяются их функцией, то есть поведением, ожидаемо меняющим состояние каких-то объектов в окружении (в случае ролей агентов — это метод/способ/практика работы, деятельность, культура), а не конструктивом/материалом (например, агентом, который рассматривается как «просто агент»). Роль Васи невозможно предсказать, если не знать, какие объекты окружения каким методом Вася меняет! **Без понимания, каким методом агент работает, в ролях не разобраться!**

Дополнительный источник путаницы с ролями — это моделирование подролей двумя вариантами иерархиями отношений, закодированных приставкой «под-»:

- По отношению композиции для роли. Скажем, логопед — это надроль для врача (ибо логопед выполняет в том числе и методы работы врача, такую же, как у любого врача, «все они изучали в вузе предмет «врачебное дело»», но вдобавок у логопеда есть метод работы по налаживанию речи, то есть общий состав/набор методов/практик/функций/способов работы логопеда больше, чем методов/практик работы врача, врач тут — подроль для логопеда. Это дальше удобно распространяется на мастерство агента: мастерство логопеда состоит из мастерства налаживать речь и общего мастерства врача, так что врач — подмастерство. В мастерстве те же отношения композиции, логопеда учить дольше, чем врача, врач тут будет подмастерством.
- По отношению род-вид. Если брать «называемого врачом» (то есть описываемого классификацией «врач»), то логопед представляется как «специализация врача». В этом представлении о мастерстве особо речь не идёт, врачи все имеют какие-то специализации, и логопед — просто часть врачей, у которых специализация по налаживанию речи. Тогда подроль — это вид, врач — это род, а по отношению специлизации логопед — подроль врача.

Мы рекомендуем опираться на линию рассуждений, связанную с функцией/«методом/практикой работы» ролевого объекта (роли), мастерством выполнения этой функции конструктивным объектом (агентом, например, человеком или даже организацией) — и поэтому считать «подроль» ролевым объектом, исполняющим составляющую метода его роли. Тут может быть большая путаница, ибо в методах (поведении) нет отношения часть-целое, поэтому аккуратно говорят о «составляющих метода», разложении метода на составляющие (по типу разложения функции в ряд, луча света в спектр — идея состава/композиции остаётся, но это не в чистом виде мереологическое то есть пространственно-временное рассмотрение, не «истинное» деление на части и целые). Подробно об этом будет в руководстве по методологии, и там же будет рассказано, что разложения метода на какие-то последовательные «шаги», в какой-то «стек» (последовательность укрупнения методов, каждый из которых надстраивается над каким-то составляющим его методом) или даже иерархию — это очень простые варианты, которые могут быть недостаточно точны для многих и многих инженерных рассмотрений. Методы описываются их теориями, эти теории могут быть представлены как алгоритмы, обобщённые с преобразований информации на самые разные физические преобразования объектов реального

мира. Алгоритмы же в общем случае плохо представимы в виде «стеков», «иерархий», «пошаговых процедур». Так, современная алгоритмика постепенно склоняется к функциональному программированию для выражения методов вычислений. При этом менеджерам надо как-то описывать культуру работы своих сотрудников, работникам культуры — описывать паттерны профильного культурного поведения людей в каких-то сообществах. Конечно, они не будут использовать знания по функциональному программированию, поэтому будут испытывать огромные трудности, ибо даже «пошаговые процедуры» очень плохо описывают то, что происходит в реальной жизни.

Мы всё-таки будем использовать «подроль» в качестве узла иерархии по отношению состава/композиции/«часть-целое» для мастерства исполнения метода этой роли. Это удобно для инженерии (например, планирования обучения).

В быту часто вы будете встречать и классификационные описания, в которых термин «под-» указывает на специализацию. Есть ещё один термин, который так же многозначен, это «мета-», у него множество самых разных значений по части каких-то уровней[151]. Мы его используем, говоря о мета-моделировании в значении «более высокий уровень объектов в иерархии по отношению классификации».

Для отношения композиции мы используем «под-» как в «подсистеме», а «над-» как в «надсистеме». В частности, подроль — это буквально будет «подсистема», а «надроль» — надсистема (системы обычно определяются по их ролям в надсистеме, подсистемы — по их ролям в системе, а роли — по функциям/методам/культурам::поведение). Иметь ли отдельный вид приставок для специализаций? Можно использовать «вид» для указания вида для рода, например «логопед вид врача, не путайте с гастроэнтерологом», «сальсеро вид танцора, другой вид — тангеро, не путайте», но классификаторы/таксономии более-менее произвольны в части выделения именно важных в рассмотрении объектов (но хороши для выделения самых разных, необязательно важных объектов[152]) и мы редко их будем использовать.

В жизни классификаторы/таксономии используются часто, в быту для них тоже используется модификаторы «под-» и «над-», что может вызывать путаницу. Как разобраться? Только по контексту: встретив «под-» и «над-», задавайтесь вопросом про то, какой тип отношений они обозначают? Какое там мастерство будет для выполнения практики — оно больше или меньше? Ещё один критерий: когда говорят о специализациях, чаще всего имеют в виду выбор взаимоисключающей роли — скажем, для роли врача это будут роли логопеда и гастроэнтеролога. В композиции/сборке по определению роли складываются, функции добавляются: функции гастроэнтеролога::роль и/или логопеда::роль складываются с функцией врача::роль. И у агента мастерство логопеда и/или гастроэнтеролога обязательно добавляется к мастерству врача (врач — это освоивший учебный предмет «врачебное дело», а логопед — ещё и какой-то другой учебный предмет, а гастроэнтеролог — третий. Поэтому «просто врач» может лечить, но конкретного ничего вылечить не может).

Ещё один выход — поступать как с методами, которые раскладываются на составляющие методы. Тогда мы не будем говорить «подроль», а будем говорить «составляющая роль», если это не специализация и «вид роли» если это

---

[151] https://ailev.livejournal.com/1053878.html

[152] https://ru.wikipedia.org/wiki/Классификация_животных_(Борхес)

специализация.

Если это иерархия специализации по отношению «род-вид», то ничего складывать/составлять не надо, надо только описывать — и можно обсуждать «выбор»: логопед::вид или гастроэнтеролог::вид для врача::род. В любом случае, обращайте внимание, когда работаете с подролями и надролями — имеется ли в виду система (и там будет отношение состава, всё физично) или вид систем (и там будут какие-то описания специализации, чаще всего показывающие контраст). «Тигр» и «мышь» не входят в состав «зверя», они просто «виды зверя»!

У Крылова в басне «Беда, коль пироги начнёт печи сапожник, а сапоги тачать пирожник» типичная метонимия. Давайте поставим типы, чтобы уточнить говоримое: «печение пирогов»::метод «пирожника»::роль, «тачание сапог»::метод «сапожника»::роль. Но у Крылова сапожник::«агент, имеющий мастерство сапожника», то есть сапожник не роль, а конструктив/материал для роли «пирожник», по недоразумению задействованный как аффорданс для «пирожника»::роль. То есть «сапожник» — не роль! То же с пирожником, который у Крылова не роль, а конструктив/материал для роли сапожника.

В реальных рабочих проектах будет ровно та же путаница между ролями и конструктивами: конструктив, который будет удачным аффордансом для одной роли, вполне может быть назначен на выполнение другой роли. А если разговор будет в терминологии путать роли и конструктивы (это, пожалуй, неизбежно!), то возникнут проблемы. Хорошим ходом тут будет переходить с языка ролей на язык функций (ну, или в случае людей и AI как создателей — язык методов/практик/культуры/стиля/способов работы, видов труда/деятельности/инженерии и т.д., синонимия тут очень обширна).

Впрочем, баснописец Крылов так и делает: он путает роли и конструктивы, но явно говорит о методах работы «печение пирогов» и «тачание сапог», указывая предмет метода работы роли (объект, меняющий состояние в ходе работы по методу: пироги, сапоги) и название действий по изменению предмета метода (печение, тачание). А кто производит эти действия с предметами? Роль, воплощённая конструктивом/материалом.

Функция/метод — это поведение воплощённой (когда она воплощена/реализована каким-то конструктивом/материалом, то есть во время использования/эксплуатации уже созданной в физическом мире) системы как паттерн изменения состояния каких-то объектов окружения (и эти изменяющие своё состояние объекты окружения иногда называют предметами метода). Для систем-создателей в их окружении мы находим целевые системы их методов создания (методов изменения состояния предметов в окружении, в конечном итоге приводящих к созданию и развитию систем). Логопедия изменяет состояние речевого мастерства, гастроэнтерология изменяет состояние желудочно-кишечного тракта, врачевание изменяет состояние организма. Ну, или логопедия::метод изменяет состояние «речевого мастерства»::«целевая система», «гастроэнтерология»::метод изменяет состояние «желудочно-кишечного тракта»::«целевая система», врачевание::метод изменяет состояние организма::«целевая система». Это всё понятно и без понятия «роли, выполняющей работу по методу». Эта роль нужна будет только тогда, когда вы озаботитесь агентом-конструктивом/материалом, который будет воплощать роль. Но если вы начнёте с агента в тот момент, когда неизвестен метод работы и его целевые объекты и действия, то вы автоматом окажетесь в ситуации логопеда

с головной болью или пирожника с сапогами: вы уже выбрали агента, а что будет делать агент — вам неизвестно. Скорее всего, мастерство агента будет не то, которое нужно для выполнения работы по необходимому методу, когда метод станет известен!

Поэтому в случае с бытовыми услугами нужно сначала разобраться, нужно тачание сапог или печение пирогов, а в случае врачей — нужно настраивать речь или устранять боли в желудке. **Нужно проходить по линии рассуждений «целевая система — методы её изменения — роль создателя — конструктив/материал создателя как аффорданс (например, агент с мастерством исполнения метода)», но не наоборот!**

Узнаём роль по её методу, а не метод по роли! Если кроме названия роли непонятно, что же именно она делает (то есть какие объекты каким методом переводит в какие состояния) — уточняйте! Роль без метода/практики — ничто! Метод/культура/стиль/стратегия/деятельность определяется по объектам, которые меняют своё состояние и действиям, которые выполняются с этими объектами для изменения их состояний!

Если мы рассматриваем не воплощаемые «умными агентами» системы-создатели и их методы создания как методы изменения состояния разных систем, которые затем на выходе дадут целевую систему, а более простой случай неодушевлённой целевой системы, изменяющей системы в своём окружении в ходе её эксплуатации, то суть рассуждения та же: **всегда от целевой системы как роли и её функции в окружении к подбору аффордансов/«подходяшек для роли» из каких-то конструктивов/материалов. И никогда не наоборот! Если вы не знаете, что делает ваша система в момент эксплуатации/использования, то вы не будете знать, из чего и как её делать в момент создания.** Поменялся только термин, «метод» на «функция» — суть рассуждения осталась той же самой, мышление об одушевлённых и неодушевлённых предметах осталось тем же самым.

Конечно, эти рассуждения про целевую систему и про системы создания целевой системы можно вытянуть в условную цепочку шагов (это не последовательность, не строгий алгоритм! Придётся делать предположения, проверять их, возвращаться, что-то уточнять, а также договаривать всех вокруг себя), для этого следовать «мантре системного мышления» как мнемонике (и не считайте, что это строгая последовательность шагов! Это описание метода, а шагами методы трудно описываются, это «логическая последовательность», в жизни же это будет выдвижение гипотез, проверки гипотез, возвраты назад в случае неудачи, попытки добыть недостающую информацию и т.д.):

- **сначала пойми роль и функцию целевой системы в окружении, «что она делает» (время эксплуатации/использования/функционирования целевой системы)**
- **потом пойми конструктив/материал, который будет аффордансом для воплощения роли, выполняющей функцию (время создания целевой системы)**
- **потом пойми методы изготовления конструктива/материала (методы изменения состояния конструктива системы из сырья в состояние готовой и работающей целевой системы)**

- **потом пойми, какие роли систем-создателей будут выполнять эти методы**
- **потом пойми, какие агенты как конструктив/материалы для ролей-создателей — люди, подразделения) будут выполнять эти методы создания целевых систем.**

И никогда логически не надо идти по этим «условным шагам» наоборот, ибо если не знаешь, что надо получить в итоге, то есть не знаешь, что и как должна делать система с её окружением, то не знаешь, какие материалы надо взять для изготовления системы и как их обрабатывать, а не знаешь, какие материалы брать и обрабатывать — не знаешь, кто с нужным мастерством и инструментами сможет взять эти материалы и обработать их так, чтобы получить успешную работающую целевую систему.

Эта «мантра системного мышления» — постоянно использующийся алгоритм, направляющий рассуждения системного мыслителя в определённом направлении: от целевой системы к создателям, а не наоборот. Используйте это как чеклист, проверяйте, все ли шаги в своём мышлении о рабочем проекте вы делаете согласно этой мантре.

Онтика изобретательства: конструктивы как аффордансы для ролей

Система физична: имеет назначение/функцию в изменении состоянии физичных объектов окружения (тоже систем). Никому не нужно, чтобы система меняла внутренние состояния, нужно, чтобы эти изменения внутренних состояний системы как-то влияли на состояние внешних систем. Никому не нужно, чтобы шестерёнки механических часов сами по себе крутились равномерно, но от часов требуется «показывание времени», в этом польза часов, это функция часов, роль конструктива «часы» — «времяпоказатель». А уж шестерёнками или электроникой это обеспечивается, или вообще Солнце используется как весьма грубые (точность до часа) и ненадёжные (возможна облачность) часы в конструктиве «солнечные часы» — это важно, но только после того, как мы разобрались, что там нужно в момент эксплуатации. А нужна функция «показывание времени» с такой-то точностью, такой-то надёжностью. Если это часы в ракете, то механические часы и солнечные часы как материальные/конструктивные объекты не подойдут, а вот электронные часы — могут подойти. Но пока мы не поглядели на «времяпоказыватель в окружении ракеты в момент взлёта», мы ничего не можем сказать про то, какие нам могут подойти конструктивы, об этом предыдущий подраздел.

Система всегда выполняет какую-то функцию во время её использования, то есть меняет состояние окружения. Система::роль и функция системы::роль определяются всегда в момент её работы/функционирования. А вот система::конструктив — это рассмотрение момента создания, когда мы прикидываем, какие материальные объекты какими способами будут обработаны, чтобы получить успешную систему::роль (грубо говоря, материал/конструктив шестерёнок и пружинок, электроники или палки и какой-то плоскости со шкалой и с размещением на открытой местности будут часами::конструктив/материалы). То, что мы выбрали из длинного ряда возможных конструктивов — это аффорданс, «подходяшка».

Система::роль может иметь имя, отличающееся от имени системы::конструктива, ибо объекты-конструктивы рассматриваются вне окружения во время использования, но во время создания. Чаще всего конструктивы/материалы, из которых создаётся система, называются по первичной функции этих конструктивов, когда подобный тип систем рассматривался как роль. Скажем, если интересует «пресс-папье»::роль с функцией «удерживание своим весом бумаг, разлетающихся на ветру», то конструктивы/«конструктивные объекты»/«материальные объекты»/материалы камень, микроскоп, молоток — все могут быть названы как аффордансы (то есть если они будут выбраны для удерживания бумаг, для выполнения функции) и «пресс-папье». От этого они не станут меньше камнем, микроскопом, молотком, но вот ролевое поведение/функция у них будет — удерживание бумаги от разлетания.

Функция системы::роль передаётся отглагольными существительными: «удерживание» (от глагола «удерживать»), в английском это -ing форма глаголов. В случае системы-создателя будет термин из синонимического ряда функция/практика/культура/метод работы, но это не меняет правил именования. Конечно, не все школы мысли придерживаются этого правила (могут и глагол использовать в инфинитиве), но всё-таки на сегодня такое именование функций — наиболее частое. И речь при именовании функции всегда идёт о времени задействования/работы/эксплуатации/использования/функционирования системы, когда она меняет состояние систем в своём окружении.

И это всегда время эксплуатации (время практикования системы создания — это время создания целевой системы, но время эксплуатации/использования/функционирования/практикования системы создания).

Без машинки типов в голове с этим трудно разобраться. Скажем, «функция инженера — задумывание, проектирование и воплощение системы». Фраза говорит о функции какого-то объекта — инженера. Если речь идёт о функции, то имеется в виду инженер::роль, инженер не в момент его создания (обучения агента, назначения агента на роль инженера), а в момент работы системы-инженера, когда инженер меняет состояние окружения, то есть какую-то целевую систему переводит из состояний «не задумана», «не спроектирована», «не воплощена» в состояние «задумана», «спроектирована», «воплощена».

Кто будет этим «инженером»? Какой-то «подходящий конструктивный объект»/аффорданс — это может оказаться Пётр Фёдорович, но может оказаться и ООО «Атомстроймонтажавтоматраздатка». Как так? А вот так: фраза говорит о функции, то есть прямо указывает на то, что рассматривается функциональный/ролевой объект «инженер» в его окружении. Фраза ничего не говорит про конструктивный/материальный объект, то есть не указывает на «подходяшку»/аффорданс, который будет затем «выполнять функцию»/«играть роль». Инженером тем самым может стать и AI-агент, «нежить», и целое предприятие.

Если вам сказано, что функция гвоздезабивального устройства — «забивание гвоздя», то есть гвоздь в окружении этого устройства переводится из состояния «не забит» в состояние «забит», то это неявное указание на тип: «гвоздезабивальное устройство»::роль, и речь идёт о времени использования этого устройства. Целевая система, меняющая состояние в итоге выполнения функции — гвоздь.

Но молоток, микроскоп или камень как аффорданс для «гвоздезабивального устройства» как системы создания «забитого гвоздя» будет играть роль этого устройства, или хитромудрый робот на квантовых кристаллах, так это неважно — это важно для момента создания гвоздезабивального устройства. Нужно разделить обсуждение функциональности (что надо сделать, назначение, функцию и роль) и конструктивности (из чего это делать), чтобы иметь возможность менять конструктивы, не меняя функцию. Если вам нужно «передвижение тонны груза из точки А в точку Б», то вы отдельно обсуждаете что там за груз, что там за точки старта и финиша, сколько этого груза — а потом только начинаете подбирать способ/метод/стиль передвижения, и совсем потом — того, кто сможет реализовать этот способ. Это основа **изобретательства** (подбор минимального набора аффордансов для выполнения функции). **Системное мышление тем самым в явном виде даёт онтику изобретательства.**

Если вы не знаете, какие аффордансы у вашей системы — не беритесь её делать! Charles Babbage и его помощница Ada Lovelace описали компьютер/computer: вычислительную машину, с функцией переводить исходные данные для каких-то математических задач в выходные данные. Функция — вычисления по изменяемому алгоритму. Но вот аффордансом они взяли механические конструктивы. Уже потом пробовали брать пневматические, электрические (реле) конструктивы, и тоже ничего не получилось: ужасающе низкая надёжность, ужасающе низкая скорость работы.

А что надо было делать? Брать в качестве конструктива (то есть аффорданса, выбранного изо всех других вариантов) радиолампы-триоды, которые были изобретены американцем Lee De Forest только через полвека после смерти Charles Babbage и Ada Lovelace, а изобретение использования триодов для компьютера произошло ещё на полвека позже. Функциональное описание компьютера (что там происходит во время его работы, чтобы получить нужный результат) было сделано, а вот с поиском аффордансов (изобретением!) были проблемы целый век, то есть два раза по полвека: изобретение усилительного триода — полвека от момента старта проекта универсального программируемого компьютера Babbage и Lovelace, затем изобретение триода и налаживание массового производства дешёвых усилительных триодов — полвека, и затем только задействование триода-конструктива в ключевом режиме «логического ключа» в составе компьютера, тем самым получили первый действительно универсальный и полезный вычислитель/computer ENIAC. И ещё несколько десятков лет пришлось ждать для замены ламповых триодов транзисторами, в том числе транзисторами на интегральных микросхемах. Только после этого компьютеры стали массовым продуктом.

Аффордансы как конструктивы/материалы обычно «приходят сбоку», то есть они являются материальными продуктами других отраслей. Клистрон как радиолампа для микроволновок пришёл на кухни совсем не из отрасли, которая производила примусы, керогазы и газовые горелки. Если вы не имеете идей о том, что может быть аффордансом для вашего описания системы (скажем, система «скатерть-самобранка», функция которой — кормление десятка человек за пять рублей), то вы рискуете оказаться в ситуации изобретателей компьютера, или изобретателей летательных аппаратов тяжелей воздуха, которые не получалось изобрести, пока не существовало лёгких двигателей внутреннего сгорания: братья Wright были недаром владельцами мастерской по изготовлению и починке мотоциклов.

Подробней все эти истории можно найти в материалах по стратегированию в руководстве по методологии. Но для понимания этого стратегирования надо иметь представление о системе::роль и её функции в момент эксплуатации, воплощающей эту систему::роль системе::конструктиве, который будет аффордансом во время создания системы::роль. И понимать, что **подбор аффорданса — это и есть изобретение.**

Моделирование: функциональный/ролевой объект (роль) и его функция, материальный/конструктивный объект как аффорданс

Заполните недостающие клетки не менее чем двумя вариантами, как в примерах из первых двух строк

Номер п/п	Функциональный/ ролевой объект	Функция	Конструктивный/материальный объект-аффорданс
Пример 1 (заполнено поле «конструктивный/материальный объект-аффорданс»)	телескоп	Увеличение изображения звёздного неба	телескоп в астрономической обсерватории; цифровой фотоаппарат с длиннофокусным объективом и дополнительной насадкой
Пример 2 (заполнены поля «функциональный/ролевой объект» и «Функция»)	Пресс-папье, камень в японском саду камней	Прижимание бумаги, заполнение пространства	камень
1.	Бумага	Отображение написанного	
2.	транскрайбер	Переведение устной речи в текст	
3.	Зеркало	Отзеркаливание изображений окружающего мира	
4.			Пенёк
5.			Серафим Петрович

6.			Компьютер

Физическое мышление в системном мышлении

В системном мышлении чаще всего (увы, но это так) используется «аристотелевская физика»: когда палец давит на стол, но стол не давит на палец, ибо палец-то живой и активный, а стол пассивный! В системном мышлении система как ролевой объект «играет свою роль»::поведение, то есть функционирует, выполняет свою функцию — но считается при этом активной и изменяющей мир вокруг себя без существенного изменения своих состояний (как минимум — без разрушения в ходе работы, устойчивыми/steady), а системы в окружении/environment считаются главным образом пассивными, претерпевающим изменения и существенно меняющими свои состояния, иногда вплоть до разрушения.

Изменения состояний в системе при выполнении ей своей функции рассматриваются потом, отдельно. Этот ход используется и в современной физике. Так, в constructor theory (теории создателей) предлагается определение создателя, как меняющего своё окружение неограниченное число раз без собственной деградации (и в пример приводят с одной стороны молекулу катализатора, которая производит катализ какой-то реакции, но после каждого акта катализа остаётся неизменной, а с другой стороны — самореплицирующегося робота, который остаётся после каждого акта репликации неизменным)[153].

Устойчивость каких-то систем в хаотичном окружении, которое с ними взаимодействует — предмет пристального изучения физиками. Атомы очень устойчивы, молекулы тоже устойчивы, а также устойчивы жизненные формы. Границы систем стараются проводить как раз по линии устойчивого существования, неразрушения в ходе использования. В физике это неравновесное устойчивое состояние, NESS (non-eqilibrium steady state[154], такое состоянию системы, в котором внешние силы поддерживают постоянное состояние дисбаланса, что приводит к свойствам, которые не меняются со временем, несмотря на то, что система находится вне равновесия), которое активно изучается в 21 веке: в отличие от равновесия, для поддержания которого не требуется какой-то энергии и работы, для поддержания неравновесного состояния участвуют энергия и работа, что в целом удерживает это состояние. Представьте себе канатоходца на канате, который удерживает равновесие под порывами ветра — это оно и есть. С одной стороны, много мелких изменений состояния целевой системы, а с другой стороны — в целом никакой катастрофы, резких изменений состояния и разрушения системы. Устойчивость ко внешним воздействиям заключается в постоянных изменениях, подстройке, как ни парадоксально это звучит. Хотя NESS в физике начинали изучать на простых системах, этот тезис подхватили химики[155], а затем и биологи[156], но речь идёт об общем описании каких-то систем, которые «балансируют» вокруг какого-то состояния, оставаясь при этом «почти теми же самыми» при довольно резких изменениях окружения.

Идея создателя/constructor в том, что он является универсальным

[153] https://www.constructortheory.org/
[154] https://www.sciencedirect.com/topics/physics-and-astronomy/nonequilibrium-steady-states
[155] https://pubs.acs.org/doi/10.1021/jp061858z
[156] https://physicallensonthecell.org/chemical-physics/non-equilibrium-steady-states

преобразователем материи/вещей примерно так же, как компьютер/вычислитель является универсальным вычислителем, если эквивалентен универсальной машине Тьюринга[157], только эти преобразования окружения согласно какому-то алгоритму/теории для универсального создателя ограничены не только вычислениями, но и изменениями физического мира — универсальность же в том, что потенциально возможны любые изменения (преобразования/transformations), допустимые законами физики. И универсальный создатель после каждого такого преобразования готов к следующему — он не теряет своей устойчивости, активно «балансирует» при взаимодействиях с окружением, «микрофлуктуирует» по сравнению с макрофлуктуациями, где теряется идентичность (например, происходит разрушение). В том числе «универсальный создатель» может, конечно, создать и новые инструменты как шаг метода, и даже копию себя («армия клонов»).

Человек (или достаточно развитый робот) в этой физической теории является как раз таким примером универсального создателя: он может добывать знания/теории/дисциплины/алгоритмы из своего взаимодействия с окружающей средой (вести исследования), создавать какие-то инструменты из окружающего его вещества и тем самым усиливать свои возможности преобразований мира, он может самореплицироваться (рожать детей).

Создатели из системного подхода — это чаще всего универсальные создатели, но все остальные системы (они сохраняют устойчивость, приводя к изменениям вокруг себя), и простой станок (он обрабатывает множество деталей согласно алгоритму, при этом оставаясь станком, он устойчив в поддержании своей формы в пространстве-времени, удерживает границу с окружающей средой, не перемешивается с ней) или компьютер (производит множество вычислений как физических процессов, удерживая свою форму, не перемешиваясь с окружающей средой), являются создателями. Но и человек, и AI-агент в роботе (выход в преобразования/transformation окружающего мира важен, нельзя ограничиваться только вычислениями/размышлением, то есть нельзя ограничиваться только работой с информацией) и даже крупная организация, выполняющая много лет разные проекты — это всё создатели, только в случае людей, AI-агентов и организаций это универсальные создатели: потенциально они могут делать любые (даже прямо на сейчас неизвестные) преобразования/transformations физического мира, если они не запрещены физическими законами.

Главным тем самым становится вопрос о том, возможно ли в принципе/теоретически по принципам физики какое-то действие, создание какой-то системы, перевод системы в какое-то состояние, или невозможно. Если возможно, то мы потенциально можем такое преобразование сделать, если только будем иметь соответствующие знания — то есть сможем разработать метод. А ресурсы для такого преобразования? Если будут знания, добудем и ресурсы!

Так что в системном мышлении используют слова о «взаимодействии» частей систем, но по факту каждый раз рассматривается не «взаимо», а просто «действие»: пара из какой-то активной системы-создателя и более-менее пассивного её окружения. Почему так?

Помним, что системное мышление — это про управление вниманием универсальных создателей, которые реализуют универсальные вычисления и универсальные преобразования (люди, AI-агенты, организованные/коллективные интеллекты). В

[157] https://www.youtube.com/watch?v=40CB12cj_aM

мышлении этих агентов сказать, что «всё со всем связано» и «всё со всем взаимодействует» и «всё на всё влияет и взаимовлияет» — это будет не очень правильно[158]. Полезней оказывается привлечь внимание к какому-то центральному объекту, заякорить на нём одну камеру внимания, а всё остальное потом рассматривать другими камерами по отношению к этому заякоренному объекту.

Если агентов много, то можно разделить общее мышление на нескольких агентов (или другими словами — собрать/федерировать мышление нескольких агентов в одно целостное рассуждение), если агенты общаются между собой, обмениваясь результатами измерений и размышлений[159]. Системное мышление как раз для такой работы, помогает федерировать (то есть объединять разнородные объекты) информацию разных агентов-создателей.

Моделирование — это прежде всего сжатие информации, модель содержит не всю информацию об объекте, а только самую важную о том, что моделируется для объекта. Так, модели механики содержат о физических телах информацию об их движении, но не цвет этих тел. Модели операционного менеджмента содержат информацию о скорости работы, но не одежде работников или языке общения работников. Системное моделирование, в основе которого лежит функциональное (времени функционирования/работы системы, рассматриваемой как роль, по изменению состояний её окружения) — это по сути ход на сжатие информации, выделение важного. Конечно, для разных агентов, участвующих в проекте создания какой-то целевой системы, важным представляется разное. Кого-то волнует вкус, кого-то цвет, кого-то (мы пишем «кого-то», ибо чаще всего агенты на сегодня — живые, но это не обязательно) мягкость, а кого-то — стоимость.

Целевая система как ролевой/функциональный объект, который играет роль/выполняет функцию (или для агентов — работает по методу) в своём окружении — это и есть ответ на это требование иметь центр внимания в ходе системного рассуждения. Если будет понятно, что будет делать система, то будет понятно и из чего её создавать. Если понятно, из чего её создавать, то будет понятно, как эту систему создать, «что надо делать». Если понятно, что делать, то будет понятно, кто такое сможет сделать и что для этого ему потребуется. В предыдущей фразе «ему» может отсылать и к **расширенному предприятию/extended enterprise** как создателю, то есть объединению разных предприятий какими-то контрактами, чтобы выполнить большой сложный проект. А поскольку предприятия созданы из универсальных создателей-людей и AI-агентов (прежде всего роботов, ибо нам нужен выход в изменения физического мира, 4E cognition), то одни и те же предприятия могут выпускать/создавать и развивать очень разные виды целевых систем в зависимости от доступности для них знаний/алгоритма/дисциплины метода того, как это делать (включая знания о том, как добыть на это необходимые ресурсы инструментария и оценить потребное время, а также сохраниться и не разрушиться в ходе работы).

Методы и работающие по ним системы (как ролевые объекты) очень часто рассматриваются ровно так: система-создатель::«ролевой объект» действует на материалы создаваемой системы (в конечном итоге эти материалы станут целевой системой и целевая система сама начнёт работать), и создаваемая система — это

---

[158] Смотрите концепт small-world networks и примеры в том числе non-small-world networks, https://en.wikipedia.org/wiki/Small-world_network

[159] https://www.sciencedirect.com/science/article/pii/S0149763423004694

просто другой ролевой объект. Обратным действием (целевой системы во время её создания на её создателей) пренебрегают, но если оно реально важно, просто проводят ещё одно рассмотрение, специально для учёта этого действия.

Если фреза режет деталь, то она тупится. Этим пренебрегают, рассматривают изменения детали, это важное! Ролевое действие фрезы: изменение формы детали (перевод состояния металлической детали из «заготовка в станке» в «деталь в станке»)! А потом отдельным рассмотрением говорят о том, что происходит с фрезой. Но уж тогда и деталь назовут «тупящей средой», а не деталью. Другая роль того же физического объекта — куска металла!

В какой-то мере это **панпсихизм** — удобный способ думать о неживых предметах так, как будто они живые. В панпсихизме, в котором кроме **анимизма** как веры в духов, живущих во всех предметах, есть и другие варианты, например, **минимальный физикализм**[160], в котором не делаются никаких предложений о существовании живых существ, кроме следования самым базовым законам физики (например, понимание того, что живые существа — это создатели/constructors, которые могут выполнять только те преобразования мира вокруг, которые разрешены законами физики, и ничего сверхъестественного). Мы будем придерживаться панпсихизма именно в версии минимального физикализма. И дальше разговор будет вестись не в терминах ньютоновской физики, а в терминах физики создателей, отчасти похожей на аристотелевскую физику, где делятся предметы-участники действия на «активные» и «пассивные», типа «молоток бьёт по гвоздю». У этого способа думать есть свои ограничения, но его нужно как минимум распознавать и понимать, о чём идёт речь в таких случаях, как понимать подобные описания.

Как ни странно, панпсихизм как способ думать про всё как в какой-то мере живое (но без анимизма! Никаких «духов вещей», никаких «чувств табуретки, на которой сидят»!) оказывается весьма современным. Одна молекула тем самым может вполне быть наблюдателем/witness/observer другой молекулы, то есть «измерять» её, взаимодействовать с ней, и тем самым решать проблему «наблюдателя» в физике.

Одна молекула — это явно ещё не живой «наблюдатель», но вполне «живой» в смысле панпсихизма в варианте минимального физикализма. В панпсихизме как минимальном физикализме сама «жизнь» определяется физическими принципами и законами, как возможность каких-то создателей самореплицироваться. Выход к универсальности создателя, то есть потенциальной возможностью выполнить любые трансформации не только информации, как в случае универсальной машины Тьюринга, но и вещества, ассоциируется с интеллектом и сознательностью.

Поскольку в физике обращаются к понятию «наблюдателя», давайте посмотрим на это как на систему-наблюдателя, то есть систему в роли наблюдателя, или «наблюдатель»::система::роль. Если «наблюдатель» — это роль, то должен быть и метод/культура наблюдения, причём начинать надо с метода, чтобы понять, что (или кто) может выполнять работу этим методом, а не наоборот, начинать с «наблюдателя» и приходить к непониманию, каким же методом он проведёт это наблюдение. Метод/способ/стиль наблюдения для роли наблюдателя может быть очень разным, и в физике по факту сводится сейчас к измерению как получению информации об окружении. То есть если рассмотреть взаимодействие молекулы с окружением как получение ей информации, это и есть «молекула-наблюдатель».

---

[160] https://academic.oup.com/nc/article/2021/2/niab013/6334115

Всё, никакого «человека-наблюдателя» не нужно и даже «наблюдающего робота». Достаточно взаимодействия рассматривать как факты измерения, то есть получения информации. Вся физика, которая занимается взаимодействиями объектов окружающего мира, становится тем самым информационной физикой, ибо всё становится выразимым в терминах информации.

Квантовые физики очень любят такой метод/способ/культуру мышления[161], и действующие в совсем другом масштабе астрофизики тоже придерживаются такого же способа рассуждений (у них и Земля в каком-то смысле слова «живая»[162]). Ну, и нужно обязательно сказать тут, что и все феномены «только живого, но не нежити» типа «сознание» тоже оказываются в таком варианте панпсихизма-минимального физикализма переинтерпретированы заново[163] и можно говорить в какой-то мере и о сознании бактерии (хотя и совсем небольшом), и о сознании колонии муравьёв[164], и даже о сознании Земли, как в только что упомянутой работе астрофизиков — и безо всякой эзотерики, исключительно в рамках законов физики!

Одним из примеров такого подхода неразличения живого и неживого (когда «функция» и «роль» задействуются для агентов в широком смысле, а не только узком смысле «интеллектуальных сознательных агентов») служит инженерный способ разработки **концепции использования** «сценарии использования»[165] . На английском это use case, но автор Ivar Jacobson оговаривал, что в шведском языке, на котором он в 1986 году предложил метод сценариев использования, вместо слова case/«судебное дело»/«история болезни»/«случай/вариант в практике» использовалось слово «сценарий».

Сценарий использования определяет взаимодействие между пытающимися достичь своих целей активными внешними **актёрами** (actors, акторы) и целевой системой. Актёры/акторы в сценариях использования (в других методах термин актёр/актор/actor может означать совсем другое! Тут мы обсуждаем только метод создания **концепции использования** системы при помощи сценариев использования) — это функциональные/ролевые объекты, которые необязательно реализуются людьми: они могут быть реализованы/воплощены в физическом мире людьми, компанией или просто оргзвеном, каким-то оборудованием, компьютерной программой, в том числе универсальной (AI). При анализе взаимодействия внешних актёров и системы как сценариев использования легко «выявлялись требования», и подход получил широкое признание в инженерии требований. Где-то с 2015-2017 годов требования перестали разрабатываться (подробней об этом будет в

---

[161] "Minimal physicalism as a scale-free substrate for cognition and consciousness", 2021, от Chris Fields, James F. Glazebrook and Michael Levin https://chrisfieldsresearch.com/min-phys-NC-2021.pdf и ещё пара работ про "нет, что вы, это не солипсизм! просто физики не понимали, чем они занимаются уже много лет!" из тамошней фразы "information can only be obtained by intervention, i.e. by actively posing a question (see Mermin 2018; Mueller 2020) for recent, somewhat philosophical discussions": "Making better sense of quantum mechanics", 2018, N. David Mermin, https://arxiv.org/abs/1809.01639 (это тот самый Мермин, который сказал про вариант интерпретации квантовой механики как "Shut up and calculate!", это ведь не Фейнман первым сказал, https://www.lassp.cornell.edu/mermin/), и ещё "Law without law: from observer states to physics via algorithmic information theory", 2017, Markus P. Mueller, https://arxiv.org/abs/1712.01826 (вот публикации автора — http://www.mpmueller.net/research.html).

[162] https://www.cambridge.org/core/journals/international-journal-of-astrobiology/article/intelligence-as-a-planetary-scale-process/5077C784D7FAC55F96072F7A7772C5E5

[163] Тут можно выделить теорию сознания, поддерживающую такой подход: IIT, integrated information theory, http://integratedinformationtheory.org/

[164] https://philpapers.org/archive/FRITAC-4.pdf

[165] https://en.wikipedia.org/wiki/Use_case

руководстве по системной инженерии), но для моделирования функционирования/использования/эксплуатации системы во внешнем для неё мире (окружении/environment) по-прежнему занимаются методом описания этого функционирования через сценарии использования.

Вот типичная диаграмма вариантов использования[166] (пример подчёркивает, что не все акторы/актёры на этих диаграммах — люди, а описание примера смотри в статье по приведённой ссылке):

Сценарии использования оказались очень удачны для описания работы/процессов/последовательностей_действий/сценариев/поведения системы и её частей. Этот способ описания стал повсеместным для инженеров, он приводит к построению функциональных/ролевых описаний. Обратите внимание, что смена состояния систем в концепции использования трудновыразима, но зато понятно «что на что воздействует» в ходе функционирования, каковы участники взаимодействий, что окружает целевую систему.

Функциональные и конструктивные рассмотрения одной и той же системы

Термин «функция», как мы обсуждали в первом разделе, имеет множество самых разных значений. Очень часто именно ролевое поведение/действие (поведение для какого-то назначения) называют функцией. Так, обычное словоупотребление — «функция молотка — забивание гвоздей». Если использовано отглагольное существительное (ing-овая форма в английском, отсылка к продолженности действия), а не глагол, то чаще всего имеется в виду функция/процесс/метод.

Работы (экземпляры процессов), которые рассматриваются как экземпляры выполнения «метода работы» («игры системы по её роли»), задействующие ресурсы/конструктивы, чаще выражаются глаголом в неопределённой форме или (если это план) в повелительном наклонении для ресурса, выполняющего работу. Работа (что надо конкретно с экземплярами объектов сделать в конкретный момент времени) будет запланирована как «забить гвоздь номер три в доску номер четыре не позже 16 часов», ожидается результатом «гвоздь номер четыре не позже 16

[166] http://www.uml-diagrams.org/examples/online-shopping-use-case-diagram-example.html

часов в доску номер четыре забит». Далее для работы можно обсуждать метод, по которому пойдёт работа. Указание на метод (что вообще надо делать) будет — «забивание гвоздя в доску» (собственно, метод/способ::поведение — это забивание, а предметы метода — гвоздь и доска), и опять-таки придётся обсуждать варианты того, как это сделать, просто названия метода обычно недостаточно (типов/паттернов поведений мало), его нужно обычно специализировать до какого-то однозначно понимаемого варианта (от рода перейти к виду — нехорошо сказать «зверь», ибо для рода «зверь» будут виды «тигр» и «мышь» одновременно). Скажем, «забивание» конкретизируется до «гвоздя в доску» (конкретизированы предметы метода), дальше можно конкретизировать «забивание гвоздя в доску автоматом для забивания гвоздей» — тут кроме знаний/теории/объяснений самого общего метода «забивания» и конкретизации его предметами «гвоздь» и «доска» в название метода попадает и инструментарий, который будет задействовать конкретная специализация метода «забивание».

Мы в системном мышлении **будем различать работы/действия по методам и методы/способы работы.** Методы упоминают типы ресурсов в момент задействования метода, а работы требуют экземпляров ресурсов, то есть материальных/конструктивных объектов, нужных для задействования метода. Поэтому работы в качестве «ресурсов работы» будут выполнять системы::конструктивы, а методы этих работ будут выполнять системы::роли (и можно обсуждать, включать или не включать в состав этих систем предметы работы метода, которые должны будут менять своё состояние, а также инструментарий, который расширяет/дополняет тело агента, выполняющего работу по методу, находясь в роли).

Отождествление функциональной и конструктивной систем с их работами и практиками будет как для «тёплого объекта» и «оранжевого объекта», а также «объекта на окне», которые просто описания одного и того же материального «чайника»: через 4D экстенсионализм.

Системы::роли (функциональные объекты как подтип физических объектов) интересны тем, что они могут исчезать из физического мира и снова появляться совершенно другими — в тот момент, когда у них появляется новый аффорданс как «исполнитель роли». «Колотилка» исчезает, когда мы перестаём колотить камнем, и вдруг появляется в виде микроскопа, когда мы начинаем колотить микроскопом.

Ещё раз повторим: системы как роли вполне физичны. Хотя некоторые философы и настаивают, что роли (ролевые/функциональные объекты) нужно считать абстрактными объектами, но инженеры и менеджеры прислушиваются к другим философам, которые указывают, что большинство людей считает **функциональные/ролевые объекты** существующими в тот момент, когда какие-то **материальные/конструктивные объекты** играют их роль, и не существующими, когда никакие объекты эти роли не играют.

Мы можем мыслить о Принце Гамлете, подразумевая, что он существует в тот момент, когда его роль играет один из актёров (например, известный артист Василий Пупкин). По Принцу Гамлету в этот момент можно условно «постучать», можно ткнуть в него (занимаемое им место в пространстве) пальцем, причём он занимает не только место в пространстве, но и место в пространстве-времени. Принц Гамлет как функциональный объект/роль как-то себя ведёт, то есть нужен в составе спектакля. А когда Принц Гамлет идёт обедать? Ответ: никогда, ибо обедать

идёт Василий Пупкин, а Принц Гамлет во время обеда не существует — он прекращает в этот момент своё существование, ибо «не играет роль»/«не выполняет функции»/«не функционирует»/«не используется». А если Вася Пупкин заболел? Тогда роль Принца Гамлета сыграет какой-то другой актёр.

Ровно то же рассуждение с насосом::конструктивом и нагнетателем::ролью. «Нагнетатель воды на третий этаж»::функциональный/ролевой объект существует, когда какой-то насос с серийным номером играет эту роль. Потом он исчезает из мира, когда этот насос ломается и его демонтируют, а когда его заменяет насос с другим серийным номером и он начинает работать, то функциональный объект появляется в мире вновь, но он реализован/воплощён совсем другими молекулами вещества. Помним, что «начинает работать» тут важно: все функциональные/ролевые рассмотрения — только во время эксплуатации/использования. И если вам кажется, что функциональное рассмотрение системы-создателя (какого-то слесаря, реализуемого то Петей, то Богданом) относится ко времени создания, то для слесаря как системы создания — это время его слесарного использования, время выполнения работы слесарными методами (выполнение слесарной функции/метода/практики/труда/культуры). Никакой путаницы: всегда помним в какое время рассматриваем целевую систему: если функциональное рассмотрение (пользы, результатов использования), то это функционирования/работы/использования, а если конструктивное (поиск аффордансов, обработка их с целью изменения состояний из сырья в готовую работающую систему), то время создания и развития. И да, могут быть длинные цепочки такого рассмотрения (создатели, создающие других создателей, и так на множество шагов до целевой системы — если вы создаёте игровые сессии, то кто-то из инженеров должен изготовить игровой софт в датацентре, а кто-то должен организовать инженеров, которые потом изготовят этот игровой софт, а кто-то должен каким-то образом создать организаторов инженеров, все эти цепочки в графах создателей в жизни довольно длинные).

Функция/«ролевое поведение»/практика/метод проведения работ назначается для работ системы, исходя из каких-то паттернов поведения систем при работах, то есть достижении результирующих изменений окружения. Так, даже печка меняет своё состояние с «холодная» на «горячая», то есть проводит работу «разогреться хорошенько на всю ночь 15 января 2054 года в комнате пять» только для того, чтобы вокруг неё стало тепло (то есть стало тепло в окружении/environment печки), а просто горячая печка сама по себе вне связи с окружением не нужна. Поэтому в качестве «печки» (функциональный объект) выбираем специализированный электробогреватель с хорошим конвекционным радиатором и достаточной мощностью (конструктив-аффорданс -тот конструктив, который выбрали), а не какой-нибудь точечный источник тепла, хорошо изолированный от помещения (например, мы не объявим печкой «термос с литром горячей воды»::конструктив). Дальше можно обсуждать, называть ли электрообогреватель, газовый обогреватель «печкой». «Печка» --это «роль» или «конструктив»? Поскольку функция первична, «обогрев», то скорее «печку на дровах» назовут сегодня «печным обогревателем», а если главное использование — готовка, то «печной кухонной плитой» (при всём понимании историчности слова «плита», указывающего не на функцию, а на конструктив «плоской поверхности»).

Во всех этих рассмотрениях нужно уходить от определений (прекращать спор о терминах, названиях и т.д., это просто «цепочки символов», в разных языках они

будут разные, это не так важно) и обсуждать вопрос, какое место в пространстве-времени занимают обсуждаемые предметы, «заземлять»/ground. Роль/ролевой объект/функциональный объект и играющий её конструктивный/материальный объект, конечно, будут занимать одно и то же место в пространстве-времени. При этом ролевой объект может исчезнуть из физического мира в любой момент, когда его роль закончится — когда конструктивный/материальный объект прекращает играть его роль. Материальный (то есть «объект из материи») объект просто так не уничтожишь — действует закон сохранения материи, материальный объект можно изменить, но он не может вдруг взять и исчезнуть, он просто меняет свои состояния и играемые им роли.

Например, я могу выделить в своей жизни «мою любимую игрушку»::роль. Дальше выбор из многочисленных конструктивов одного аффорданса/подходяшки как «моей любимой игрушки» происходит на основе того, какие я ожидаю результаты от выполнения этим конструктивом своей роли, какие изменения в своём окружении (то есть во мне) этот конструктив-в-роли будет вызывать. Итак, «любимая игрушка»::роль, аффордансами были для неё «плюшевый мишка»::конструктив в период 55 лет назад, «игрушечный самолётик»::конструктив в период 50 лет назад и «кофемашина»::конструктив сегодня. В промежутках, может быть, мне было не до игр, и этот ролевой/функциональный объект «моя любимая игрушка» в этот период вовсе не существовал. Физические индивиды конструктивов, играющие роль функционального объекта «моя любимая игрушка» несколько раз менялись, а вот функция «приоритетное участие в моих играх, развлечение меня» оставалась той же.

«Моя любимая игрушка» (интересно, а если снять выделяющие кавычки с термина — вы распознаете это как систему в роли? В жизни у ваших систем не будет таких подсказок!) в тот момент, когда она существует, вполне занимает место в пространстве — по ней можно постучать, её можно понюхать, о ней можно говорить как о физически существующем предмете. Но, конечно, вы будете стучать и нюхать конструктивные/материальные объекты, играющие эту роль в те или иные моменты. Если вы захотите понюхать Принца Гамлета, то вы унюхаете только запах Васи Пупкина, играющего его роль. Или запах Богдана, если играет Богдан. Или не унюхаете ничего, если Вася Пупкин или Богдан не будет играть эту роль — без исполнителя роли (конструктива!) не будет никакого Принца Гамлета. Но по факту запах Васи Пупкина или Богдана и будет запахом Принца Гамлета, как их внешний вид и будет внешним видом Принца Гамлета, как и их функция::«поведение в ходе игры по роли» будет последовательностью действий Принца Гамлета.

Зачем нужны ролевые/функциональные объекты? Чтобы отделить в мышлении выполнение каких-то действий (цель: изменить одно состояние целевой системы на другое) от объектов, выполняющих действия (системы-создатели с их инструментами). Действий (в случае агентов — методов работы) в мире не так много (можно сравнить число их типов условно с числом глаголов в языке — несколько десятков тысяч), а вот функциональных объектов, пригодных для действий (как и существительных, которых в языке сотни тысяч) — огромное разнообразие. Поэтому можно обсуждать метод/деятельность/практику/функцию работы «агронома» (роль инженера для растений::систем) или «менеджера» (роль инженера организаций::систем) и накапливать знания об этом методе/деятельности/практике/культуре/«виде труда»/инженерии — независимо от того, кто выполняет эту роль сейчас, Вася или «AI-агент Томас».

Точно так же можно выделить объект «водяной насос»::роль и его функцию «повышение давления воды» и дальше использовать или насос::конструктив с каким-то серийным номером, изготовленный Муромским заводом в этой роли (это будет аффорданс), или «водонапорную башню»::конструктив, которую придётся построить силами какого-то подрядчика или собственными силами (тогда аффордансом будет она). Рассуждение тут одно и то же про роли и экземпляры «родины», «дерева», «интернет-провайдера» и «водяного насоса». Все ролевые/функциональные рассуждения и подбор аффордансов (подбор аффорданса — изобретение! Надо ещё догадаться использовать усилительный триод в ключевом режиме для создания компьютера, мотоциклетный мотор как двигатель для самолёта!) устроены абсолютно одинаково. И подбор аффордансов многоуровневый (помним про системные уровни! Рассуждение повторяется на каждом уровне, «рекурсивно»). И это рассуждение одинаково и про людей, и про животных, и про системы искусственного интеллекта («нежить»), и про совсем уж неживые системы.

Вопросы. Конструктивные и функциональные объекты

Любимой игрушкой Пети с года до шести лет был плюшевый мишка Винни. Сейчас Пете 16 лет, и его любимая игрушка — игра WarCraft, которая стала любимой с того момента, как диск с ней родители подарили ему на пятнадцатилетие.

Выберите верные утверждения:

+++ Любимая игрушка Пети — функциональный объект ### Да, это функциональный/ролевой объект, имя «любимая игрушка» указывает на роль, тут просто не упомянуто, какой конструктивный/материальный объект-аффорданс играет его роль в момент использования.

+++ Любимая игрушка Пети — роль ### Да, это функциональный/ролевой объект, имя «любимая игрушка» — имя роли, просто не упомянуто, какой аффорданс играет эту роль в момент использования.

+++ Петин плюшевый мишка Винни — конструктив ### Да, это физический объект-конструктив, он занимает место в пространстве-времени (хотя это время может быть в прошлом, но у нас 4D, это не меняет рассуждения). Мы не знаем, какую роль играет этот объект, но мы можем его найти в окружающем мире, выделить вниманием.

--- Петин плюшевый мишка Винни — функциональный объект ### Нет, это конструктив, а не роль, пока непонятно назначение его поведения в окружении в момент использования. Он, например, может быть «любимой игрушкой»::роль. А может быть и «забытой игрушкой»::роль. Может быть не игрушкой, а «украшением интерьера»::роль.

+++ Петин диск с компьютерной игрой Warcraft — физический объект ### Да, это физический объект, можно указать место в пространстве-времени, которое этот объект занимает.

+++ Компьютерная игра Warcraft, в которую играет Петя в свой день рождения — аффорданс ### Да, в состав этой игры как физического объекта входит всё, что необходимо для игры — прежде всего

компьютер и экран, меняющие свои состояния во времени, и именно в этот четырёхмерный объект-игру играет Петя, а не в другой четырехмерный объект, который мог бы выбрать для игры Петя.

+++ Записанная на диске игра Warcraft — это документация с описанием игры Warcraft, в которую будет играть Петя в свой день рождения. ### Да, записанный на диске код компьютерной программы — это описание, оно как-то (хотя и не на 100% однозначно и не во всех подробностях) описывает физический объект, а именно игру, в которую играет Петя в свой день рождения.

Главный конструктор НПО Автоматика Леонид Шалимов, ответственный за пуск с космодрома Восточный, уволен после инцидента с «Союзом».
Выберите верные утверждения:

+++ «Главный конструктор НПО Автоматика» представляет собой четырехмерный функциональный объект; ### Да, речь идёт о функциональном объекте, роль которого исполняет в какой-то момент конструктив-аффорданс Леонид Шалимов как «подходяшка» конструктива для роли главного конструктора.

--- «Главный конструктор НПО Автоматика» представляет собой свойство четырехмерного объекта Леонид Шалимов; ### Нет, это не свойство физического объекта, это физический функциональный объект: роль конструктива/материала Леонид Шалимов.

--- Леонид Шалимов представляет собой трехмерный объект и является частью четырехмерного функционального объекта «Главный конструктор НПО Автоматика»; ### Нет, Леонид Шалимов существует во времени, как и любые другие физические объекты, то есть он четырёхмерный объект

+++ На протяжении некоторого времени до инцидента с «Союзом» материальный объект Леонид Шалимов и функциональный объект «главный конструктор НПО Автоматика» совпадали. ### Да, поскольку занимаемое ими место в пространстве-времени совпадало, то это был один и тот-же физический объект.

Производство систем как конструктивов в их потенциальных ролях

Даже высокоинтеллектуальные системы-создатели обычно не назначают функции произвольного вида сами себе, они берут функции «подсмотренными» из культуры — это же паттерны поведения, поэтому важна повторяемость (паттерн наблюдается на ряде поведений, это же «шаблон»). Так что метод/способ/практика/функция отражает в каком-то смысле «одинаковость»/шаблонность/похожесть, следование одному стилю/культуре/методу поведений. И после этого можно говорить примерно то же о роли. А дальше или в языке такой термин для функции как поведения уже есть (обогрев — обогреватель), или нет, тогда его надо придумать (сепуление — сепулька).

Не молоток назначил себе роль забивателя гвоздей, не молоток имеет функцию забивания гвоздей. Мы можем взять микроскоп и назначить его «молотком» — забивать им гвозди. «Молоток» при этом — не более чем термин роли для

микроскопа::конструктив (или камня::конструктив, или даже молотка::конструктив), название не должно путать. Поведение микроскопа в так названной роли, его функция, «игра по роли забивала» — забивание гвоздей. Если «забивание гвоздей» относится не к микроскопу, а к Васе-с-микроскопом, то «забивание гвоздей» часто будет обозвано не «функцией», а «методом/практикой/способом/культурой работы», «видом труда»/деятельностью»/«вариантом/видом инженерии». С функциями/методами и выполняющими их ролями возможны самые разные «почти синонимы» даже в названиях типов, вы уже должны были их заметить, мы старались в тексте руководства использовать множество синонимических рядов для культуры/стиля/«вида труда»/способа/метода работы (но не самой работы! В руководстве по методологии будет объяснено отличие: работы подразумевают ресурсное рассмотрение «кто когда над чем работает», а методы/способы работы — содержательное, «как это всё сделать, неважно когда и с какими именно экземплярами объектов»).

Можно сказать «культура забивания гвоздей», почему бы и нет — хотя про забивание гвоздей так не говорят, а будут обсуждать «метод работы». А вот какое-нибудь «танцевание танго» легко могут назвать культурой или стилем танцевания, но редко «методом работы» или даже «видом труда». Главное понимать, что «танцевание танго» и «забивание гвоздей» — это объекты одного типа: функция/метод/культура рабочего поведения танцора и «функция/метод/культура» рабочего поведения плотника). Синонимию функций забивания гвоздей для плотника-с-инструментом и молотка-или-микроскопа-без-плотника (инструмент и агент-с-инструментом вроде как выполняют одну и ту же функцию, но по факту их роли и функции разнятся) — такую синонимию тоже придётся как-то убирать. В больших проектах с именованием тысяч и тысяч ролей и функций/методов может возникнуть огромная путаница, тем более что молоток или микроскоп могут сейчас оказаться с искусственным интеллектом, и это будет ещё больше путать с пониманием чужих и выбором своих терминов.

Почему важно брать культурно-обусловленные термины, а не выдумывать их самостоятельно по случаю? Например, Васю в роли инженера называть инженером, а не «улучшателем жизни» или «создателем успешных систем»? Прежде всего это вопрос понимания: вам не надо будет подробно рассказывать, чем занимается Вася как инженер, вы можете отослать за описаниями в культуру (то есть предложить «погуглить» или спросить AI-ассистента). Интересно, что при этом могут всплыть и какие-то учебники (или регламенты, инструкции, международные и отраслевые стандарты) по методам работы, о которых вы не знали. Почитайте их, наши инженеры-менеджеры с удивлением узнавали, что многие их рабочие проблемы описаны в руководствах программы рабочего развития, равно как описаны способы решения этих проблем — просто они не догадывались о том, что их работа не уникальна, многие люди на планете занимались и занимаются этой работой, поэтому есть паттерны (методы/практики/культуры/стили) работы, которым можно следовать, чтобы работать в какой-то роли с меньшими проблемами. Решать задачи (когда понятно, что делать, и нужно только собрать ресурсы и выполнить работу по известному методу) много проще, чем преодолевать проблемы (когда вообще не понятно, что происходит и что делать — но дальше работать почему-то не получается).

Так что старайтесь не изобретать терминологические велосипеды и не изобретать

самостоятельно методы работы и не переопределять культурно-обусловленные роли. Конечно, всё что в учебниках/инструкциях/стандартах, надо адаптировать к рабочим ситуациям, но всё-таки лучше доезжать до реальных проблем по рельсам уже совершённых кем-то объяснений для тех или иных методов работы, используемых какими-то уже существующими ролями, перед тем как адаптировать их. Ну, или готовьтесь пожинать плоды новичковых ошибок. Вы, конечно, можете затратить огромное количество времени на обучение игре на диджейском пульте путём проб и ошибок, но пойти в школу диджеев или хотя бы пройти курс диджея по самоучителю — это в разы и разы более короткий путь. К методам работы менеджмента и роли менеджеров это рассуждение так же относится, хотя понимается почему-то хуже. «Ты умеешь играть на скрипке? Нет, но думаю, что получится» — этот анекдот почему-то в менеджменте оказывается не анекдотом, а основным путём рассуждений. Квалифицированный агент, который имеет мастерство инженера-теплотехника или даже инженера личности, занимавшегося проблемами HR, идёт заниматься операционным менеджментом — не зная ничего об этой культуре, хотя на эту тему есть немало учебников. Ну, и результат — огромное число менеджерских ошибок, «изобретение кривых велосипедов». Поэтому запоминаем: **за каждым методом/практикой/культурой работы, деятельностью/видом труда/инженерии есть учебник (часто в виде руководства, регламента, инструкции, корпоративного стандарта, онлайн-курса, и т.д.), и подписываясь на какую-то роль в проекте — вы должны хотя бы полистать такой учебник для вашего метода** (если только вы не работаете на таком фронтире, где этот метод надо создать, тогда вы сами будете такой учебник писать для тех, кто пойдёт по вашим следам, например, ваших сотрудников).

Если выбрана терминология с «функцией», то функция выполняется ролью/ «функциональным объектом»/«физическим объектом как ролевым». Или, что то же самое, просто другими словами: ролевое поведение выполняется деятельностной ролью (метод/практика/функция — это «ролевое поведение», вы обязаны разглядеть в «ролевом поведении» упоминание метода/способа/практики/культуры/стиля/способа работы). Или функциональный «объект» вдруг назовётся функциональным «элементом», при этом игнорируется тот факт, что «элемент» означает что-то неразделимое дальше на части. Или в организации не будут использовать термины из учебника/стандарта метода «управление конфигурацией для машиностроительных производств», но зато для своих изделий они будут говорить о «рабочем процессе комплектации» — и это окажется тем же самым, что «управление конфигурацией»::метод, только какие-то проблемы из учебника/стандарта «управления конфигурацией» оказываются не замеченными, и терминология вся сплошь придумана на месте, а не взята из литературы. Конечно, при этом можно ожидать проблем: хорошо бы понять, откуда там взялись несоответствия методу из учебника и что делать с проблемами, которые пока никто не заметил, но которые обязательно скоро себя проявят — и их бы лучше предотвратить.

Слова-термины важны и не важны! Просто каждый раз пытайтесь разобраться, какого типа понятие, обозначаемое термином — и из какой дисциплины/теории/знаний какого метода это понятие, какой вариант/вид общего метода пользуется именно этим термином.

Знания/объяснения передаются из ситуации в ситуацию в виде описания

в учебниках, корпоративных регламентах, промышленных стандартах «нормативных методов работы»/«норм культуры»/«норм поведения»::«предписанные паттерны поведения» для систем::ролей/«функциональных объектов» (время работы системы в её роли), а не норм поведения для разных конструктивных/материальных объектов, выбранных аффордансами для выполнения роли (время создания и развития системы, время работы создателей).

Но если вы производитель конструктивов и ожидаете, что они будут как-то задействованы (то есть их используют как аффорданс для каких-то систем::роль), как назвать эти объекты? Скажем, вы выпускаете насос. А заказчик выбирает из разных аффордансов изменения напора воды: купить ваш насос или использовать водонапорную башню и трубы (и тогда насос в башню будет совсем другим)?

В случае производства каких-то «потенциальных аффордансов» вы можете использовать приём, когда целевые системы называются ролью по их основному методу работы, если это «одушевлённый агент», или основной функции, если это что-то не слишком разумное (для насоса — название по функции «насасывания», компрессора — по функции «сжатие/компрессирование газа», и т.д.). Системы прежде всего рассматриваются как ролевые/функциональные объекты в тот момент времени, когда они выполняют свою роль, то есть готовы и работают/функционируют/используются, приносят пользу, изменяя в ходе работы своё рабочее/operation окружение/environment.

У заказчика ваш «насос»::«основная ожидаемая роль» запросто может быть назван «нагнетателем антифриза во втором контуре охлаждения»::роль, если ролевое его назначение/функция — нагнетание антифриза во втором контуре. Завод насосов продаст «насос», а вот купивший насос инженер (тоже может быть предприятием! «инженер» — это роль, которую может играть предприятие в целом!) — переобзовёт его кратко «нагнетатель», и это нормально. **Системное мышление позволяет не запутаться с разными наименованиями одной и той же системы, причём у производителей конструктивов подсистем и создателей целевых систем эти именования различаются, это типовая ситуация.**

Вне контекста использования инженерных способностей Маша — это просто Маша::«агент в узком смысле»/«почти универсальный создатель». Маша много кем может быть, как и микроскоп, который может мелкие предметы показывать, но можно им орехи колоть, гвозди забивать, использовать как пресс-папье. Маша может много чего, много больше микроскопа! Но в момент выполнения функции инженера («Маша-конструктив играет роль инженера»), Машу будут называть по роли —«инженер». И это нормально, хотя мы бы для таких случаев сходу требовали уточнения (инженер по холодильным установкам, инженер-электронщик, инженер-программист — это всё-таки очень разные инженеры. И нужно обязательно уточнять метод, которой занимается Маша как инженер: какой именно это вид инженерии, чем уже можно определить специализацию, тем лучше).

Системы именуются обычно по их ролям, определяемым первичным их назначением, то есть по их чаще всего выполняемой функции, тому, что они делают в окружающем их мире. Представьте, что целевой системы нет — что не произойдёт важного, если убрать её из мира, если она не будет работать? Какие роли в текущей культуре обычно выполняют вот

это «важное действие, невозможное без убранной из мира системы»? Вот так систему и назовите.

Когда мы именуем микроскоп, то прежде всего имеем в виду то, что он позволяет «мелко смотреть» в тот момент, когда он полностью изготовлен и работает. Если бы мы считали, что микроскопом нужно главным образом что-то колотить (орехи, например, колоть), назвали бы его «колотилка».

Как всегда в языке, самые древние названия имеют неясное происхождение и часто указывают не на роль/функцию, а на форму (молоток, маленький молот — но вот сам молот — это от метода молотьбы/обмолота[167] битьём, в основе глагол «молотить», то есть «бить») или что-то другое. Но если мы разрабатываем системы, или хотим понять что-то про системы, то в названии правильно искать не конструктивный/материальный объект, представляющий систему, а роль — и указание на функцию/действия этой роли.

Ну, а если системой вдруг является человек или AI-агент или тем более организация из людей, AI-агентов и самого разного обрудования, то обычно мало что можно сходу сказать о назначении/функции такой системы в её окружении, ибо с интеллектуальными агентами как универсальными создателями обычно всё сложно. И с ролевым поведением/функциями людей и универсальных AI-агентов придётся всегда разбираться отдельно и специально. Но **мышление про людей-или-AI-в-роли и какой-нибудь микроскоп-в-роли будет устроено одинаково!**

Мантра системного мышления

Традиционно поднимается вопрос: не начинать ли сразу с ролей? Мы ведь писали, что люди трудно думают о поведении, то есть трудно думают о функциях/практиках/культурах/методах работы. Кажется, что поэтому сначала надо говорить про роли (это ведь легче), а затем обсуждать их методы работы (это труднее)! Но «сначала роли, затем что они делают» — это оказывается в проектах неправильным. Роли появляются в рассуждении много позже, и мы лишний раз об этом повторим (эту последовательность мышления нужно повторять, повторять, повторять: это **«мантра системного мышления»**):

- **Что надо изменить в окружении целевой системы.** Сначала надо понять, какие изменения в окружении целевой системы нужны. Вот если убрать целевую систему из мира, поломать её — от чего все будут плакать? Если не заметят, то немедленно прекращайте проект создания и развития системы, она не нужна, она ничего не меняет в окружающем мире в момент её использования (то, что её создатели не получат зарплаты — это отговорка. Они не принесут непоправимой пользы созданием системы, мир не изменится к лучшему от создания системы, поэтому создателям не меняющей окружение системы платить не надо, разве в порядке благотворительности. Если это не явная и осознанная благотворительность, то требование создателями ресурсов для ненужной никому кроме них работы — наглость и мошенничество).
- **Каким способом/методом меняем окружение целевой системы.** Затем надо выбрать способ, каким система изменяет этот мир, функцию/метод работы системы. Это рассмотрение поведения, которое должно случиться в момент использования системы с какими-то ролевыми

[167] https://ru.wikipedia.org/wiki/Молотьба

объектами, чтобы случились изменения состояния окружения, определённые в предыдущем пункте. Вот тут разные расчёты, относящиеся к моменту использования (что там в момент использования течёт, какая энергия нужна, сколько времени это всё будет занимать и т.д.).

- **Какова роль целевой системы.** И только тут вы можете сказать, какое в культуре принято имя для целевых систем, выполняющих подобные функции — сказать роль целевой системы. **Роль целевой системы только тут!**
- **Из чего будем делать целевую систему.** Затем надо выбрать аффордансы для целевой системы (а поскольку в системах много системных уровней, то часто это сводится к выбору аффордансов для хорошо выявленных подсистем). Это изобретательская задача, чистое творчество.
- **Какими методами работы делаем целевую систему.** Затем надо понять, какими методами надо работать с этими конструктивами/материалами, чтобы целевая система смогла нормально функционировать. Нужно использовать сварку, обучение, лазерную резку, командообразование?
- **Какова роль создателя.** После выявления того, что там надо и как делать, надо задаться вопросом про роль создателя, который это будет делать с конструктивами-аффордансами, превращая их в конечном итоге в функционирующую/используемую систему. Выбрать культурно-обусловленное имя роли, посмотреть учебник для агента (или хотя бы статьи, посты специалистов по методам работы роли), который будет конструктивом-аффордансом для роли. **Роль создателя только на этом шаге!**
- **Кто/что будет создателем.** Только после этого берём конструктивы для потенциальных создателей (людей, оборудование, AI-агентов, предприятия) и выбираем из них аффордансы/подходяшки, которые сумеют выполнить роли из предыдущего пункта. Ибо надо всегда опасаться выбрать пирожника из умелых сапожников.

Конечно, хождение по этим шагам происходит не как «водопад/каскад», который всегда падает вниз, и никогда не происходит возвратов назад. Нет, на каждом шаге высказываются догадки, которые критикуются, с ними проводятся эксперименты (обычный ход познания!), и в случае опровержения этих догадок приходится возвращаться на предыдущие шаги, часто даже к самому началу: пониманию того, зачем вообще нужна создаваемая и развиваемая система. Но логическая последовательность шагов — та, что написана. Конечно, в жизни приходится решать и обратные задачи, «вот тут у нас люди Петя и Вася, которые умеют разжигать огонь трением. Как бы мы могли их приспособить к делу, чтобы они хотя бы прокормили сегодня сами себя?». Или «вот тут наши учёные изобрели сепульку. Куда бы её воткнуть, чтобы нам заплатили за её массовое производство?». Но это не прямые, а обратные задачи — их решения интересны, но лучше всё-таки решать прямую задачу в каждом проекте.

Ещё одна трудность — это то, что создатели могут «создавать создателей» (скажем, люди-основатели предприятия могут создавать предприятие, а уже предприятие будет делать целевую систему). Но это просто последовательное задействование этих рассуждений для каких-то путей на графе создателей.

Основная проблема тут в том, что этой логической последовательности шагов системного рассуждения редко придерживаются в рабочих проектах. Большинство

наших инженеров-менеджеров раньше демонстрировали знание этого рассуждения при отвязанном от рабочего проекта разговоре, но также демонстрировали неприменение его в собственных реальных проектах — приходилось проводить дополнительный тренинг (повторение, повторение, повторение рассуждения для реального проекта, обучение нейронной сетки в мозгу инженера-менеджера многократным повторением паттерна), чтобы это рассуждение не осталось «рассуждением из учебника», а стало целенаправленным методом мышления, осознанно и постоянно используемым в рабочих проектах. Сейчас ситуация стала лучше, но на всякий случай подумайте о себе: вы реально следуете наставлениям нашего руководства в работе, воспринимаете его как регламент мышления, или же как «учебник» и поэтому разделяете обучение и работу? Как часто вы реально идёте по шагам рассуждения из мантры системного мышления?

Задание по следованию «мантре системного мышления»

Проставьте отметку о выполнении:

1. Выполните для своего проекта пункты мантры системного мышления, взяв какое-то одно звено (создатель-создаваемая система) в цепочке создания от вас-создателя до целевой системы (отдельно пропишите этот путь на графе создателей от вас-создателя до целевой системы и выберите пару смежных узлов графа, где одна система создаёт другую). Напишите пост с рассуждениями по пунктам мантры для этого звена. Отметьте в посте, о чём пришлось подумать впервые (то есть не задумывались об этом раньше), если не пропускать шаги рассуждения. Что надо поменять в проекте, чтобы использовать результаты прохождения этого системного рассуждения по системной мантре? Что и кому надо предпринять, чтобы повысить шансы успешности проекта? В частности, что надо предпринять лично вам?

Второе и третье поколения системного подхода

Много лет в системном подходе считалось, что системы как бы «объективны» в плане их назначения. Скажем, самолёт — всем же понятно, что это за система, какое её назначение/функция! Или радиолокатор — никаких разногласий, зачем он нужен и какое у него окружение! Или даже лабораторная мышь, которую изучают биологи — её назначение/функцию не нужно обсуждать специально, правда ведь?!

Системный подход подавался как метод, которым в этой «объективно имеющейся системе» можно отмоделировать самое важное — которое тоже представлялось очевидным, разве что нужно было немного эту очевидность уточнить и выразить численно. Ничего субъективного, «чистая наука», вполне поддающаяся формализации. Учебники системного подхода в его первом поколении легко было распознать по обилию в них математики. Если в учебнике системного мышления/подхода много формул — это верный признак, что это изложение системного мышления не позже 70-х годов 20 века. И чем больше биологии и всякой «самоорганизации», тем более ранняя это книжка.

Но в конце семидесятых годов прошлого века системные мыслители обратили внимание, что инженерными, а не простыми физическими и биологическими системами всегда занимаются в плане их создания и развития какие-то создатели-люди. Это было связано главным образом с тем, что в мире перешли от изучения самих по себе растущих (биологических, «естественных») систем к системной инженерии — и тамошние радиолокаторы и самолёты не росли сами по себе в лесу

или поле, их приходилось кому-то делать, каким-то другим системам-создателям. И делать эти системы их создателям приходилось для чего-то, что нужно было ещё кому-то (заказчикам), то есть тоже для людей (никаких AI-агентов в этот момент, разумеется, не было). Чем больше был инженерный проект, чем больше он включал людей, тем больше об этом «для чего нужно делать систему» нужно было договариваться, учитывая самые разные интересы людей, «объективность» понимания назначения системы перестала быть очевидной. Назначение системы оказалось плодом договорённостей множества людей.

Люди (и в одиночку, и в командах, и предприятиями, и расширенными предприятиями — то есть множеством предприятий, объединённых в рамках выполнения какого-то крупного контракта создания и развития целевой системы, например, атомной электростанции) задавали системам роли в каких-то окружениях (ещё более больших системах), уточняли функции, которые эти системы должны были выполнять в своём окружении. Нет создающих систему создателей-людей — нет системы, а есть какой-то «просто объект», непонятно откуда взявшийся (ибо его никто не задавал, никто на него не обратил внимания, никому он не нужен для его деятельности, никто не может уточнить его границы, и любые фантазийные описания такого объекта нельзя обсуждать).

К системам, которые никого не интересуют, системное мышление не применишь — просто некому будет его применять, никто ж не заинтересован! В проектах инженерии «железной», программной, обучения и образования (инженерии личности), менеджмента (инженерии предприятий) и самых разных других проектах оказалось, что их целевые **системы не «объективны», они субъективны!** Их определяют агенты (в узком смысле, то есть «интеллектуальные агенты, способные к планированию», люди и AI-агенты), которые по отношению к самым разным системам проекта играют роли создателей/constructors::роль, пользователей::роль этих систем, жертв::роль этих систем, и так далее.

Вариант системного подхода, появившийся где-то с середины семидесятых по середину восьмидесятых годов 20 века, который учитывает, что одни системы создаются и развиваются другими системами-создателями, часто включающих и людей, получил название **системного подхода 2.0 («второе поколение системного подхода»). Математики в нём стало меньше, разговора о людях — больше. Кроме описания целевой системы стало важно описывать системы-создатели этой целевой системы.**

Роли агентов по отношению к самым разным (целевым, подсистемам, надсистемам, системам создания в графах создателей) системам оказались крайне разнообразны.

Роль по участию (помощь или вред — от роли проекту системы или от проекта системы этой роли) в проектах замысливания, создания, модификации, эксплуатации, уничтожения системы так и назвали «**проектной ролью**» (stakeholder role, заинтересованная сторона, действующее лицо). Эти проектные роли исполняют агенты-актёры/акторы/actors (термин из метода use case, но не только, термин ведь общераспространённый). В системной инженерии часто для агентов::конструктив использовался термин stakeholders, хотя слово stakeholder иногда и используют и для обозначения самой роли: путают stakeholder-актёра и stakeholder-роль, в которой сейчас актёр.

Деятельностные, трудовые, культурные, практические, инженерные (вы уже поняли, что тут может быть множество синонимов, «проектная роль» тут только

один из них) роли, конструктив/материал систем которых — агенты, обычно берутся культурно-обусловленными — по паттерну их деятельности/«виду труда»/«виду культуры»/практике/«виду инженерии»/«методу/способу работы»/функции и т.д. Вряд ли кто-то выполняет работу каким-то способом/методом/стилем/практикой/функцией первым в мире, так что и метод работы и роль уже наверняка как-то уже названы, вот и надо не изобретать новые термины, а брать эти названия метода и роли из языка. Если кто-то::агент-конструктив (например, Джон Смит) занят получением несанкционированного доступа к данным, то это занятие назовут взламыванием::метод, его назовут хакером::роль.

В первых вариантах этого нового (то есть 70х годов прошлого века) варианта системного подхода, в котором системы делались людьми, а не они сами как-то «объективно» существовали и развивались, проектные роли учитывались только те, которые влияли на систему и связанный с ней проект, причём влияли так сильно, что нельзя было этого не учитывать (например, заказывал создание системы, платил за это создание — не учтёшь такую роль, не будет денег на работы!). Позже поправились: те, на кого влияет целевая система и её создатели в рамках проекта создания, тоже считаются проектными ролями — это не только те, кто может наступить вам на ногу, но и кому на ногу наступаете (или можете наступить) вы! Мышление в отношении людей в системном подходе стало более «ньютоновским», менее «аристотелевским» («палец давит на стол, и стол давит на палец»): люди взаимодействуют с системами, а не только действуют на системы!

Похожие рассуждения про «наблюдателя» в физике (мы приводили ссылки на подобные исследования в предыдущем разделе) и рассуждения про активное восприятие в биологических системах, развиваемые группой Karl Friston[168], дали новый поворот этой линии рассуждений в системном мышлении. Оказывается, дело не столько в людях как «сознательных существах», сколько в самом таком способе рассуждений, при котором одни системы могут как бы (это «бы» тут важно) «измерять» другие системы — и люди-то как существа разумные точно могут «измерять» совсем неживые системы, и других людей, и организации людей, и самые разные другие системы. Эта способность свидетельствовать (то есть что-то там измерять) существование других систем и даже собственной системы (self-evidence, выяснение/измерение состояния себя, часто разбирается в работах Fields и Glazebrook) оказывается в силу панпсихизма в варианте минимального физикализма присущей вообще всем системам! **Все системы можно считать (хотя и в разной мере) деятельными/практичными/активными/агентными/вменяемыми и даже разумными/сознательными!**

Это «все системы» относится не только к животным и далее растениям, бактериям, или каким-то компьютерным системам, но и к организациям (для муравьёв это муравейник — и мы приводили ссылку на работу, где оценивается степень «сознательности» муравейника в целом по сравнению с «сознательностью» отдельных муравьёв, для людей это команда проекта, предприятие со многими проектами, и даже сообщества и общества, где много предприятий, а про человечество вместе с Землёй как «живую сущность» мы уже приводили ссылку на

[168] https://www.activeinference.org/ — институт, занимающийся развитием идей active inference как деятельного/практического/проактивного рассуждения/inference в системах самого разного рода.

работу астрофизиков).

Биологи уже давно обсуждают «проблему индивида»[169], ибо непонятно, какую систему (или объект, не сводимый к системе, но чаще всего ищут именно систему или хотя бы тип систем) в природе считать главным субъектом эволюции: геном? Организм, который развился из генома? Популяцию/семью/прайд/стаю (самец и самка — это уже не организм, а популяция)? Популяцию в ареале обитания? Это же самые разные системы! И все живые (в какой-то мере) и обладающие сознанием (в какой-то мере), так что при рассмотрении влияния одних систем на другие нужно как-то учитывать не только роли людей-индивидов, но и роли совершенно других систем! Тезис очень непривычный, но приведённая нами литература показывает, что этот тезис позволяет продуктивно обсуждать такие ситуации, которые непонятно как обсуждать, если считать отдельных людей чем-то исключительным в плане агентности, единственными представителями «агентов в узком смысле».

Второе поколение системного мышления внесло в рассмотрение людей-создателей, ибо от физики-биологии мы ушли в инженерию, и без этого было уже нельзя, но если люди как агенты оказались исключительны в системном подходе, то мы уже потеряли безмасштабность в обсуждениях. Системный подход 2.0 (который создавался с середины 70х по середину 80х годов прошлого века) к началу 20х годов 21 века насчитывает уже 40 лет — немалый срок! Но он оказывается системным подходом масштаба систем одного организационного проекта и срока жизни одного человеческого проекта. Это сильно сужает рамки рассмотрения систем.

Всё больше и больше мыслителей переходят к обсуждению универсальных создателей, необязательно людей и коллективов людей. Примером тут может служить и исследовательская программа David Deutsch по разработке теории создания/constructor theory[170]. Особенность английского языка: у Deutsch это создатель/строитель, а не «конструктор» из конструкторского бюро или металлический «конструктор»! Создатель::система меняет окружающую среду каким-то определённым образом, при этом сам не меняясь и сохраняя возможность менять эту среду этим образом много раз. Простейший «создатель» — это молекула катализатора, более сложный «создатель» — это робот с универсальным компьютером внутри, ещё более сложный — это человек, но и общество тоже выглядит таким «создателем». Создатели по Дойчу могут участвовать в графах создателей (помним, что в физике системы брались как уже данные, в биологии они сами ели-пили-выращивались, а вот в инженерии нужны были создатели::система, которые создавали целевые системы), в том числе создатели могут, если у них есть соответствующие знания/программа (человек, и даже завод имеют универсальный вычислитель, теоретически эквивалентный машине Тьюринга) создавать других создателей. Создатели также могут ремонтировать и менять себя, а не только своё окружение. Так что создатели-по-Дойчу вполне пригодны:

- Для обсуждения физики (основная идея Дойча была в том, чтобы предложить тип безмасштабного физического объекта, который бы позволил обсуждать и микромир с квантовой физикой, и макромир с теорией гравитации, в физике безмасштабность сейчас очень насущная проблема)
- Для обсуждения инженерии (любой станок — создатель! И человек —

[169] https://plato.stanford.edu/archives/win2021/entries/biology-individual/
[170] https://www.constructortheory.org/

создатель! И завод — создатель! Разница только в том, что есть «просто создатели», но есть универсальные создатели, которые теоретически могли бы создать что угодно, не запрещённое законами физики, например, последовательно создавая инструменты для создания этого «чего угодно» и создавая себе подобных создателей, чтобы увеличить масштабы своих созданий.)

- Для обсуждения жизни как непрерывного усложнения систем (мы упоминали работы Ванчурина, Вольфа, Кацнельсона, Кунина на эту тему — идею о том, что неустроенности между системными уровнями приводят к росту сложности организации вещества, это и есть жизнь)
- Для единообразного обсуждения графов создателей в системном мышлении, причём создателями могут быть и машины, и люди, и организации из людей и машин, и AI-агенты.

Есть ли какая-то разница между классическим создателем из системной инженерии (традиционный термин для него — enabling system) и создателем из теории Deutsch (constructor)? Это одно и то же понятие, обозначаемое в английском разными словами и в русскоязычном тексте руководства одним словом, или разные понятия? Это одно и то же понятие, но с нюансом: constructor это из теории Дойча, по факту это особый класс систем, который не разрушается после проведения каких-то изменений состояния окружающих систем. Большинство enabling systems именно таковы: это агенты (индивидуальные и коллективные) и их инструменты, которые создают и развивают целевую систему (но в том числе и друг друга в графе создания), но при этом сами не «расходуются», сохраняют свою способность работать дальше после окончания какой-то очередной работы над изменением состояния очередной системы. Поэтому и создателя, и целевую систему мы можем назвать constructor, если её рассматриваем в рамках constructor theory. А enabling system у системных инженеров будет отсылкой к системе в графе создателей, причём во время создания и развития целевой системы. Но в целом — никакой разницы, ибо даже целевая система вполне в момент своей работы может что-то изменять в своём окружении, быть constructor из constructor theory. Поэтому по-русски мы оба термина переводим как «создатель».

Сейчас выходит довольно много работ, которые условно можно отнести к **третьему поколению системного подхода, системному мышлению 3.0[171].** Работы по системному подходу третьего поколения подчёркивают безмасштабность рассмотрения в размерах систем (от квантовых до космологических) и времени (от фемтосекунд до миллиардов лет, чтобы уметь рассмотреть и квантовомеханические системы с их скоротекущими процессами, но и космические системы с их миллиардами лет, необходимыми в том числе и для эволюции жизни).

Тренд в том, чтобы люди не выпячивались как нечто особенное из описаний физического мира. Объяснения особой роли людей тем, что «человек — это звучит гордо!», или тем, что «это наш вид, мы должны быть его патриотами!», или тем, что «только люди разумны, они цари природы!» — плохие, с такими объяснениями не построишь надёжных объяснений окружающего мира и не создашь реально сложных систем. С учётом того, что на Земле появляются ещё и AI-агенты, а также набирает обороты генная инженерия, эти объяснения становятся уж совсем плохими. Наука и научный метод начинают подменяться произвольно взятыми «с

[171] https://arxiv.org/abs/2310.11524

потолка» этическими утверждениями.

Проблемы с этими объяснениями картины мира как «картины мира, зависящей от её описания именно живыми людьми» как раз и приводят к предложению всё новых и новых работ с обсуждением физической и биологической проблематики на базе безмасштабности в применении понятий системного подхода. Для этого сейчас ищется и терминология: людей заменяют «агентами» (вы это уже заметили в тексте нашего руководства), а сейчас и «агентность» оказывается не самым главным — разные системы просто обладают разными способностями к тому, что связывалось с «агентностью», то есть они все в разной степени сознательны, у них разная память и способность моделировать, разные вычислительные возможности для планирования действий, разная вменяемость. В любом случае, описания физического мира стремятся сегодня делать так, чтобы это были безмасштабные в пространстве и безмасштабные во времени описания физического мира самых разных систем, а не описания «мира людей, как он видится именно людям, как привыкли видеть люди в прошлом веке».

Цели деантропоморфизации описаний мира в языке и мышлении (окончательный уход от антропоцентризма[172]):

- уход в мышлении и языке от привязки к конкретному системному уровню «человека» в эволюции вселенной. Если мы имеем многоуровневую эволюцию, то нужно обеспечить внятный язык разговора о проблеме индивида в биологии, говорить о многоуровневости индивидов, иметь имя для «эволюционной единицы» (разговор о генетической и меметической эволюциях, репликаторах, генах, мемах). И можно при этом понимать, что «индивид», «агент» (в широком смысле слова), IPU (information processing unit)[173] — это одно и то же! И молекула тоже в какой-то мере индивид/агент/IPU, и человек-как-один-организм, и даже общество!
- уход от парохиального/parochial (частного, для текущей ситуации, в которой находятся именно люди на Земле) мышления, посадка мышления на общую привязку к физике. Астрофизики и математики — это физические системы, хотя и очень сложно устроенные, страна — это тоже физическая система, хотя ещё сложней устроена, чем астрофизики и математики, роли которых играют люди, хотя уже иногда и AI-агенты. Эти роли астрофизиков и математиков могли бы играть и исследовательские институты людей-астрофизиков и людей-математиков, вместе с задействуемыми в этих институтах AI-агентами.
- выявление трансдисциплинарных рассуждений по линии усложнения «инертное вещество»-существо-«разумная личность»-организация-сообщество-общество. В части простой физичности это создатели/constructors, в части выхода в мир разумных существ это агенты и мультиагентные системы. Терминология может отличаться, но уж точно не «human/человек», причём создатели становятся универсальными создателями, universal constructor[174], то есть в конечном итоге речь идёт о потенциально любых (теоретически любых! Как универсальная машина Тьюринга может теоретически выполнить любые вычисления, если речь идёт о чём-то математически вычислимом) преобразованиях вещества в целевые

[172] https://ru.wikipedia.org/wiki/Антропоцентризм
[173] https://www.pnas.org/doi/10.1073/pnas.2120037119
[174] https://www.youtube.com/watch?v=40CB12cj_aM

системы, если это не запрещено законами физики. Фактически и constructor theory, и multilevel evolution, и active inference, и агентный подход в философии (agency)[175] в целом работают в эту сторону.

- рассматриваем не только животных и людей, но и киборгов, и искусственных интеллектов, вместе с их оборудованием по изменению физического мира, а далее выход в их организации/коллективы и сообщества, и даже общества и всё «агентечество» (если бы оставались в рамках антропоцентризма, то написали бы «человечество»).
- выход на новую праксеологию[176] (общую теорию целенаправленного действия, близкую к методологии), социологию, политику, экономику, правоведение: преодоление многочисленных проблем, возникающих при обсуждении неустроенностей между разными системными уровнями в обществе как людей, так и не совсем людей, в том числе учёт распределённых вычислений и действий в фирме, в рынке, работа с пулами ресурсов (собственность и коллективная собственность), включая вычислительные/мыслительные ресурсы и ресурсы действия по изменению мира к лучшему (при самом разном понимании этого «к лучшему», помним про противоречия систем разных системных уровней).
- отвязка проблемы управления вниманием (сознания) от особенностей именно человека, обсуждение не только людей, но и людей совместно с животными и компьютерами, в том числе роботами с датчиками и актуаторами/эффекторами: агентов в узком смысле как универсальных создателей/constructors с какими-то способностями к моделированию и планированию, причём самой разной природы, включая их организации и популяции: команды, фирмы, эко-системы, сообщества, общества, но ещё и одноранговые сети агентов в роях/swarms и много чего подобного, с учётом разнородности участников этих агентских/создательских организаций.
- Более продуктивный разговор про этику (экстропианство[177], трансгуманизм[178], биоэтика, этика AI, проблемы подчинения людей и не-людей обществу, общества человечеству и всё такое) в условиях бесконечного биологического и технологического развития и наличия многих системных уровней с неизбежными противоречиями между системами этих уровней и механизмами, подобными механизмам «неустроенностей»[179].
- ... и много другого, что проще становится обсуждать при отказе от чёткой привязки только к одному системному уровню «человека-индивида». Такая привязка даёт ведь не только «птолемеевскую модель человека»[180], но ещё и «птолемеевскую модель мира» вокруг только человеческого вида существ, при всей их декларируемой особости-разумности. От этой парохиальной, частной «для людей на Земле» модели мира надо уходить к более общей, неантропоцентричной и безмасштабной, основанной на непрекращающейся эволюции вселенной (часть которой — дарвиновская эволюция жизни на Земле, а также сегодняшняя недарвиновская техноэволюция).

---

[175] https://plato.stanford.edu/entries/agency/
[176] https://en.wikipedia.org/wiki/Praxeology
[177] https://en.wikipedia.org/wiki/Extropianism
[178] https://en.wikipedia.org/wiki/Transhumanism
[179] https://en.wikipedia.org/wiki/Geometrical_frustration
[180] https://ailev.livejournal.com/1390574.html

Методология: агенты, роли, предметы интереса, интересы

В современном системном мышлении мы не акцентируем, кто будет играть роль «разумного существа»: человек ли, человек с усилением его интеллекта за счёт компьютера, организация из людей и компьютеров, или компьютер с искусственным интеллектом (или искусственный интеллект с компьютером: это как посмотреть). Все они будут делать что-то такое же, что обычно делают сейчас люди, причём неотличимо от них (в научной фантастике весьма популярны сюжеты, когда за «юридическим лицом» скрывается какой-нибудь «искусственный интеллект»[181]).

Если мы используем язык методов работы и работающих по этим методам ролей, а не язык с указанием исполнителей ролей, то нам неважна природа исполнителя роли (неважен конструктив/материал для функционального объекта). Если нам надо получить рожок с мороженым, то нужен какой-то исполнитель функции «продажа рожка с мороженым», ибо рожок сам себя не продаст! Неважно, какой агент будет выполнять эту функцию: Петя-мороженщик (вот оно, название роли у агента Пети!), вендинговый автомат номер 4 на втором этаже (его мороженщиком по роли не назовут, но «продажа рожка с мороженым» у него в числе функций есть!), ресторан фастфуда (его тоже не назовут «мороженщиком», но рожок с мороженым вам там продадут!) или даже целая сеть ресторанов фастфуда (тоже не «мороженщик», но функцию «продажа рожка с мороженым» вам выполнит, причём можно ещё и будет выбирать ближайшее место, где вы это мороженое получите).

Функция — наиболее устойчивый объект в рассмотрении, она меняет состояния каких-то объектов окружающего мира (рожок запрошен, рожок обещан, рожок изготавливается, рожок выдан, а что там с деньгами — пока игнорируем). Дальше рассматриваем культурно-обусловленную роль «мороженщик», что для человека просто, но мы вполне можем так называть и вендинговый автомат (который может быть и мороженщиком, и бариста — в зависимости о того, что запросит клиент), и даже ресторан фастфуда в тот краткий миг, когда он будет поставлять нам мороженое. Мы можем по роли называть и какие-то другие сущности, которых мы пока даже не знаем, даже названия которых только придумываются. Например, biocyborg[182], hybrot[183] и так далее, при этом AGI как artificial general intelligence тоже часто используется не как «интеллект» какого-то существа, а как «интеллектуальное существо» без отсылки на конкретное телесное/аппаратное/биологическое/машинное воплощение.

Так что дальше мы будем много говорить о ролях и их функциях, а когда поминать «агентов» как исполнителей ролей, то это отнюдь не всегда люди. При разговоре о **ролях** и **агентах** это могут быть совсем не отдельные люди-индивиды, а иногда и не совсем живые сущности-объекты, а иногда и большие организации: разговор в системном мышлении ведётся по большей части безмасштабный и не антропоцентричный. И синонимы вместо «функции» тоже для интеллектуальных и коллективных агентов могут быть другие — метод/способ работы, практика/культура/стиль/деятельность, вид труда, вид инженерии.

В дальнейшем разговоре мы будем иметь в виду под агентами главным образом агентов в узком значении, то есть планирующих интеллектуальных агентов, или как их ещё называют, **рациональных агентов** (хотя рациональность и связанные с

---

[181] Пункт 5 в https://ailev.livejournal.com/1258352.html

[182] https://www.tandfonline.com/doi/full/10.1080/14636778.2021.2007064

[183] https://en.wikipedia.org/wiki/Hybrot

ней теории принятия решений мы подробно и не рассматриваем, это предмет других руководств — по рациональной работе, по системному мышлению). Рациональные агенты на сегодняшний день чаще всего именно люди, условно к ним можно отнести и организации людей, но и AI-агенты уже во многом проявляют рациональность.

Обычно рациональных агентов, которые играют роли в проектах по созданию и развитию каких-то систем, интересует целый ряд **«важных/интересных характеристик»/«предметов интереса»** (concern, «озабоченность», реже — interest, иногда даже driver как «то, что важно», ключевая/целевая/ведущая характеристика, объект пристального внимания для какого-то «метода работы»/практики/деятельности).

«Предметом интереса»/«важной характеристикой»/«интересной характеристикой» может быть любая характеристика системы или проекта, типовая для многих проектов, или уникальная только для этого проекта. Температура, срок разработки, возможность регулировки и настройки, время наработки на отказ: **предметом интереса может быть любая важная для какой-то проектной роли характеристика.** Стоимость, производительность, ремонтопригодность, функции и фичи/возможности системы, сроки годности, безопасность, и так далее: что угодно, что важно для успешного перевода каких-то объектов разными методами работы из одного состояния в другое.

Обращаем внимание, что характеристика или предмет интереса — это не значение характеристики! Температура воды может быть предметом интереса в бассейне или на пляже, она крайне важна для работ по методу «купание в воде», но мы ничего пока не сказали о значении характеристики, значении температуры! Грубо говоря, мы сказали только о наименовании шкалы, по которой потом будем судить о значении — ремонтопригодность как предмет интереса подразумевает, что потом мы будем как-то оценивать значение этой ремонтопригодности, большое оно, среднее, маленькое или ещё какое. Но «предмет интереса», «характеристика» — это свойство, а не значение свойства.

Когда планируют разбить этот предмет интереса на более мелкие, то могут говорить «область/зона интереса»/**«area of concern»**, а в такой области интереса будут несколько частных предметов интереса. **Областью интереса может быть любой набор характеристик, которые трудовые/деятельностные/проектные/орг роли считают важными.** Скажем, область интереса клиента может включать характеристики самой системы (что она делает и как хорошо), цену целевой системы, время изготовления, место получения готовой системы. И опять: мы ничего не сказали пока о значениях этих характеристик! Мы пока обсуждаем в терминах шкал/свойств, а не значений этих шкал/свойств.

Обратите внимание, что мы только что расширили синонимический ряд для «роли»: добавили эпитеты, указывающие на разные контексты использования понятия «роль»: проектный или организационный контексты ведут к уточнениям для рациональных агентов — «проектная роль» (роль по исполнению каких-то методов работы в текущем проекте), «орг-роль» (роль по исполнению каких-то методов в организации). Конструктивы/материалы, а также аффордансы тут те же самые, что и для других ролей более-менее универсальных создателей — люди, подразделения в организации или целые организации, коллективы с их компьютерами и

оборудованием, а также помним об AI-агентах.

Разные роли имеют разные **интересы/предпочтения в «предметах интереса»/«важных характеристиках»**. Их иногда так и называют: **ролевые интересы, ролевые предпочтения**. Это означает, что речь идёт не о предпочтениях агента-конструктива Васи, играющего роль Принца Гамлета, а о предпочтениях Принца Гамлета по его роли.

Если это Даша-продавец, то речь идёт о предпочтениях не лично Даши, а о предпочтениях роли «продавец», исполняющей работы методом «продажи».

Если общий предмет интереса/важная характеристика/интересная характеристика проектных ролей покупателя и продавца — цена, то ролевое предпочтение/интерес покупателя в минимизации цены, а предпочтение/интерес продавца — в максимизации цены. Поэтому говорим о двух понятиях: характеристике с её значением и предпочтении, ибо для двух и более ролей характеристика может быть одна, а вот предпочтения/интересы разные. Не путаем: интерес/предпочтение как нацеленность на конкретное значение на шкале/характеристике/свойстве против «предмет интереса»/«предмет предпочтения» как сама шкала/характеристика/свойство.

Например, цена::«предмет интереса» будет одна для всех договаривающихся в ходе проекта ролей, а сам интерес/предпочтение (иметь цену побольше или поменьше) для каждой из ролей будет разный. Предпочтение/интерес — это куда тащить в согласованиях значение характеристики/свойства: делать его побольше или поменьше, если значения непрерывны, или поближе к одному значению или другому, если состояния дискретны («жарко — открой дверь, все уже употели», «сквозняк ведь — закрой дверь, я только отболел простудой, потеть — это нормально»: предмет интереса — «открытость двери», интересы разные!).

Люди и их организации (а иногда и не только люди, и не только их организации, дальше мы не будем делать этих оговорок — но где сказано «люди», там будьте бдительны, в жизни это может оказаться *другое*) тут рассматриваются как автономные **агенты**, которые **исполняют роли**. Иногда агента в качестве исполнителя ролей называют **актёром/актором/actor::конструктив,** иногда говорят **оргзвено::конструктив** (в отличие от оргроли::функциональный объект, слово «подразделение» мы не используем, ибо оргзвеном можно назвать и предприятие в целом, и подразделение, и отдельного человека, и какие-то коллегиальные органы — советы, комиссии, которые чаще всего не считаются подразделениями).

Агент-создатель занимается «работой по методу»/деятельностью/трудом/практикой/инженерией/созданием, поскольку имеет **намерение/intention**[184], направленное на устранение какого-то комплекса неустроенностей на самых разных системных уровнях, которое заставляет его выполнять работу по какому-то методу, направленную на изменение модели мира, мира, модели себя, себя. Работая по методу, агент играет роль и преследует ролевые интересы, то есть добивается предпочтений в каких-то интересных ему характеристиках тех или иных систем. Намерение/intent к действию — это элемент предпринимательства/проактивности/исследовательского любопытства/пытливости как оценки неопределённого будущего и творческого

[184] https://plato.stanford.edu/entries/intention/

предложения каких-то изменений мира (например, создание каких-то новых систем) с целью убрать неприятные сюрпризы от мироздания, побуждение к творчеству.

Есть догадка подхода active inference[185], которую пока не опровергли, поэтому мы принимаем её всерьёз: **все без исключения системы любых масштабов (как устойчивые к внешним воздействиям объекты, которые как-то ухитряются сохранить себя в физическом мире) занимаются действием, реализующим принцип минимизации свободной энергии из физики. Рациональные агенты занимаются проактивным действием, то есть планируют возможные неприятные сюрпризы в будущем — и действуют на упреждение, а также объединяются в коллективы, чтобы действовать эффективней[186].**

Так ведут себя и молекулы, и люди, и организации, и так далее до человечества в целом. Подробности можно узнать, пройдясь по ссылкам к работе Chris Fields, Karl Friston, James F. Glazebrook, Michael Levin, «A free energy principle for generic quantum systems»[187], 2021. Конечно, в каждом конкретном случае речь идёт о самых разных намерениях и действиях, но так уж оказывается, что в силу абсолютно физического принципа наименьшего действия[188] агент::система в меру степени своей разумности/рациональности, объёма доступной памяти (можем всегда подключить компьютерную память и память других людей!) и вычислителя (кроме мозга можем подключить ещё другие мозги и компьютерные вычислители, в том числе масштаба датацентра) реализует физический принцип минимальности свободной энергии: минимизирует ожидаемый неприятный сюрприз от мироздания, то есть минимизирует свои шансы исчезнуть, разрушиться, рассеяться под натиском энтропии. Если вы будете ломать камень, то он будет сопротивляться этому, уж как может. Если будете ломать человека, то он это может предвидеть и либо убежит, либо (это будет уже вам сюрприз) даст ответку — и ещё не лично сам, а с друзьями и при помощи разных неприятных вам инструментов-оружия. Как ведут себя при попытке их разрушить сообщества и общества — это вопрос интересный, но ответы тут трудны, ибо спросить их нельзя (любой человек будет готов говорить от их имени, и говорить все эти люди-представители будут разное). Можно заметить, что живут сообщества и общества иногда меньше, чем отдельные люди, но иногда и во много раз дольше, чем люди, а человечество так и вообще существенно дольше, чем отдельные люди — то есть таки сообщества, общества и человечество тоже как-то сопротивляются напастям мироздания, и даже более успешно, чем отдельные люди. Но **говорить о сообществах и обществах антропоморфно (например, что у них есть намерение/intent, которое мы можем узнать и учесть, что у общества есть интересы — это всё язык обсуждения агентов-индивидов) нужно опасаться, развитого языка для такого разговора пока нет.**

Если вернуться к таким рациональным агентам, как отдельные люди и организации, то намерение не только заставляет преследовать ролевой интерес, но часто заставляет актёра ещё и переключаться между ролями, тем самым меняя ролевые интересы. Поэтому важно не только понимать предметы интереса и предпочтения в этих предметах интереса для проектной роли, играемой каким-то агентом-актёром

---

[185] https://www.activeinference.org/
[186] https://www.sciencedirect.com/science/article/pii/S0149763423004694
[187] https://arxiv.org/abs/2112.15242
[188] https://ru.wikipedia.org/wiki/Принцип_наименьшего_действия

(Богданом, который играет роль инженера-теплотехника), но и понимать намерения самого актёра как исполнителя ролей (например, Богдан хочет есть, поэтому играемый им инженер-теплотехник неожиданно сворачивает разговор по существу и разговор заканчивается быстро, но безрезультатно, а Богдан переключается в роль едока и быстро уходит туда, где он может поесть).

Для реализации/воплощения/проведения в жизнь намерений агенты выдвигают **стратегии**, основанные на как-то понимаемых ими причинно-следственных связях в ситуации. По большому счёту, стратегия — это продолжение синонимического ряда метод/практика/…. Это некоторый паттерн действий, и в этом плане стратегия/метод работы — это всё одно и то же, за мелкими нюансами, «способ, которым мы добьёмся успеха в получении результата». На основе стратегии (это метод, то есть нет привязки к конкретному времени и экземплярам ресурсов) затем разрабатывают **план действий,** в котором для реализации ролевых объектов подбирают конструктивы-аффордансы как ресурсы и планируют работы, реализующие стратегию/метод в какие-то определённые моменты времени, когда будут доступны ресурсы.

Всё, что происходит с самими методами (как они создаются, как их сравнивать, как они эволюционируют, как реализуются работами) изучается **методологией**, как учением о методах работы агентов. В настоящем руководстве даются элементы методологии, но само понятие метода/труда/деятельности/практики/инженерии/стратегии будет дополнительно рассмотрено в отдельном руководстве по методологии (как думать об изменении мира к лучшему), а общие методы инженерии (что делать для изменения мира к лучшему) в отдельном руководстве по системной инженерии, а затем специализации этих методов инженерии будут рассмотрены в кругозорных (инженерия личности, системный менеджмент) и далее прикладных руководствах инженерии.

Люди как агенты-актёры обычно очень изобретательны. Перед тем как сделать выбор/принять решение, они генерируют и оценивают разные варианты решений, занимаются творчеством — если вы им предложите выбрать между красным и синим, они по мыслительной норме принятия решений обязаны задаться вопросом, какие ещё выборы могут быть, и затем неожиданно выберут зелёное и приложат силы для того, чтобы эта опция выбора появилась. Так что агенты могут запустить совсем неочевидную для вас цепочку действий даже при сугубо положительных для вас предпочтениях.

Например: предмет интереса: головная боль, предпочтение/интерес: голова не должна болеть, стратегия/метод лечения: декапитация, «отрубить голову — нет головы, не будет головной боли», роль для этого метода: палач с предметом интереса/характеристикой «наличие головы» и интересом/предпочтением «чтобы головы не было», далее найти агента на роль палача и выполнить работы декапитации::метод в доступное для агента-палача время одним из изученных агентом, исполняющим роль палача, эффективных вариантов этого метода. Будет ли реализована такая последовательность событий — это существенно зависит от степени рациональности агента, умении разбираться с клубком целей (интересы) и средств (методы/практики). Не все агенты проходят курсы, где их обучают рациональному принятию решений, а не абы какому (например, «интуитивному принятию решений»)!

Ещё для реализации намерения агента в его стратегию и планы может входить задействование других агентов в других ролях, которые могут выполнять какие-то совсем другие методы, и им наш агент может за это даже заплатить! Но наш агент может отвлечься при этом от исполнения своей роли и выполнить работу других ролей сам, если имеет для этого квалификацию. Если не имеет, то тоже может — только работа другой роли будет выполнена плохо, но этого агент может и не понять!

Это окончательно запутывает ситуацию, но методологических понятий «агент», «актёр»:: «исполнитель роли», «роль», «предмет интереса/важная характеристика системы или проекта», «интерес/предпочтение», «намерение», «стратегия», «план» должно хватить для подробного разбирательства с самыми разными ситуациями, связанными с поведением людей в проектах, включая обсуждение их конфликтов и кооперации. Цель руководства ровно в том, чтобы наставить на работу с этими понятиями для управления вниманием в сложных ситуациях со многими агентами, которые реализуют самые разные намерения в силу самых разных интересов. Вы должны найти в жизни объекты этих типов.

Конечно, прежде всего надо обращать внимание на действия (искать метод работы, затем разобраться, что это за роль, затем разобраться, какие «предметы интереса»/«важные характеристики объектов практики» есть у этой роли, какой интерес преследует роль). Но надо ещё и обращать внимание на слова, если действия пока невидимы, а доступны только речи или тексты агента, который «непонятно чего хочет». Вася::агент, следовательно в разговоре со мной он имеет какое-то намерение и играет какую-то роль — и какое это намерение, какая роль? И насколько рационален Вася, насколько безумны могут быть его действия по реализации этих намерений? Я::агент, а какие мои намерения, какие мои роли при общении с Васей? И какие после осознавания их я должен предпринять действия?

К сожалению, методологические рассмотрения затрудняются тем, что:

- В ситуации обычно участвуют несколько агентов/актёров.
- Каждый агент имеет невидимые остальным агентам (необязательно людям) в проекте намерения.
- Каждый агент участвует в нескольких проектах, и выигрыш агента в одном проекте может ведь вести к проигрышу в другом проекте, и агент займётся генерацией идей по оптимизации ситуации. Часто — многоуровневой оптимизации, ибо будут влиять «неустроенности» от конфликтов между системными уровнями: в каком-то предмете интереса, например, зарплате, интерес сотрудника один «поднять зарплату», интерес предприятия «платить сотруднику поменьше». Агент может увеличивать общий выигрыш свой или даже не свой (если он договорился о кооперации с другими агентами). То есть агент может выполнять множество ролей, о части которых вы не знаете, в том числе и в разных проектах, и в одном проекте, некоторые из этих ролей могут конфликтовать (особенно, если они ответственны за разные системные уровни, ибо между системными уровнями неминуемы конфликты, ведущие в конечном итоге к неустроенности в виде множества самых разных вариантов решения, ни одно из которых не оптимальное, но все как-то «одинаково плохие, хотя и разные»).
- Каждый агент строит невидимую остальным агентам (в том числе людям) в

проекте стратегию задействования ролей в разных проектах и разрабатывает невидимый остальным агентам (в том числе людям) в проекте план действий. Если агент видит, что его стратегия не реализовывается, то он неожиданно для остальных может поменять эту стратегию и изменить планы, в том числе это может означать, что он будет отыгрывать совершенно другие роли с другими важными характеристиками!

- Каждая роль может иметь множество рассматриваемых ей важных характеристик/предметов интереса с предпочтениями/интересами в них.
- Это мы ещё не коснулись того, что агенты затем выполняют планы (производят работы) по методам/практикам, которые хорошо известны выполняемым ими ролям. И тут агент может быть дважды некомпетентен, то есть дважды не мастер: 1. в своей роли, он может неумело её отыгрывать или даже вообще не знать, и 2. плохо владеть интеллект-стеком, плохо мыслить, то есть принимать нерациональные и неадекватные решения при встрече с любой ситуацией, которая раньше ему не встречалась.
- А ещё у агентов есть полномочия по распоряжению своим и чужим трудом и оборудованием (грубо говоря, есть должности в организации). И агентом могут путаться роли свои и своих сотрудников.
- И тут ещё не обсуждаем методы как предмет методологии, да и сами работы как предмет операционного менеджмента (задействование ресурсов с максимизацией прохода работ через эти ресурсы, включая самого агента-исполнителя роли как ресурс).

Методологическое моделирование

Моделировать графически что-нибудь в этой ситуации не получится[189]. Вот пример попытки такого моделирования: начинаем рисовать для агента-1 и его намерения, потом добавляем второго агента и его намерение, упоминаем его стратегию/метод и план действий, пару важных характеристик и предпочтений первой его роли, потом добавляем вторую роль второго агента, и тут понимаем, что диаграмма уже сложная и продолжать в таком духе дальше — только запутаться. Но мы ещё не нарисовали методы, по которым действуют роли! И на картинке ещё нет систем, с которыми работают методы! И даже с уже затронутыми агентами, намерениями, стратегиями, планами действия, предметами интереса и предпочтениями мы возвращаемся к перечислению затруднений из предыдущего абзаца, и понимаем, что мы только-только начали моделирование!

[189] Невозможность графического моделирования разбирается в книге А.Левенчук, «Визуальное мышление. Доклад о том, почему им нельзя обольщаться», 2018, https://ridero.ru/books/vizualnoe_myshlenie/

Но это ведь всё важные объекты в ситуации, которые обязательно нужно рассматривать. Так работает методологическое мышление, которое находит в ситуации объекты пропущенных типов и обращает на них внимание, заставляет о них рассуждать. Без этого не будет и системного мышления (помним, что важные характеристики/предметы интереса — это характеристики самых разных систем, встречающихся в проекте, все эти проявляющие свои интересы роли работают с системами).

Какой тут выход? Проводить все мыслительные операции в уме? Тоже не выход, во внимании все системы и всех агентов с их быстро распухающими описаниями не удержишь. Поэтому выход в табличном (или даже текстовом) моделировании. Объектов и их отношений в реальном мире много, а вот типов объектов в методологии (агент, намерение, стратегия, план, роль, важная характеристика, предпочтение, мастерство, и пока в методологии ограничимся этим набором понятий. Метод работы как таковой, работу, должность, будем рассматривать потом) не так уж много, и эти типы будут упомянуты в колонках довольно длинных таблиц.

Как это выглядит в жизни? Подробней об этом рассказано будет в руководстве по методологии. Там будет показано, как из графов объектов и графов состояний объектов делать списки, а затем списки представлять в форме удобно редактируемых таблиц. Ключевое в решении проблемы системного моделирования — это понимание, что у объектов внимания в проекте есть типы, и объекты типов из наших руководств (типы мета-мета-модели). От объектов проекта «из жизни» надо переходить к типам «из учебника» и далее думать, используя эти типы из учебника для понимания того, что делать с объектами из жизни.

Мы в руководствах обозначаем тип значком **::** (два двоеточия). В бытовой речи это

часто было бы или «тип объекта» («агент Вася») или «объект-тип» («Вася-агент»), но мы не полагаемся на совсем уж бытовой язык и отношение классификации показываем значком **::**, приводя сначала объект, а затем его тип. Мы не говорим «система самолёт» или «самолёт-система», в жизни мы опускаем указание типа, это норма. Но зачем вообще использовать указание типа, если в жизни тип не указывается?

Тип улучшает понимание (Форд::автомобиль, Форд::президент США, Форд::основатель ФордМоторс::компания — типы существенно облегчают понимание значения слов), но главное — типы помогают в моделировании. Моделирование — это создание модели, то есть объекта, который в каких-то важных чертах повторяет поведение (часто в мышлении, в вычислении) моделируемого объекта. Это означает, что модель существенно сжимает информацию о моделируемом объекте, указывает только на важное. Вот типы и используются для указания на важное.

В проекте для каждого объекта/сущности/понятия нужно разобраться с его типом. Если речь идёт о каком-то трансдисциплинарном (то есть системном, методологическом и т.д.) рассуждении из нашего руководства, то надо найти в жизни недостающие объекты, обратить на них внимание — и делать это письменно, «системное моделирование». Это и есть «задействование системного мышления» (не пишешь — не думаешь! Не моделируешь — не думаешь системно!).

Методологическое мышление, системное мышление и все остальные виды мышления — это способы управлять вниманием в сложной ситуации, чтобы найти объекты, для которых удобно обсуждать причинно-следственные связи, удобно строить объяснения. Весь материал про методы, роли и агентов — это как раз материал про типы, которые как-то связаны в тексте руководства, между ними прописаны отношения.

Так что если кто-то что-то в проекте делает (если ничего не делает — не обращайте внимания!), то он занимается работой по какому-то методу. Каков этот метод/практика/культура/стиль? Есть ли другие способы выполнить работу, получить ожидаемые результаты быстрее и дешевле? Этот «кто-то» тем самым — агент в роли! Какая это культурно-обусловленная роль? Когда нашли роль, то идёте в рассуждении дальше: у выполняющей какой-то метод работы роли будут какие-то важные характеристики/предметы интереса, в которых роль будет иметь свои предпочтения/интересы, которые метод работы по этой роли хотел бы видеть реализованными.

То есть если вы видите, что кто-то что-то продаёт (причём продажа может быть реализована очень по-разному! Вам ещё нужно догадаться, что какие-то действия агента представляют собой «продажу»::«метод работы»). Тем самым вы видите агента/актёра/«исполнителя роли»/оргзвено как продавца::роль. Если есть «продавец», то у него немедленно ищем предмет интереса (цена) и предпочтение (цена должна быть повыше). Дальше можно уже выдвигать свои стратегии (методы, которыми действовать дальше) и строить реализующие их планы. Например, искать вторую роль у агента (Васи, играющего роль продавца), или идти искать другого продавца, или предлагать сотрудничество (обнаружить, что нужно по таким ценам у Васи не покупать, а становиться его дилером и бросать свой предыдущий проект, то есть выходить из одной своей роли, а то и проекта, и реализовывать совсем другую стратегию).

Если имеющие противоположные предпочтения/интересы в одной важной характеристике системы или проекта роли исполняются одним оргзвеном/агентом/актёром, то такая ситуация называется «конфликт интересов», и это очень частая ситуация. Вспомним о поговорке «пусти козла огород охранять»: сам козёл считает, что он сбалансированно соблюдёт предпочтения по поеданию огородной капусты и по охране капусты от этого поедания. Люди со стороны будут другого мнения. Конфликт интересов изнутри головы носителя этого конфликта плохо заметен, но хорошо заметен обычно другими участниками проекта. Себе же человек (или представитель оргзвена — организации со многими людьми, но находящейся в конфликте интересов) кажется экспертом, глубоко знакомым с разными своими конфликтующими ролями, и поэтому принимающим «взвешенные решения». Но нет, эти решения для других участников ситуации не кажутся «взвешенными»! Козёл, охраняющий огород, конечно, принимает «взвешенные решения» по тому, сколько от огорода отъесть, а сколько сохранить от любых поползновений!

Положительность по части предпочтений в их предметах интереса для проектных ролей необязательна. Просто **предпочтения «отрицательных героев» и «антиклиентов» учитываются в проектах с обратным знаком** — ворам не дают украсть, убийцам не дают убить, террористам не дают совершить теракт. При этом помним, что в одном проекте будет «проклятый шпион и предатель», но тот же человек в другом проекте — «доблестный разведчик, рискующий жизнью в святой борьбе». То есть эта «положительность предпочтений» связана с этическими рассуждениями, подробней об этом в разделе этики руководства по интеллект-стеку фундаментальных методов мышления.

Проектные/трудовые/деятельностные/организационные роли — это именно роли, их играют, в них не «перевоплощаются». Агенты реализуют ролевое мастерство: безопасно входят в роль, безопасно выходят из роли, удерживаются в роли продолжительное время. И никогда не забывают, что можно сменить роль, что это «игра по роли». **Вася Пупкин, вообразивший, что он не играет Принца Гамлета в пьесе Шекспира, а что он и есть Принц Гамлет — это псих. При смене рассуждения про роль «Принц Гамлет» Васи Пупкина на рассуждение о роли «инженер-архитектор» или «врач-кардиолог» Ксении Самойловны тут принципиально ничего не меняется. Если Ксения Самойловна забудет, что она на работе играет роль инженера архитектора и вообразит, что она и есть везде и всюду инженер архитектор (персонаж ролевой игры или пьесы, роль), то мы её сочтём сошедшей с ума.**

Пётр Петрович играет роль мужа в проекте одной системы, покупателя в проекте, связанном с другой системой (и в этот момент он может как продолжать играть роль мужа, если покупка связана с ролью мужа, так и муж может на это время исчезать — так же, как исчезает Принц Гамлет, пока играющий его актёр играет роль Отелло в другом спектакле, или даже просто спит дома, отдыхая от театра), пациента в проекте, связанном с третьей системой. Если Пётр Петрович реально перевоплотился в Принца Гамлета, то это в актёрской профессии признак того, что Пётр Петрович сошёл с ума и путает себя и Гамлета. То же самое, если человек на работе забывает, что он играет роль инженера или менеджера: он сошёл с ума, если считает, что он и есть инженер или менеджер! Роли играют, в них не перевоплощаются!

Проектная роль, деятельностная роль, функциональная роль, организационная роль, трудовая роль, командная роль, культурная роль, просто роль: вы называете роль какого-то агента с разными эпитетами, но это не так важно. Это всё синонимы, просто разными уточнениями вы будете акцентировать те или иные оттенки значений, тут синонимия того же рода, как синонимия методов/практик/культур работы. Об этой синонимии мы ещё не раз напомним. И роль сама по себе не так уж и важна. Важно, что функциональный объект рассматривается в момент его функционирования. Роль — это система как функциональный объект, для функционального объекта важно, какие изменения производит роль в окружении, то есть что эта роль делает, какую функцию/метод/практику работы эти роль выполняет. Роль всегда рассматриваем в момент её работы/функционирования, оцениваем изменения, которые её поведение делает в окружающем мире (то есть оцениваем, например, по какому-то чек-листу, какие объекты в окружении системы-роли поменяли какое одно своё состояние на какое другое состояние).

Моделирование: аннотирование типами с использованием нотации ::

Для предложений из таблички сделайте дубль, указав типы всех объектов и отношений, как показано в первой строке.

Номер п/п	Без типов	С типами
Пример	Летит самолёт.	Летит::функция самолёт::система.
1.	Молоток бьёт по гвоздю.	
2.	Вася работает инженером	
3.	Компьютер состоит из корпуса, материнской платы, клавиатуры и экрана.	
4.	Роза — это цветок.	

Вопросы. Конфликт интересов

Отметьте правильное утверждение:

--- Вася хочет покрасить девайс в розовый цвет, а Катя — в голубой. Между ними конфликт интересов. ### Нет, это не конфликт интересов, ибо речь идёт о разных агентах, а не одном агенте.

--- Вася обещал Кате покрасить девайс в голубой цвет, и обещал Вере не красить девайс, обещал руководству, что он покрасит по итогам исследования (оно показало: надо красить в зелёный), а у него самого предпочтение красить в розовый. У руководства Васи конфликт интересов. ### Нет, конфликт интересов у Васи, у руководства единственный интерес: красить в зелёный (ибо так показало исследование).

+++ Вася хочет покрасить девайс в розовый цвет, но исследования говорят, что надо бы красить в зелёный. Вася должен красить девайс прямо сейчас, но он в раздумьях: у него конфликт интересов. ### Да, Вася в роли «любителя розового»

и Вася в роли «реализатора результатов исследований» имеют разные предпочтения к одной характеристике, цвету девайса. Разные предпочтения к одной характеристике у одного и того же человека — это и есть конфликт интересов.

Роли аналитиков и инженеров. Должности

Роли создателей выполняют самые разные агенты: люди, люди с компьютерами и станками, системы AI, предприятия. Все они меняют мир к лучшему, создавая и развивая те или иные целевые системы. Создатели — это деятели, а не по-бытовому понимаемые «наблюдатели-свидетели», как в физике! Даже в физике уже пришли к выводу, что не бывает не меняющих мир наблюдателей. Чтобы что-то «пронаблюдать», то есть измерить, надо измерительному инструменту и измеряемой системе провзаимодействовать.

В физике наблюдателем может быть вообще не агент в узком смысле (человек или AI, которые могут планировать), но в рамках минимального физикализма — какая-то молекула, которая измеряет своё окружение[190]. Но дальше мы будем говорить о тех ситуациях, когда «наблюдателем» или даже «аналитиком» называют активного агента, который проводит измерения («наблюдает») и даже делает какие-то суждения о ситуации, но не предпринимает мер по изменению ситуации к лучшему. Помним, что у наблюдателя или аналитика результатом наблюдения (хотя он сам для себя может быть очень деятельным, например, залезть на высокую сосну, чтобы поглядеть оттуда на мир через бинокль) будет какое-то описание, «аналитический отчёт», месяц интенсивной работы аналитика. В этом отчёте обычно есть оценка ситуации, но нет стратегии (метода работы, который поможет улучшить ситуацию), а тем более — нет конструктивного (выбор ресурсов-аффордансов, реализующих стратегию) плана действий по изменению ситуации (часть отчёта «рекомендации» редко включается в «аналитический отчёт», чаще «приговаривается голосом» или даётся отдельным документом, но неформально, не входит в работу аналитика), а главное — кроме отчёта и опциональных рекомендаций по выбору стратегии как варианта какого-то метода работы как основы для составления плана действий/работ с реалистичными ресурсами, нет самих активных действий агента по изменению ситуации (то есть нет самих действий по физическому изменению состояния систем, возможно, систем из графа создателей) к лучшему, всё ограничивается «прожектами»: описаниями возможного будущего без активной работы по претворению этих описаний в жизнь, да и сами описания возможного будущего вполне могут быть утопичными, а ответственность за реализацию утопий вроде как оказывается не у предложившего эти утопии аналитика.

Наблюдатели/аналитики обычно ничего не меняют в мире, хотя у них может быть предмет их ролевого интереса (характеристика, какую они почему-то считают важной у тех объектов, за которыми они наблюдают) на уровне «любопытненько», и даже могут быть предпочтения в этом предмете интереса, но агент-исполнитель роли наблюдателя не имеет намерения быть не просто наблюдателем, а что-то менять в наблюдаемой им ситуации к лучшему. Намерения реального изменения у просто наблюдателя нет. На деятельность в проектах наблюдатели смотрят «как

[190] Мы уже приводили эту работу, но напомним (и дальше читайте по ссылкам из этого текста): "Minimal physicalism as a scale-free substrate for cognition and consciousness", 2021, от Chris Fields, James F. Glazebrook and Michael Levin https://chrisfieldsresearch.com/min-phys-NC-2021.pdf

происходящее в телевизоре». От аналитической работы, выполняемой специально отряженными для неё агентами-аналитиками обычно ничего в проекте не меняется, плюс добавляется испорченный телефон при пересказе мыслей реальных деятелей сторонними наблюдателями/зрителями/аналитиками. В футболе есть множество людей, которые с любопытством смотрят на игру по телевизору, но результаты игры зависят только от тех, кто бегает по полю и пинает мяч, а так же тех, кто «делал» этих игроков (тренеров, массажистов, спортивных врачей). Конечно, многие из этих «трибунных/диванных аналитиков» лучше знают, что делать пинающим мяч, а также как готовить команды! Но их мнение мало на что влияет.

В этом плане «аналитики» обычно похожи на бытовых «болельщиков»: им ничего не нужно сделать, чтобы предъявить сделанное в физическом мире, им только нужно что-то понять и приготовить «объективный отчёт». В редких случаях они ещё и делают «предложение» для последующего «утверждения». Вот эти «предложения», которые потом кто-то «утверждает» — это верный признак, что мы попали на аналитика, а не инженера/деятеля. Инженер выпускает «проектную документацию», он не «делает предложение по выпуску», которое должен утвердить потом какой-то другой инженер как «начальник»! Если дело проходит через выделенного аналитика, а не инженера, то:

- У аналитика нет «шкуры на кону», он только «предлагает», то есть его предложения качеством похуже, чем у инженера-деятеля, который ставит свою шкуру на кон[191]. При этом аналитик может называться «инженер» по должности, но мы же смотрим на метод работы, «что же он делает, что меняет состояния в его работе, каким способом это изменение состояний происходит», чтобы назвать роль, а не смотрим на название должности или самоназвание. Только метод работы — критерий для выявления роли! Метод «подготовки предложений», а не «выпуска принятых инженерных решений», не метод «проектирования» — всё, перед нами аналитик. Вы делали какие-то «предложения», которые потом кто-то утверждал перед реализацией? Всё, вы — аналитик.
- Аналитик часто смотрит внимательно на какую-то ситуацию, а затем передаёт её суперсжатое описание (модель) разработчикам-деятелям. Тут проблема испорченного телефона: как описание ситуации может существенно искажать ситуацию, так и чтение модели ситуации тоже может быть выполнено с ошибками. Разделение труда важно, но «аналитики» вроде «инженеров по требованиям» в жизни исчезли, потому как проблема испорченного телефона оказалась серьёзной (подробней об этом будет в руководстве по системной инженерии).
- Это дико медленно, потому как связано с каскадированием, когда добавляется лишний такт времени ожидания на коммуникацию начальников и подчинённых на нескольких уровнях. «Предложение-утверждение» вместо самостоятельного «выпуска решения» на четырёх уровнях, со временем рассмотрения предложений одна неделя — и всё, добавляйте три недели в проект. Если приказано «дать предложения» и это спущено вниз, то добавьте для четырёх уровней ещё три недели, всего шесть недель. Четыре уровня — это генеральный, его зам, начальник отдела и сразу исполнитель. В крупных организациях каскадирование добавляет ещё больше времени, поэтому там

[191] Нассим Николас Талеб, «Рискуя собственной шкурой. Скрытая асимметрия повседневной жизни» (2018).

всё так медленно.

А иногда ещё аналитиком декларируется отсутствие предпочтений в каких-то интересах («объективный», исследовательский взгляд) — у исполнителя роли аналитика обычно нет деятельностного намерения менять жизнь в сторону реализации предпочтения, нет сопряжённой роли **синтетика**, если не самостоятельно реализующего решение, то хотя бы синтезирующего описание решения. **Синтез** — это обратная декомпозиции/анализу операция, обычно означает проектирование каких-то собираемых в систему аффордансов, то есть изобретение, а затем изготовление самой системы «в физике» согласно синтезированному проекту/design. Инженерия тем самым включает и анализ, и синтез, причём без синтеза — не инженерия! **В нашем руководстве по системному мышлению мы ничего не говорим о «системном анализе», и это намеренно! Руководство это не для аналитиков, которые описывают мир, а для деятелей: инженеров-менеджеров самых разных систем (киберфизических, организмов, личностей, коллективов, сообществ) которые меняют мир к лучшему!**

Аналитик в его классическом виде «создателя аналитического отчёта» может забрать время и ресурсы, чтобы «наблюдать», ему нужно как-то проводить «измерения», опрашивать людей. Скажем, сбегать с трибуны и опрашивать игроков, на футбольном поле. Это даже может принести пользу: он случайно может спросить о чём-то таком, что не приходило в голову тому игроку, кого он будет расспрашивать — и случайная польза будет нанесена! Но много чаще — не нанесена, если за этим «анализом» не последует «синтеза». Но если есть и «синтез», то роль «аналитиком» не назовут, будет другое название.

Если хотите сделать из человека-актёра, играющего роль аналитика, настоящего проактивного актёра-деятеля/практика, просто дайте ему **должность** инженера. Должность — это именование оргзвена, которое выполняет самые разные деятельности, но при этом должность подразумевает какое-то распоряжение ресурсами этой должности (скажем, должность директора предприятия предполагает выполнение самых разных ролей, но кроме того и распоряжение ресурсами предприятия).

Обычно достаточно просто переименовать «аналитика»::должность в «инженера»::должность. Иногда полезно детализировать культурно-обусловленное название инженерной или менеджерской роли для должности: давать бывшему аналитику::должность новую должность «инженера-архитектора»::должность или «инженера-разработчика»). После этого инженер (бывший аналитик) начнёт не «предлагать» что-то людям, принимающим на себя его ответственность за результаты его работы, но сам «решать», то есть принимать инженерные решения, выпускать их от своего имени (а не имени начальника), брать на себя ответственность за реализацию этих решений. Должности — это про ответственность (что бы это слово ни значило, про значение «ответственности» нужно каждый раз договариваться отдельно). Ещё должности — это механизм поднятия статуса того, кто на должности. Скажем, покупатели не хотят иметь дело с продавцом::должность, они требуют для общения с собой менеджера, начальника продавцов. С этим легко справились, просто переименовали «продавца»::должность в «менеджера по продажам»::должность. Если это консалтинговая фирма, то «менеджера по продажам» можно легко переименовать в «вице-президента» (так, например, сделали в консалтинговых подразделениях

IBM, вы вряд ли найдёте где столько вице-президентов, сколько в IBM). Никогда не смотрите на должности, смотрите на выполняемые на этих должностях роли, а роли находите по методам/способам работы. Что агент делает — это и есть выполнение его метода работы, игра его роли. Старайтесь должность именовать по основному методу работы основной роли, на выполнение которой берётся агент на эту должность, а «ответственность», «уровень вознаграждения», «квалификацию агента на должности» можете выражать словами «главный», «старший», «ведущий» и прочими не обязывающими эпитетами.

Не путаем должности и роли: человек на должности инженера может сутки напролёт играть роль менеджера, планируя работы, и наоборот, человек на должности менеджера может сутки напролёт разбираться в инженерных задачах. Элон Маск сообщает о себе, что 70% времени занят решением инженерных задач, то есть играет роль инженера-технаря (инженера киберфизических систем, в просторечии именно это чаще всего и называют «инженер»)! Элон Маск — это агент/актёр, и оказывается, что роль основателя фирмы как инженера вновь создающейся организации и даже менеджера-организатора и операционного менеджера как инженера уже созданной организации (он ведь CEO и Tesla, и SpaceX, некоторое время был CEO Twitter/X) им выполняется меньше времени, чем роль инженера киберфизических систем!

Должности описываются должностными инструкциями, а вот роли — описаниями методов работы, которую выполняют роли. Методы работы описываются чаще всего учебниками/регламентами/стандартами, включающими и описание знаний/теорий и инструкции по задействованию инструментария метода. По какому учебнику (есть варианты отраслевых и корпоративных стандартов, инструкций и регламентов, форма тут не важна — это всё у нас будет «учебник» работы по какому-то методу/способу/практике/культуре) строит своё мышление человек, который находится на должности инженера? Если по учебнику «операционный менеджмент», то он играет роль операционного менеджера, а должность нужна будет только для того, чтобы вовремя с точки зрения администрации назначать ему зарплату. Должности важны, когда меряешься пузом: властью, полномочиями по распоряжению трудом своим и чужим, оборудованием и прочими ресурсами. Но если речь идёт о том, как именно человек раскидывает мозгами, то должности сразу идут побоку и должно быть разбирательство с ролями. **Думают и действуют агенты как актёры в ролях, то есть роли, а должности только «реализуют полномочия».**

Если агент в должности «аналитик» говорит, что «моё мнение таково, что в ракете должно быть три ступени, а не две» (замаскированное слово для «предложения» к последующему «утверждению»), то кто-то::«агент на должности»/оргзвено должен будет по этому мнению принять какое-то решение. Если инженер::должность говорит «в ракете будет три ступени, а не две», то это и есть принятое решение, не нужно дополнительного утверждения каким-нибудь синтетиком::должность. А для аналитика нужно чьё-то утверждение для его «предложений». Чувствуете разницу ситуаций? «Предложения» против «принятых решений»: как вы лодку назовёте, так она и поплывёт, одни и те же утверждения будут иметь разный распорядительный статус в зависимости от того, какая культурно-обусловленная роль в какой культурно-обусловленной должности будет их делать. Культурно-обусловленная — это взятая из расхожих мемокомплексов, а не специально придуманная. Если вы придумали, что «аналитик» у вас — это полноценный инженер, то не обманывайте

себя, это как назвать «белое» полноценным «синим». Вы об этом будете знать и даже сообщите сотрудникам, но культура перемелет все эти ваши именования к завтрашнему утру.

Роль аналитика при сопряжении её с ролью принимающего решения синтетика в общей для аналитической и синтетической работы роли инженера становится полноценной проектной ролью. Если «аналитик» что-то предлагает, как рекомендацию — это уже не «аналитик», это «синтетик», рекомендацию-то нужно создать, принять инженерные решения (а не только выполнить понимание), и это не аналитическая работа, а синтетическая!

Должности часто выводятся из типичных для них ролей. Если должность «аналитик», то не ожидайте от людей на этих должностях синтетической работы. Если она и будет, то «вопреки», в нарушение культурно-обусловленных соглашений. Это не означает, что нет такого метода, как «системный анализ», но выполняющая его роль «аналитик» лучше бы была назначена тому же человеку, который делает и системный/модульный синтез, и у этого человека-инженера были бы полномочия и ответственность по принятию решений синтеза, а не только «ответственность и полномочия по проверкам и пониманию». **Деятельностный подход: «понял/объяснил — делай, то есть принимай понятое/объяснённое всерьёз и меняй на этой основе мир!»**.

Если человеку-актёру, находящемуся в проектной/деятельной роли, что-то в системе или проекте, который занимается этой системой, не нравится, или наоборот, нравится (то есть у него есть важная характеристика, для которой у него выполняются или не выполняются ролевые предпочтения) — он начинает что-то проактивно предпринимать, «исследовать делом», «активно зондировать», но он не просто «активно наблюдает/измеряет», он реализует свои предпочтения, для чего делает попытки изменения ситуации, задействуя метод своей роли. Вдобавок актёр может и переключаться между разными ролями, реализуя какую-то стратегию/деятельность по реализации намерения[192]. А если актёр, играющий какую-то роль, ничего не предпринимает для реализации своего интереса/предпочтения в важной характеристике как предмете интереса — то его просто не учитывают в проекте. Он прохожий, случайный для проекта человек, у него тогда нет проектной роли, нет выполняемых по роли действий, он не трудится в каком-то из проектов цепочки создания, затрагивающей целевую систему или систем в графе создателей целевой системы. Если кто-то просто проходит мимо окна конструкторского бюро, то он явно не играет никакой проектной роли в проекте, над которым работает это конструкторское бюро, хотя мы вполне можем видеть такого человека (особенно, если у нас нет собранности, и внимание блуждает по случайным объектам). Но если мимо окна проходит потенциальный покупатель системы, который может содействовать успеху проекта своими деньгами, то мы постараемся его заманить и учесть его интересы.

Обычно агенты (скажем, люди) занимаются какой-то своей привычной деятельностью/трудом в какой-то привычной роли, и тут в их жизни появляется (или может появиться) очередная система — кому-то эта система даёт новые возможности (например, даёт новые «фичи» пользователям::роль, или членам команды разработчиков::роль, которые рады новому проекту как возможности

[192] Тут мы обращаемся к «прагматическому повороту» в философии, https://plato.stanford.edu/entries/pragmatism/

продолжить развитие системы как бесконечной череды её непрерывно улучшаемых версий), кому-то она мешает (и эти люди будут играть роль конкурентов или сторонников какой-то политической или религиозной идеи — не сомневайтесь, они будут активно играть эти роли и изобретательно реализовывать свои предпочтения!). Агенты, в жизни которых появилась новая система-возможность или система-угроза начинают удерживать эту новую систему во внимании — и действуют, меняют мир, чтобы или использовать возможность, или устранить угрозу.

Конечно, материал про «аналитиков» против «инженеров» — это больше материал руководства по системному менеджменту, но он достаточно иллюстративен, чтобы мы показали именно на этом примере отношения между методами работы, ролями и должностями. В любом случае, для должности и даже для роли верно высказывание Козьмы Пруткова[193]: «Если на клетке слона прочтёшь надпись «буйвол» — не верь глазам своим». А чему верить? По делам (методам работы, какими делаются дела) и используемой терминологии в этих делах узнаем роль. А должность — ещё непонятно, какие реальные полномочия у носителя громкой должности. Вполне возможно, что даже какой-нибудь «главный деятель»::должность не полномочен что-то делать, то есть распоряжаться ресурсами, а он только «предлагает» для «утверждения». Помним, что в костюмах сейчас ходят охранники, а те, кого они охраняют, ходят в маечках. Так что не встречайте по одёжке, не судите о людях по названиям их должностей и даже по называемым ими ролям. Судите по методам/культурам работы, оценивайте использование терминов теории/дисциплины/знаний методов в речи.

Моделирование: аналитика и инженерия

Отмоделируйте свою работу, разделив её на работу по методу «анализ» в роли «аналитик» и методу «инженерия» в широко понимаемой роли «инженер» (инженер — это одновременно и аналитик, и синтетик). «Широко понимаемой роли «инженер»» понимать как: менеджер — это инженер организации, педагог — инженер личности, врач — инженер организма, и т.д. Для работы в вашей роли аналитика впишите процент времени, который вы делаете что-то, что не ведёт к вашим «предложениям», которые становятся «решениями», принятыми другими людьми, у которых есть на это «полномочия» (помним, что инженеры принимают свои решения сами, а аналитики только делают предложения, полномочий что-то поменять в мире у них обычно нет). Отмоделируйте и вашу инженерную работу: какой процент времени меняете своими решениями окружающий мир, при этом все понимают, что это ваши решения, ваша ответственность и не вмешиваются в них (хотя и принимают активное участие в обсуждениях-согласованиях, все инженеры согласовывают свои решения, но это не «разрешительные» согласования у начальства, а содержательные согласования с коллегами-инженерами). Запишите эти цифры в табличку.

Процент времени работы аналитика	Процент времени работы инженера

[193] https://ru.wikipedia.org/wiki/Козьма_Прутков

Ответьте на вопросы. Если вы политик, чьи решения меняют мир, или менеджер, чьи решения меняют мир, или фермер, чьи решения меняют мир, и так далее, то мы в целях данного упражнения тоже будем называть вас «инженером». Решения «аналитика» не меняют мир, они информируют других людей о мнении аналитика (и ещё хуже, они информируют других людей о мнении людей, у которых аналитик брал интервью, а не о результате мыслительной работы самого аналитика! Это шанс получить «испорченный телефон»).

Вопрос	Ответ (в произвольной форме)
Вы себя считаете аналитиком или инженером?	
Ваш начальник вас считает аналитиком или инженером?	
Ваши коллеги считают вас аналитиком или инженером?	

Внешние и внутренние проектные роли

«Говорю система — подразумеваю наличие создателей::система и их проектных ролей, создающих эту систему, и создателей::система, создающих эту систему в составе каких-то надсистем. Говорю проектная роль — подразумеваю систему, которая создаётся или эксплуатируется этой ролью» — это самые азы системного мышления (система в её окружении) в его стыке с методологией (система-создатель, «работающая по какому-то методу»/«каким-то способом»/«в каком-то стиле»). Можно уточнять дальше, какие именно системы (целевые, создатели где-то в графе создателей, надсистемы, подсистемы), какие проекты создания этих систем приходится рассматривать, но сути рассуждения это не меняет.

Проектная роль (исполняемая агентом-конструктивом для этой роли) логически в этом рассуждении появляется раньше, чем появляется система: если нет агента, который играет какую-то роль, чтобы осуществить своё намерение, то систему просто некому выделить своим вниманием из объектов окружающего мира. И наоборот, если система выделена как объект из многообразия объектов окружающего мира, то для чего и кто провёл её границы в мире именно таким образом, а не каким-то другим? Для чего он вообще заинтересовался системой, какие характеристики системы важны такому агенту, для задействования какого метода в какой роли этого агента?

Конечно, проектные роли играют не любые люди/агенты/актёры, которым система «интересненька», не более чем «любопытна» (как наблюдателям, но без желания пальцем потом шевельнуть). Нет, проектные роли — это те, которые в принципе будут действовать в физическом мире, если им эта система нужна (или наоборот, мешает какими-то своими характеристиками). Эти роли исполняются реальными агентами (людьми, организациями, AI-агентами) меняющими в конечном итоге физический мир, а не только «мир воображения»/«ментальный мир»/«мир описаний». Как «наблюдатель» из по-бытовому понимаемой физики не меняет мира, он не деятель/практик, так и другие «наблюдающие зеваки» — это не проектные роли, они просто прохожие, их не нужно делать объектами внимания в проекте. Проектные роли — это «действующие лица», они оживают, когда их

играют актёры как «исполнители»! И они влияют на целевые системы и системы создания, важные характеристики которых их интересуют. Они обязаны выявляться в проекте и удерживаться вниманием в ходе рассуждений о системе и проекте её создания.

Собаки лают, а караван идёт: собаки тут не проектные роли, они не влияют на караван. А вот если купец не оплатит проход каравана, то караван идти не будет. Купец — проектная роль, он занимает деятельностную/активную позицию по отношению к каравану. Проходящие мимо агенты — это не агенты, имеющие проектные роли. Проектные роли — это агенты, которые мимо не пройдут, это те агенты, чьи роли проект обязательно зацепит или которые обязательно зацепят проект.

Обычно с системой связано много проектов, необязательно это ровно один проект в смысле классического проектного управления, то есть работа одной команды. В одном проекте одной команды систему могут замыслить, в другом — спроектировать, в третьем — изготовить, в четвёртом — эксплуатировать, потом перепродать, в пятом — продолжать эксплуатировать, потом в шестом — модернизировать, в седьмом — продолжать эксплуатировать, в восьмом — ликвидировать, но потом всё-таки обнаружить, что проект по развитию системы надо продолжать — и появится девятый проект. Проектная роль может быть в любом из этих проектов, например, мы уже при замысливании и создании системы учитываем проектные роли людей, которые будут потом систему эксплуатировать: роли и подроли операторов, роли и виды ролей пользователей системы. Если мы делаем видеоигру, то роль игрока мы учитываем сразу, хотя до выпуска игры может быть ещё много лет — да и непонятно, кто именно (какой агент) будет играть эту роль, игроков вообще могут быть потом миллионы.

В системном мышлении регулярно мы думаем о будущем «как будто оно уже существует», это нормально для 4D экстенсионализма. Говорить, что «я не знаю, какой будет система, ибо системы-то ещё нет» — это игнорирование инженерного опыта человечества. **Все проекты по созданию систем начинаются с того, что системы ещё нет, а о ней уже думают!**

Думать о том, как (методы/«виды труда»/«рабочие процессы») и кем (какими ролями, актёрами в каких ролях) будет эксплуатироваться система, которой ещё нет, которая едва-едва замыслена в её MVP — это не только нормально, это просто необходимо, с этого вообще нужно начинать думать о системе! Если система не будет кому-то нужна, не будет кем-то эксплуатироваться/использоваться, то зачем её вообще делать?!

А если будет эксплуатироваться, то кем, какими проектными ролями, кому она и для чего нужна, какие её важные характеристики для этого? Личность исполнителя роли (какой именно агент будет играть роль) не нужна, она тут не так важна, но саму роль нужно указать! Грубо говоря, вы должны рассказать всё о Принце Гамлете в спектакле или хоббите в ролевой игре ещё до спектакля или самого игрового мероприятия, и неважно, кто именно потом будет исполнять эту роль («исполнитель»), черты характера актёра или игрока, а не «действующего лица»::роль. Роль много важней, чем её исполнитель. Хотя мастерство исполнителя роли важно, да. Но хуже будет, если это будет мастерство исполнения вообще не той роли! Помним анекдот про голубя, логопеда, гастроэнтеролога: мы даже не можем понять уместность нахождения какого-то агента в проекте, если неизвестно,

что должен делать агент, то есть неизвестен метод работы и поэтому неизвестна роль агента.

Проекты создания и развития систем — это обычно коллективные проекты, и в этих проектах обычно по поводу систем договариваются так, чтобы максимально удовлетворить интересам проектных ролей.

Граница системы — это граница места в физическом мире, которое занимает система в момент эксплуатации/использования/функционирования. Граница системы определяет, какие части входят в систему (и поэтому их необходимо спроектировать и изготовить в проекте создания и развития системы), а какие не входят в систему, а входят в системы окружения (и поэтому можно ожидать, что они уже развёрнуты какими-то другими агентами, не входящими в команду проекта).

Вот эта граница системы прежде всего и определяется командой агентов (людей или даже предприятий со всеми их компьютерами и заводским оборудованием), играющих проектные роли создателей системы. Лучшее культурно-обусловленное понимание состава проектных ролей даётся в руководстве по системной инженерии. Там говорится о ролях визионера, разработчика, архитектора, прикладного методолога, создателя внутренней инженерной платформы — и у каждой роли есть методы, которые они задействуют в работе. Конечно, для разных видов систем («железные», программные, организмы, личности, организации, сообщества) используются разные виды/варианты методов создания и развития систем, поэтому отличаются и роли. Например, для создания мастерства::система нужны культуртрегеры, методологи, методисты, преподаватели/наставники, а ещё нужен деканат.

Внутренние проектные роли — это проектные роли, которые играет **команда проекта**, а **внешние роли** — все остальные, прежде всего заботящиеся о внешних по отношению к границе целевой системы системах, то есть системах в рабочем/операционном окружении целевой системы и системах в графе создателей целевой системы. Скажем, внешние проектные роли могут быть в проекте создания интерьера, в состав которого входят декоративные часы, а внутренние проектные роли — в проекте создания самих часов. Внешние проектные роли в службе маркетинга могут создавать клиентуру для проекта создания декоративных часов — приводить команды самых разных разработчиков интерьеров.

В англоязычной литературе часто для внешней проектной роли используется термин stakeholder (и в русском — калька «стейкхолдер»), но будьте осторожными:

- Stakeholder иногда означает только внешнюю роль, иногда — любую проектную роль.
- Stakeholder иногда означает именно роль, но часто — агента-исполнителя роли, то есть путают Васю Пупкина и играемого им Принца Гамлета. Иногда поэтому уточняют: stakeholder и stakeholder role, иногда для исполнителя роли stakeholder representative, особенно когда этих исполнителей ролей много, например, есть 1000 пользователей какого-то устройства, так вот ответственного за выражение интересов пользователей назовут stakeholder representative. Но часто уточнений не делают, и тогда Принца Гамлета начинают спрашивать, как ему спалось между спектаклями, а Васю Пупкина — что он думает о «Быть или не быть, вот в чём вопрос!». Эта путаница с понятием stakeholder очень часта, поэтому в нашем руководстве

мы избегаем этого синонима проектной роли.

Команда проекта определяет, что (какие части) в целевой системе нужно, а что (какие части) не нужно иметь в её составе, чтобы система могла выполнить свою роль/функцию/назначение в окружении и не стоила при этом запредельно дорого (больше, чем могут заплатить за её создание внешние роли), а затем изготавливает эти части и налаживает совместную работу изготовленных частей. А внешние проектные роли определяют функции/назначение системы в её окружении. Обычно ваши внешние проектные роли — это какие-то проектные роли внутри организации-заказчика, но есть и роли «без организации» (скажем, игрок компьютерной игры, или хакер-одиночка, они тут будут сами по себе) и роли из других организаций (скажем, «безопасник» из какого-нибудь госнадзора). Вы, конечно, можете переименовать их так, как вам нравится, но мы не рекомендуем это делать, то есть рекомендуем не переназывать роли «финансист», «разработчик», «безопасник» в организации-заказчика одной ролью «пользователь» в вашем проекте. Правильно так и оставить «финансист», «разработчик клиента», «безопасник клиента» (это отличит его от «безопасника госнадзора»). И помним, что кроме ролей в организации-заказчике/клиенте в проекте надо бы учитывать и многие другие роли — регуляторы, представители широкой публики, и даже хакеры, воры, террористы и прочие злоумышленники (их интересы надо тоже учитывать, но с обратным знаком: чтобы они не смогли реализоваться).

Все исполняющие эти внутренние и внешние проектные роли люди/организации/агенты/актёры/оргзвенья (а хоть и каждое оргзвено будет исполнять несколько ролей, или одну роль будет играть несколько оргзвеньев) проявляют находчивость и изобретательность в вопросе удовлетворения своих интересов — исходя каждый из своих агентских намерений, занимая какие-то проектные роли (ещё и в разных проектах), и договариваясь о важных/интересных для методов работы этих ролей характеристиках системы и проекта её создания и/или эксплуатации.

Система в глазах смотрящего: если никто::роль не смотрит, т.е. ни в чьём методе работы система оказывается не нужна, то и нет системы, нет её границы, нет назначенного/ролевого поведения/функции. **Система субъективна и существует как объект внимания только в рамках договорённостей конкретного проекта**. Инженерный проект существует как непрерывные переговоры внутренних проектных ролей, а также внутренних и внешних проектных ролей по поводу достижения согласия в понимании того, какую систему делают участники проекта и как эта система своим функционированием в момент эксплуатации удовлетворяет интересы внешних проектных ролей. Это неважно, что агенты, выполняющие роли, как-то меняются в ходе проекта: Вася-инженер уволился, его заменила Маша-инженер, всё продолжается. А описание проекта в терминах инженерных методов работы и роли инженера — оно осталось, оно не меняется в зависимости от смены конструктива.

Одни агенты в одних ролях договариваются о границах системы как «А и Б входят, В не входит», а в других ролях в том же проекте, или в другом проекте те же люди/оргзвенья/актёры/агенты вполне могут выделить систему из окружающего мира своим вниманием в других границах, «А и В входят, Б не входит», а также требовать других значений для важных характеристик системы, если это лучше отвечает их интересам в этих ролях. **И это нормально, тут нет никаких противоречий, это просто начальный этап коллективной деятельности:**

всем проектным ролям нужно договариваться о том, какова же система, которую мы замышляем, создаём, эксплуатируем, модифицируем, ликвидируем. А поскольку в ходе проекта все его обстоятельства непрерывно меняются, то договориться «раз и на всегда» не получится, надо постоянно договариваться! Договариваться по учёту своих интересов в том числе и в вопросах того, что вообще является системой, создаваемой и развиваемой в проекте, придётся ролям, даже исполняемым одним человеком-актёром. Поскольку ролей всегда много, то придётся постоянно принимать какие-то решения по согласованию интересов этих ролей по важным для них предметам интереса/характеристикам системы и особенностям ведения проекта по созданию и развитию системы.

Никакие призывы к «объективности» тут не помогут, придётся договариваться с учётом того, что предметы интереса у проектных ролей обычно общие, а вот предпочтения в значениях характеристик — разные. Одному нужно, чтобы система ездила побыстрее (характеристика: скорость, предпочтение: побыстрее), другому нужно, чтобы система была полегче (характеристика: вес, предпочтение: полегче). Первый для удовлетворения своего интереса планирует оборудовать систему мощным аккумулятором, который имеет большой вес — а второй понимает, что ему нужно срочно действовать, чтобы этого не случилось! Они немедленно начинают договариваться. Все проекты — это непрерывные договорки разных ролей друг с другом. Ролей, поэтому устранение особенно строптивых сотрудников часто не поможет: самые добрые сотрудники принимают на себя роль уволенного из проекта — и оказывается, что ничего не изменилось, приходится всё равно учитывать ролевые интересы (интересы Принца Гамлета, а не Васи или Пети, которые играют роль).

Не договорились — и проект немедленно разваливается. Договорились — и проект успешен (помним, что успешная система как раз и определяется как «по поводу этой системы все договорились»).

Успешность системы определяется сегодня не «объективно», а субъективно — по удовлетворению потребностей (needs) заказчиков, пользователей и самых разных других проектных ролей, включая договаривающихся с ними внутренних ролей команды проекта[194]. Агенты (люди, а также AI-агенты и самые разные их организации), играющие самые разные роли по отношению к системе и проекту по её созданию, должны быть довольны — успех в этом. Нельзя узнать про успех системы, глядя только на саму систему. Как система (её границы!), так и её успех — в глазах смотрящего. Разные агенты и даже одни и те же агенты (в разных проектных ролях!) могут быть разных мнений по поводу успеха системы. Так что задача создания системы прежде всего — это договорить между собой проектные роли (выглядеть это будет как договорённости исполнителей ролей: «Маша договорилась с Сашей», но правильно понимать, что это «инженер договорился с менеджером», желательно такое обсуждать в ролях, а не в лицах). Задача тут — предлагать такую систему или такой метод работы, по поводу которого им не нужно будет договариваться, ибо они сразу будут

[194] http://sebokwiki.org/wiki/Systems_Engineering_Overview — Systems engineering (SE) is an interdisciplinary approach and means to enable the realization of successful systems. Successful systems must satisfy the needs of their customers, users and other stakeholders.

реализовывать предпочтения всех ролей.

Системный подход, системное мышление, начиная со второго поколения — это и про создающих системы агентов в их разных ролях по отношению к системе, а не только про саму создаваемую систему. Система создаётся как результат договорённостей всех этих агентов. Поведение агентов в проектах создания и развития систем не произвольно, а паттернировано/шаблонизировано, определяется их культурно-обусловленной (чаще всего понимаемой как «профессия») ролью.

Единственный вариант «объективности» — это хорошо организованная субъективность[195], когда все проектные роли (внешние и команда) договорятся о том, какова их система, и что они от неё получат.

Если при создании системы забыли о какой-то проектной роли, потеряли её, забыли пригласить кого-то, кто сыграет эту роль — не будет успешной системы: не будут обнаружены или спроектированы и изготовлены какие-то части системы, не будут выполнены работы по каким-то методам, нужным для успеха системы, и выяснится это уже после неудачи проекта.

Неудача проекта (то, что чьи-то интересы не учтены) обязательно обнаружится: этот недовольный «кто-то в роли» не просто наблюдатель, это деятель! Какой-то агент (один человек или огромная организация), чья работа по методу его роли затрагивается системой или проектом по её созданию и развитию, начнёт активно и изобретательно играть эту забытую командой проекта роль, и его действия станут хорошо заметны команде проекта. Даже если это «никто и звать его никак», исполнитель роли с неучтёнными интересами может найти влиятельных друзей, или организует кампанию в прессе: проигнорировать агента-в-роли обычно нельзя, агенты крайне изобретательны и могут ещё и перещёлкивать свои роли для того, чтобы добиться удовлетворения интереса какой-то одной из ролей.

Поучаствуйте в какой-нибудь деловой игре, где нужно импровизировать, придерживаясь ролей — вы удивитесь собственной изобретательности. В жизни всё то же самое, только на это (в отличие от игры) даже не обращают внимания: роли играют, не задумываясь над ними, а зря! Если понимаешь, что ты играешь роль, изобретательность «игры по роли», изобретательность «актёрских этюдов» можно включать прицельно и сознательно — и она обычно сильно содействует успеху.

Моделирование: внешние проектные роли и успешность системы и её проекта

Выпишите от 15 внешних ролей разных агентов (людей и организаций) вашего текущего проекта (если один человек или организация играют несколько ролей — выписывайте все). Укажите их интересы (предмет интереса и куда его предпочитает двигать роль). Проставьте ваше ожидание (по десятибалльной шкале) по выполнению выписанных ролевых предпочтений по итогам реализации проекта.

Номер по порядку	Внешняя проектная роль	Интерес	Ваше ожидание, что предпочтение роли будет соблюдено

[195] Высказывание Г.П.Щедровицкого.

Подумайте над свежезаполненной табличкой. Ответьте на вопросы.

Вопрос	Ответ
Ожидаете ли вы, что ваш проект будет успешным согласно определению успешности?	
Какая роль будет считать проект неуспешным (и даже вредным)?	
Можно ли сделать проект более успешным? Кто и что для этого должен сделать?	

Культурная обусловленность проектных ролей

Нас прежде всего интересуют приёмы мышления, и особенно интересует сохранение опыта — перенос опыта из ситуации в ситуацию, из проекта в проект. Мышление происходит не столько с фактами, сколько со знаниями: абстрагированными из фактов о конкретных физических объектах знаниями о самом важном. «Физическое тело» абстрагирует огромное число ситуаций, в которых действуют законы механики: от полёта космического корабля на Луну до броска мяча в баскетбольную корзину. «Целевая система» абстрагирует огромное число ситуаций, в которых создают и развивают какую-то систему — от проекта выпуска смартфона до строительства городского квартала.

Когда интеллект встречается с ситуацией, которую он раньше не встречал, ему нужно выделить какие-то опорные объекты внимания в этой ситуации, которые позволили бы ему зацепиться и попробовать какие-то привычные ходы мышления, дающие максимальную ожидаемую вероятную результативность, а не просто случайные ходы с заведомо низкой результативностью. Эти ходы мышления должны помочь находить объяснения происходящего в ситуации до того момента, когда можно будет получить новую информацию и откорректироваться. Системное мышление — часть этого общего знания методов интеллект-стека.

Знание о методах работы и задействующих их ролях (методология), нормативном разделении труда, то есть нормативном разделении инженерии на отдельные инженерные методы и выделение их для работы разных агентов в разных ролях в ходе создания и развития систем (системная инженерия) — тоже часть этого общего знания. Существенно усиливает это знание трудовой кругозор — понимание, как может быть для разных видов создаваемых систем устроено проектное разделение труда (деление инженерии систем какой-то природы на варианты методов этой инженерии и дальше назначение ролей, которые выполняют составляющие методы для этих вариантов инженерии, на разных агентов). Так, для «железной» и программной инженерии выделяют роли визионера, разработчика, архитектора, инженера внутренней производственной платформы. Для обучения (инженерии личности) — аналогичные роли, но которые работают по немного другим методам, характерным для создания и развития мастерства как целевой системы. Это роли

культуртрегера, автора учебных материалов, архитектора учебной программы, деканата. Для менеджмента (инженерии организации) — бизнесмена, организатора, орг-архитектора, администратора, у которых методы работают с организацией как целевой системой. Конечно, все эти роли вместе с методами их работы в порядке разделения труда дробятся и дальше, на ещё более специализированные как роли, так и методы.

Знание эффективного и результативного метода работы и надлежащего выполнения этого метода в профессиональной/деятельностной/проектной роли, опробованное с успехом самыми разными агентами во многих и многих проектах — оно культурно-обусловленное, то есть не придумывается заново в каждой конкретной ситуации. Системный подход напрямую обращается к знаниям цивилизации, то есть к культуре, накопленной человечеством. Термин «культура» имеет два основных значения:

- Метод работы, «культура как культурное поведение» в длинном синонимическом ряду «метод/способ работы»/культура/стиль/практика/деятельность/труд.
- совокупность человеческих (а сейчас — и более чем человеческих) знаний о мире и материальных/конструктивных объектов («материальная культура»), имеющих искусственное (то есть созданных людьми, а сейчас и роботами) происхождение.

Тут мы говорим о том, что системный подход использует накопленное человечеством знание о методах работы и работающих по ним ролях, а не придумывает всё с нуля в каждом проекте. Если не знаете, какие методы нужны для достижения каких-то результатов и тем самым агенты в каких ролях вам понадобятся — погуглите или спросите AI-помощника. Так, если вам надо кого-то накормить не в ресторане, а прямо на месте — это кейтеринг/catering, а если надо провести фестиваль — это менеджмент мероприятий/event management, и этому учат даже в вузах. Вы, конечно, можете что-то и сами сообразить, но это будет любительское кормление, любительский фестиваль. Отличаться будут тем, что профессиональное исполнение роли стабильно и при разных неизбежных неожиданностях будет давать результат, а любительское исполнение — не очень стабильно, иногда бывает ровно на том же уровне результата, что и профессиональное исполнение, а иногда при пустяковых изменениях в окружении — полностью провальным. Профи не делают новичковых ошибок.

Использование системного подхода предполагает какой-то деятельностный/трудовой/прикладной кругозор: если вы не знаете культуры, вы не увидите, какие именно роли играют по отношению к проекту люди — их важные характеристики покажутся вам неважными, а предпочтения покажутся вам блажью, личными особенностями агента, а не будут распознаны как цивилизационные, культурно-обусловленные, ролевые предметы интереса и связанные с ними предпочтения в состояниях этих предметов интереса. Если вы не знаете «из учебника», что роли разработчика и архитектора обычно конфликтуют, то вы будете удивляться, почему вокруг некоторых агентов, выполняющих для самых разных систем эти роли (и их должности будут называться совсем по-другому) вспыхивают «производственные конфликты», которые в зависимости от стиля общения будут перерастать в личностные или не перерастать, но для незнакомых с современной системной инженерией и менеджментом эти конфликты будут необъяснимы. А если вы знакомы с объяснениями современных методов разработки

и тамошних ролей, то вы будете знать, что делать. Если вы знаете, как из почти любого настраиваемого софта сделать универсальный моделер — вы будете знать, как в какой-то организации быстро и легко выполнить организационное моделирование. Если не знаете — успех будет только случайным.

Если вы встретили Васю::агента, который интересуется прочностью конструкции системы::характеристика (прочность конструкции:: «важная характеристика»/«предмет интереса»), то это вряд ли выполнение Васей менеджмента::метод/практика в роли менеджер::роль. Разговаривать с Васей о сроках::«предмет интереса» и бюджетах::«предмет интереса» может оказаться непродуктивно, не та проектная роль и её метод работы. И наоборот, если кто-то отказывается обсуждать особенности конструкции, но готов часами говорить о ресурсах для выполнения работ, стоимости этих ресурсов, примерных сроках окончания работ — это менеджер.

Трудового кругозора должно хватать для надёжного распознавания этих ролей в рабочих проектах, причём на довольно детальном уровне! Как вы не спутаете гинеколога и дантиста, считая их «просто врачами», так вы не должны путать разные виды ролей не только в медицине, но и во всех других методах/практиках/«культурах работы»/«видах инженерии»/деятельностях. Если у вас нет методологического/трудового/деятельностного/практического/культурного/стилевого кругозора, то у вас будут огромные трудности в системном мышлении: вы не сможете разобраться с ролями в вашем проекте.

Проектные (профессиональные, то есть по методам работы в проекте, а не по должностям!) роли не выдумывают. Они культурно-обусловлены. **Если не быть заведомо знакомым с трудовой/производственной культурой, то придётся тяжко: действия людей в проекте будут казаться непросчитываемыми, случайными. Но это не так. Действия людей (то есть работы по каким-то методам) происходят «в ролях», они абсолютно не случайны, их легко предсказать. Если в театре вы знаете пьесу Шекспира, то не удивитесь, что мавр кого-то там задушил после вопроса о молитве на ночь — мавр ведь не человек, а человек-в-роли, он ведь действует по роли, а не сочиняет сам свои действия «по наитию и ситуации». В его действиях есть паттерн, и он обязательно проявится. Если пьесы Шекспира вы не знаете — вы будете удивляться и самому появлению мавра, и далее каждой реплике.**

Деятельность/практика/труд/инженерия/метод отличается от случайных отдельных операций и действий с подвернувшимися под руку объектами, чтобы привести какие-то случайные их характеристики в случайно выбранные в качестве предпочтительных состояния. Деятельность/практика/труд/инженерия/метод — это целенаправленные в чём-то аналогичные/повторяющиеся/паттернированные действия/работы самых разных агентов, которых мы рассматриваем как актёров с ролевым (то есть типовым, по роли) целенаправленным поведением в соответствии с их предпочтениями в предметах их ролевого/функционального интереса.

Инженерия тут стоит в общем синонимическом ряду потому, что связана с изменением мира, то есть созданием новых или развитием/модификацией уже имеющихся систем. И ещё обратите внимание, что мы не включили сюда слово «работа» намеренно, и там синоним к работам не «деятельность», а «действия».

Эта разница между методами/практиками/способами/культурами (как что-то сделать) работ и самими работами (сам экземпляр делания с его ресурсами, в отрыве от «как») уже обсуждалась кратко в нашем руководстве по системному мышлению, но подробно она будет разъяснена в руководстве по методологии методологии.

Деятельность/практика/труд/инженерия — это синонимы для метода, способа выполнения работ, это не отсылка к использованию ресурсов в сам момент работы, это отсылка к паттерну поведения, «какие типы объектов как взаимодействуют, чтобы получить нужные состояния нужных объектов в результате». С работами связан «операционный менеджмент» как «эксплуатационная инженерия предприятия», а для неживых систем — эксплуатационная инженерия, где неважно, как именно функционирует система, но важно конкретное время её функционирования, сам экземпляр системы (если речь может идти о замене системы в ходе ремонта) и прочие аспекты управления активами/asset management, тоже метода менеджмента. Работы — это логистика и ресурсы (чтобы всё вовремя и не потерялось), методы работ — это чтобы всё правильно сделать, содержательное инженерное рассмотрение.

И ещё мы при обсуждении методов/деятельностей всё-таки говорим про создателей в их сложных взаимоотношениях в графе создания целевой системы, часто это люди, а с людьми всё сильно запутанно, агенты сложны в их описании в разговоре. Когда дело доходит до неживых целевых систем, разговор становится попроще, и вместо длинных синонимических рядов чаще всего говорят о функциях и функциональных объектах. Иногда функциональные объекты тоже имеют синонимы, скажем, часто их называют «теги»/tags, ибо на конструктивные объекты в «железной» инженерии часто навешивают бирки/tags с кодом функции (например, такие коды берут из ISO 81346).

Разговор про агентов как актёров, которые в своих ролях следуют методу (или играют практические/трудовые/инженерные роли — просто разные лексические способы сказать одно и то же), очень похож на ролевые игры, которыми увлекаются довольно узкие хоббийные сообщества «ролевиков»[196], хотя первая же ассоциация — это игра в театральных пьесах. Разница там главным образом в том, что в театральных пьесах все персонажи/роли/действующие лица говорят написанные автором реплики в заданные моменты (следуют пьесе/сценарию), а в ролевых играх персонажи/роли/действующие лица сами определяют свои действия и реплики, исходя из текущей ситуации и некоторых принципов (тот же сценарий, но определённый более свободно). В «Гамлете» Шекспира все реплики «Гамлета, принца датского» известны, а вот в ролевой игре «средневековая Дания» играющий принца Гамлета актёр/игрок сочиняет свои реплики сам, но исходя из обстоятельств ситуации. В любом случае его Гамлет не будет интересоваться ценами билетов на самолёт, чтобы улететь в Англию, это противоречит принципам отыгрывания роли в ролевой игре.

Принципы, которыми руководствуются для выполнения своего метода/деятельности/практики/труда те или иные проектные/деятельностные/организационные/трудовые/профессиональные роли прописываются в учебниках, им учат в вузах. Иногда эти принципы прописывают в международных, отраслевых, корпоративных стандартах. Иногда даже в

[196] https://ru.wikipedia.org/wiki/Ролевая_игра

регламентах, протоколах (например, врачебных), инструкциях для работы. Учат обычно не работе в должности, ибо работа в должности будет требовать выполнения множества ролей! Учат выполнению работы по определённому методу, игре по определённой роли, следование в какой-то роли какому-то сценарию/паттерну работы! А уж как эти роли потом раскладываются по должностям — это другой вопрос.

Агенты, выполняющие методы работы своих проектных ролей, не занимаются в проектах чем хотят, их действия не произвольны! Они, конечно, импровизируют — но их импровизация похожа на импровизацию актёров в театре: что бы они ни делали, в конечном итоге оказывается, что эти действия как-то обусловлены их ролью, не полностью выдуманы, они не могут быть «любыми». Если вам кажется, что инженер неправ в своих предпочтениях, то вы будете добиваться замены исполнителя этой роли — и совсем неудивительно будет обнаружить, что покладистый новый исполнитель роли, разобравшись в новом для себя проекте, вдруг заявит те же предметы интереса и ровно те же в них интересы, что и предыдущий исполнитель роли инженера! **Роль оказывается важнее личности исполнителя! По мере роста мастерства исполнения метода работы, используемого ролью, все агенты-исполнители этой роли с самыми разными личностями становятся похожи.** При низком мастерстве они просто не следуют методу работы, то есть не играют свою роль, поэтому делают много новичковых ошибок. Нильс Бор как раз определял профи как людей, которые не делают новичковых ошибок. И вот тут вопрос в том, что такое «новичковая ошибка»: в 17 веке врачу и повару (агентам при исполнении работы в этих ролях) необязательно было мыть руки или работать в стерильных перчатках, а в 21 веке — обязательно, немытые руки — новичковая ошибка, причём за такие ошибки надо вообще увольнять агента из профессии, не допускать к работе.

О методах/деятельностях/«видах труда»/практиках/занятиях/культурах/стилях работы мы думаем «в классах/типах», работой по определённым методам занимаются классы агентов (людей/организаций/агентов/актёров в их ролях), например, люди в ролях инженеров, художников, воспитателей детских садов, любовников, космонавтов, учителей, спортсменов. Пока они играют роли, они занимаются описанными в учебниках (инструкциях, корпоративных стандартах, регламентах, уложениях) методами работы, реализуют ролевые предпочтения в важных характеристиках самых разных систем (как целевых для проекта создания системы, так и систем-создателей). Когда эти люди занимаются другими «методами/способами работы»/деятельностями/практиками/занятиями/ культурами/«видами труда/инженерии», они играют другие роли, проявляют другие ролевые интересы в других предметах интереса. Если вы хотите добиться ответа от какого-то чат-бота, то вы должны не просто его спрашивать, а сказать, от имени какой роли по поводу какого метода работы вы хотите услышать ответ. Если вы спросите менеджера, как дела в проекте, то услышите «мы опаздываем со сдачей текущей версии на два дня», а от инженера услышите «мы меняем насосы второго контура, они не прошли испытаний». Всегда помним: «-- Сколько будет дважды два? — А мы покупаем, или продаём?».

Если вы хотите понять «предметы интереса»/озабоченности/concerns и связанные с ними предпочтения какого-то агента (человека, организации AI-агента), вам нужно знать культурно-обусловленные (то есть прописанные в учебниках, общие для всех, а не чьи-то конкретно!) предметы интереса и интересы/предпочтения в

них для той роли, которую он играет. Этот агент будет выполнять какую-то деятельность/практику/занятие/труд/метод работы, чтобы добиться желаемого им предпочтения в предмете своего интереса. Если это финансист, то он будет стремиться избежать кассового разрыва, если это менеджер, то ему нужно получить продукт вовремя — и возможный кассовый разрыв его будет волновать ровно в той мере, что придётся договариваться с финансистом, который не хочет брать кредит, ибо на эту тему финансисту придётся договариваться с бизнесменом (например, его роль может играть один или даже несколько «основателей фирмы»/founders), который не хочет брать лишние риски.

У вас должен быть трудовой/деятельностный кругозор, вы должны быть знакомыми со множеством методов работы, узнавать в жизни множество ролей, которые идут по этим методам/протоколам/сценариям, знать их предметы интереса/важные для них характеристики систем и проектов и предпочтения/интересы, которые они будут реализовывать в проекте по поводу этих предметов интереса. В наших руководствах это помогут сделать руководства по системной инженерии, инженерии личности, системного менеджмента (инженерии организации), но этих руководств, конечно, совершенно недостаточно. При этом «производственный опыт» может не помочь, если «двадцатилетний опыт — это однолетний опыт, повторённый двадцать раз». В кругозоре всё-таки важно разнообразие ситуаций, о которых ты имеешь знания/объяснения/теорию.

Трудовой кругозор даст вам способность видеть во всех агентах прежде всего не людей-актёров с их «чертами характера» и «личными особенностями» (ну, или «корпоративными культурами» с какой-то антропоморфизацией поведения организации, как будто это человек, или уподоблением AI-агента человеку), а конкретные методы работы, которые эти агенты задействуют в своих ролях, способы ведения агентами работы. Только после того, как вы поймёте, что там делает агент, вы определите играемую им роль, можно будет обсудить, хорош ли для этой роли играющий её актёр — будь это человек, организация, AI-агент или даже простой автомат, то есть можно оценить степень квалификации агента в роли.

Буквально: сначала вы понимаете, что речь идёт о «заварке кофе»::метод, далее говорите «кофе заваривают обычно бариста::роль», далее смотрите, какова квалификация Маши или Пети как бариста (в какой степени они владеют мастерством бариста, то есть в трудных ситуациях могут справляться с завариванием кофе, не делая новичковых ошибок).

Если агенты не будут выполнять какие-то методы работы (не будут что-то делать согласно роли, «по методу»), то есть если они будут бездействовать, то вас они не должны интересовать: не тратьте на них своего внимания. «Зритель», который не играет в вашей игре, не участвует в вашем представлении, не хочет заплатить, но живо этим представлении интересуется без всякого намерения поучаствовать — это не зритель (агент во внешней проектной роли «зрителя представления»), это просто прохожий.

Системное мышление — это управление вниманием, обращение внимания на важное, игнорирование неважного. Роли, выполняющие свои методы работы в отношении целевой системы или создателей этой целевой системы — это важно, а агенты, которые не выполняют работы по методам каких-то ролей так, что они влияют на проект (то есть на системы-создатели в ходе их проектной деятельности) и его целевую систему в ходе её работы — это не важно, их игнорируем.

Добавьте сюда этические рассуждения: если ваша система или проект по её созданию будет травить какого-то агента, да так, что он полумёртвый и возразить не может, такого агента надо обязательно учесть в роли затрагиваемого системой: например, роль «отравленный нашей системой» и метод исполнения этой роли — «страдание». Не игнорируйте ролей тех агентов, кого затрагивает ваша система и создатели вашей системы! Культура/стиль работы «тихое умирание всех, кто прошёл мимо нашей трубы» не так будет вам заметна, как культура «жёсткая драка с создателями тех, кто прошёл мимо нашей трубы, чтобы они убрали подальше от нашей трубы всё ими созданное», но вы должны отследить такое поведение, определить роль исполнителя работ этой культуры «пострадавший от нашей трубы» (другое выражение для «все, кто прошёл мимо нашей трубы», учитывающее смену состояния от «прохождения мимо нашей трубы») и учесть её предпочтения.

Должности, личные качества, личные предпочтения-не-по-роли, всё это оказывается не таким важным, как игра какой-то культурно-обусловленной роли, в том числе мастерство игры этой роли, мастерство применения методов работы ролью. Вот и думать поэтому нужно сначала о роли и методах/способах/практиках/«рабочих процессах» этой роли, и только потом о должности и личных качествах агента на этой должности, играющего нужную роль.

Системное мышление — это управление вниманием, управление выделением фигуры из фона. Вы должны смотреть на громко и эмоционально что-то орущего на совещании Ивана Ивановича::актёра, и воспринимать не его личную речь, а его речь как произнесение текста-в-роли (да, это трудно, это нужно тренировать своё внимание: слышать текст роли и реагировать на этот текст, а не слышать эмоции Ивана Ивановича в его крике. Конечно, для ведения разговора надо отслеживать и эмоции самих актёров, но без содержательного рассмотрения ролевых реплик найти выход из ситуации, вызвавшей крики, не получится).

Трудового кругозора должно хватить, чтобы распознать роль орущего Ивана Ивановича, внимания должно хватить, чтобы удерживать эту роль в фокусе, а не личность или должность Ивана Ивановича. Это видение роли не должно быть блуждающим, внимание должно надёжно захватывать видение роли! И дальше отвечать нужно этой роли, а не личности или должности. Это ещё сложней: смотреть на живого человека Васю Пупкина и отвечать не ему, а Принцу Гамлету, которого этот Вася Пупкин играет — неважно, хорошо или плохо играет. Отвечать нужно роли! Системность мышления в этом!

Если на вас орёт Глафира Полуэктовна, что вы земляной червь и ничего не соображаете в сложностях административного документооборота, вам нужно отфильтровать из её речи всё незначимое (включая пассажи про вас как земляного червя) и вычленить «текст по роли». Понять, в чём проблема той роли, которую играет Глафира Полуэктовна (например, речь могла зайти о методах управления конфигурацией, которые Глафира Полуэктовна будет называть не «управление конфигурацией», а «комплектация» — это нужно чётко сообразить и захватить вниманием для отслеживания). Далее нужно сообразить, какое предпочтение роли управляющего конфигурацией (на языке Глафиры Полуэктовны это будет «менеджер комплектации») в какой важной для этой роли характеристике, затем дать содержательный ответ (например, нужно будет рассказать, почему не вырастет число конфигурационных коллизий, если реализовать ваши предложения). Всё остальное будет непродуктивно, эмоции агентов останутся, а

дело с места не сдвинется.

Если будете отвечать сами как роль всегда другим ролям по поводу их интересов/предпочтений, придерживаясь разговора о предмете интереса/важной характеристике и что может влиять на её значение, а не будете отвечать людям-исполнителям-ролей/агентам или должностям, и сами не будете вести переговоры как «безролевой актёр» или «должность без роли» (принцип «я начальник ты дурак», до предметов интереса разговор не доходит), то вы со всеми быстро договоритесь. Это ролевое мастерство, этому нужно учиться, это нужно тренировать.

Проектная роль — это не конкретный человек::конструктив или даже оргзвено::конструктив со множеством людей с их инструментами и компьютерами. Проектная роль — это типовая (культурно-обусловленная, которая обычно известна другим людям «из культуры», из их деятельностного кругозора) роль, которую играют агенты, выполняя типовые действия с типовыми инструментами, типовыми рабочими продуктами, преследуя типовые предпочтения в типовых предметах интереса.

Удерживая внимание на роли, вы будете действовать культурно, то есть использовать знания человечества, а не свои слабые и часто неверные догадки о происходящем. Догадки ваши помогут, когда речь пойдёт об очень редких методах работы, там вы включите интеллект и разберётесь. Но по большей части вы будете разбираться с хорошо известными в культуре методами/способами работы и выполняющими их ролями.

Увы, люди по большей части случайно попадают в проект как исполнители профессиональных ролей, хотя и могут быть гениальными актёрами — но просто их никто не учит лучшим методам работы. Дарья Ивановна как инженер-проектировщик вполне могла бы быть заменена Серафимом Эдуардовичем. Это не сильно поменяло бы результаты работы инженера-проектировщика::роль, если оба они хорошо владеют методами инженерного проектирования. Даже переход от Серафима Эдуардовича к John Doe мало бы повлиял: инженер-проектировщик в их исполнении делал бы примерно то же самое при сравнимом уровне квалификации в выполнении работ по выбранному (иногда не осознанно) методу проектирования, и выполняемые работы по методу отличались бы примерно как игра разных актёров одной и той же роли: в чём-то лучше, в чём-то хуже, но уж точно это не превращалось бы в отыгрывание роли продавца в универмаге или отыгрывание роли инженера-испытателя. Способ, которым работает инженер-проектировщик, не уникален, методы его работы берутся из культуры. Если Серафим Эдуардович::инженер-проектировщик не обучен как инженер-проектировщик, он не может выдумать метод проектирования из своей головы, кулибинство тут вряд ли сработает. Выполнение ролей требует мастерства, этому мастерству нужно учиться, метод/способ/стиль выполнения работ этого мастерства должен мыслиться отдельно от конкретных агентов-лиц (физических лиц, юридических лиц, «компьютерных лиц»).

Этот поворот от «объективности научного мира» к «субъективности деятельностного мира» и обращение к обязательному рассмотрению проектных ролей (в английском — stakeholders) в мире произошёл примерно в 1975-1985 годах. В СССР как раз в 1985 году начались гласность и перестройка, из системных мыслителей на эту тему в те времена активно говорили представители

системнометодологического движения (последователи Г.П.Щедровицкого[197]), в их языке использовалось очень близкое к проектной роли понятие «позиция» и они первые начали говорить о том, что привнесение деятельностной субъективности в системное движение через понятие «позиционера» означает появление нового, второго поколения системного подхода. В современной терминологии можно было бы говорить о **системном подходе 2.0** (и помним, что в последние несколько лет произошёл поворот к **системному подходу 3.0** — проектные роли, оказывается, нужно рассматривать для безмасштабных агентов, рациональность и осознанность которых могут существенно отличаться от привычных, приписываемых агентам-homo sapiens, а ещё нужно учитывать техно-эволюцию с её закономерностями).

Вот график частоты упоминания слова stakeholder в библиотеке англоязычных книг Гугла[198], и это примерно отражает распространение системного подхода в его второй версии (прежде всего в литературе по проектному управлению, ибо там первыми начали говорить про людей в системах-создателях, а затем в системной инженерии. Проектное управление как метод менеджмента тоже появилось как приложение системного мышления к менеджменту, как и системная инженерия появилась как приложение системного мышления к инженерии):

Игровая метафора

Проще всего обсуждать поведение, обусловленное выполнением метода как своего рода **игру живого действия/live-action role-playing game**[199], которую разыгрывают разные составы игроков. Об игре живого действия можно думать как о такой театральной постановке, в которой нет предписанных реплик и чёткого сюжета, хотя обозначены роли и канва сюжета — остальное будет импровизацией «отыгрывающих роли» агентов/актёров. Второе важное отличие, что в ролевых играх идёт отыгрыш ролей не «на зрителя», а «себе и другим актёрам». Простейшей ролевой игрой будет игра «дочки-матери» («давай ты как будто будешь мама, а я папа, а ты будешь у нас ребёнком», или «давай ты будешь пожарным и будешь меня

---

[197] https://ru.wikipedia.org/wiki/Щедровицкий,_Георгий_Петрович

[198] https://books.google.com/ngrams/graph?content=stakeholder&year_start=1950&year_end=2019&corpus=en-US-2019&smoothing=3

[199] https://ru.wikipedia.org/wiki/Ролевая_игра

спасать», иногда эти детские игры называют **сюжетно-ролевыми**[200]), в которую играют маленькие дети, или «в войнушку» (где «ты будешь за красных, а я за синих и будем воевать»). Ещё проще спортивные игры, типа футбола, там всё идёт по простым правилам, которые немного различаются для разных ролей (вратарь может брать мяч руками, а вот другие игроки команды — нет). Спортивные игры могут быть и «себе и другим актёрам», и «на зрителя», но в целом их отличие в наличии чётко обозначенной цели игры и методов её достижения. В ролевых играх живого действия с целями и методами их достижения свободы много больше, и они больше походят на рабочие и личные проекты. Вот как выглядят игроки в классической игре прямого действия:

В театральной пьесе они выглядят примерно так же красочно, только их реплики уже предписаны, поэтому выполнение ролями работ по какому-то методу в проектах чуть дальше от театральной игры и чуть ближе к игре живого действия или сюжетно-ролевой игре.

Важнейший тут момент — **отыгрывание роли/role-playing**[201] как изменение своего поведения так, чтобы оно было как-то похожим на методы работы выбранной роли. Неважно, этот переход на «игру по роли» полностью осознан игроком/актёром именно в ролевой игре или театральной пьесе, или проходит мимо сознания и вне внимания агента, выполняющего какую-то работу в жизни — эта работа выполняется каким-то методом, следовательно агент играет роль в этом методе. Если это покупка булочки в кафе, то осознаешь ты это, или не осознаёшь — это ты выполняешь роль «покупатель» в методе обмена вещами, причём добровольного обмена, причём это «купля-продажа» с ролями «покупатель» и «продавец» (и там легко можно насчитать ещё множество других ролей: например, покупать булочку можно ребёнку, а платить не со своего счёта — и тогда роли плательщика, бенефициара, покупателя могут разойтись, каждый будет выполнять какую-то часть купли-продажи, и это мы ещё не обсуждали дробление ролей продавца. Неважно, называют ли себя как-то агенты (и даже люди ли эти агенты), важно что они делают — и по этим действиям (метод/способ) узнаем их роль.

Если ты ведёшь себя не случайно, а следуешь какому-то методу, то ты начал

---

[200] https://www.akusherstvo.ru/guide/syuzhetno-rolevye-igry-dlya-detej-pochemu-oni-vazhny/
[201] https://en.wikipedia.org/wiki/Role-playing

отыгрывать роль в какой-то ролевой игре, следуя какому-то сюжету. Смело можно считать, что это роль из какой-то игры (той же «купли-продажи» или «изучение курса в вузе» в роли «студент»), знаешь ты об этом, или нет, называешь ты это как-то, или нет. Жизнь тем самым в системном мышлении просто выполнение действий самыми разными методами/способами в самых разных ролях, вся жизнь — «игра в ролевых играх по их ролям», причём ролей очень много и ролевых игр тоже много.

Пришёл в пиццерию и заказываешь пиццу — играешь роль посетителя пиццерии. И уж если начал её играть, то играй именно её. Так, если ты попросишь на раздаче вместо пиццы продать тебе женские босоножки 37 размера, ты немедленно увидишь всю эту вроде как незаметную ролевую структуру, по которой люди выполняют работы каким-то методом/способом, удерживая себя в какой-то роли для этого метода. На твой запрос босоножек в пиццерии её сотрудники выйдут из своих привычных рабочих ролей и начнут разбираться с твоей ролью — каким методом какого результата ты пытаешься достичь (какова твоя функция как системы по отношению к твоему окружению, в котором они находятся), какая твоя роль, какие твои интересы? Если ты на работе в проекте сделаешь что-то подобное, сыграв «не по роли», то есть демонстрируя глубокий отход от предписываемого ролью метода работы, тоже немедленно получишь разбирательство с уточнением ролей и используемых этими ролями методов. Если ты неожиданно поменял вариант своего метода работы (скажем, после чтения учебника по операционному менеджменту перестал брать по десять-двадцать работ/tasks сразу, но стал параллельно выполнять не более двух работ сразу), то результат будет такой же: попытка сотрудников разобраться, что происходит, какие методы работы ты выполняешь в какой роли, почему эти методы (каких результатов ты хочешь добиться), почему в этой роли используешь этот вариант/вид метода.

Если выход за пределы «работы как ролевой игры» очевиден, то есть вы демонстрируете поведение роли в проекте не из числа ожидаемых проектных/организационных ролей, то результаты будут очень разные в зависимости от отыгрываемых коллегами ролей и того, какие роли они считают допустимыми отыгрывать в ходе проектной работы. Попробуйте попросить коллегу принести вам пиццу, или почесать спинку. Результаты просьбы точно покажут, что работы делаются в проекте не случайные, не все просьбы возможны: отношение будет примерно как в игре «дочки-матери», где папа вдруг решил выйти из своей роли и сыграть роль внучки. Лидерство — это тот метод игры, которым игрока в роли папы будут возвращать к выполнению взятой на себя роли. На работе лидерство делает то же самое: если сотруднику поручено выполнять какую-то роль в деятельности, то есть проводить работы с определёнными объектами по какому-то методу, а не выполнять произвольные действия в произвольные моменты времени с произвольными объектами, то лидерство будет стараться возвращать его в эту роль.

В рабочей обстановке «синих воротничков»[202] всё может быть менее красочно в плане одежды, но принцип разделения коллективной деятельности по ролям (разделения труда) тот же.

[202] https://en.wikipedia.org/wiki/Blue-collar_worker

В офисной обстановке с ролями и проектами всё то же самое — они радикально отличаются, но одеты все более-менее одинаково[203], так что вам придётся догадаться, кто::агент есть кто::роль иногда по репликам преследующих ролевые интересы агентов, а иногда и по их действиям (ибо можно бить по морде, говоря в этот момент избиваемому вежливые слова про погоду за окном — не всегда слова соответствуют делам, методы работы — это всё-таки про дела, изменения состояния физического мира в конечном итоге, описания тут крайне важны, но вторичны, они могут разительно не соответствовать реальности, разговоры выглядят все одинаково, а вот обсуждаемые в разговорах ситуации разительно различаются, в том числе отличаются от того, что обсуждается в разговорах)[204]:

Мало того, что действия агентов могут расходиться с репликами, ещё и каждый присутствующий в проекте агент (часть из которых будет отсутствовать, но они оказываются доступны через компьютерные сети, это тоже надо будет заметить) может отыгрывать и несколько ролей, и вы можете не успевать отследить, какую роль он отыгрывает в каждый момент времени: при конфликте интересов реплики могут быть от одной роли (скажем, козла-охранителя огорода), а действия — от другой (скажем, козла-едока капусты). Форма одежды как раз позволяет легче разобраться с ролью в важных ситуациях. Надел судейскую мантию — всё, роль судьи, пока мантию не снял. Надел пиджак с галстуком, засунул наушник в ухо —

[203] https://en.wikipedia.org/wiki/White-collar_worker

[204] Иллюстрация znamkamarada

всё, ты охранник, охраняешь тех, кто ходит в майках и джинсах, но только на время, пока ты в форме.

Несмотря на огромную разницу в интерпретации ролей игроками и мастерами ролевых игр живого действия, или даже в упрощённом варианте (реплики известны и выучены наизусть, ситуации чётко повторяемы) интерпретации ролей актёрами и их режиссёрами в разных театрах, и даже в одном театре в разные дни, всё-таки есть огромный смысл обсуждать сами сюжеты игр или сами пьесы и задействованные в них роли (**методологическую действительность,** methodology realm, действительность деятельности — классы, общее, типовое), а не только их отдельные исполнения/отыгрывания/перформансы (**операционную действительность** конкретного проекта, endeavour realm, действительность отдельных исполнений работ с конкретными датами и конкретными исполнителями).

Метафора ролевой игры задаёт отыгрыш по роли по какому-то методу и его знаниям/сценарию/алгоритму, задаваемым сюжетом этой игры, хотя и с возможными вариациями (в отличие от «безвариантных» отыгрышей театральных ролей). Сюжет/знания/теория метода описывает разные типовые ситуации, в которых агент-игрок будет действовать в роли, проводя **работы** (выполнение задач/tasks с конкретными ресурсами::конструктив в конкретный момент времени), но работы по алгоритму/сценарию/теорию, находясь в своей роли, достигая каких-то постоянно меняющихся и согласовываемых игровых (а в жизни — проектных) интересов. Важно, что методы работы и интересы разных ролей отличаются (и в ролевых играх, и в театре, и в жизни рабочей, хоббийной и личной), поэтому и методы работы оказываются разными: короли удерживают свои земли и пытаются завоевать чужие, сапожники тачают сапоги, колдуны колдуют, воины воюют, инженеры пытаются изменить мир к лучшему, менеджеры организовывают агентов на изменение мира к лучшему.

Игра/метод/практика/деятельность/функция может играться/выполняться/задействоваться много раз, метод работы повторяется много раз с разными конструктивными объектами, играющими роль функциональных объектов, задействуемых методом. Хотя каждый отыгрыш игры, или в более простом, менее импровизационном варианте — каждое исполнение пьесы в театре, в чём-то уникальны, мышление может экономиться за счёт «выноса за скобки» всего того, что повторяемо. Метод/способ подразумевает задействование мастерства, аналогичного софтверной программе, следующей алгоритму/знаниям/дисциплине/теории метода и поддержанного инструментами — тем самым метод приложим к самым разным сочетаниям входных данных и/или материалов, а на выходе выдаёт разные результаты для этих данных и/или материалов. Хороший метод работает с самыми разными объектами, если они подходящих для него типов — точно так же, как компьютерная программа работает с самыми разными входными данными, если они нужных для неё типов.

Культурная обусловленность ролей и методов, использование терминологии из культуры — важны, они дают стабильность методологическим (то есть о методах работы каких-то ролей) рассуждениям. Колдун всегда колдует, инженер-электронщик всегда разрабатывает, чинит, испытывает электронику — в какой бы это ни было ролевой или даже деловой игре, театральной пьесе, рабочем проекте. Это не зависит от того, исполняет ли эти роли мужчина, женщина, пожилой или молодой человек, большая организация, робот, каждое исполнение производится

одним и тем же актёром/агентом/игроком, или разными. Хотя, конечно, ролевое мастерство (в театре его называют актёрским мастерством: безопасный для психики вход и, главное, выход из роли, а также удержание внимания на игре в роли, отсутствие случайных выходов из роли, умение импровизировать без выхода из роли) и талант тут влияют сильно, но предполагается, что актёры/игроки/работники/сотрудники вполне владеют мастерством игры по роли — и колдун таки знает, как колдовать, а электронщик — как разрабатывать электронику. **Разница рабочих, хоббийных, личных проектов и театра с ролевыми играми только в том, что в жизни нужно честно и полностью отыгрывать свою роль, даже когда на тебя никто не смотрит, а в играх и пьесах — только изображать/показывать этот отыгрыш зрителям, «как будто».**

Знание принципов освобождает от знания фактов (тут можно указать на книгу «Программистский камень»[205] — в ней людей делят на «картостроителей» и «паковщиков» ровно на этом основании: строят ли они карту «принципов», или запоминают каждый отдельный встреченный маршрут, т.е. знают много фактов и их «двадцатилетний опыт работы — это однолетний опыт, повторённый двадцать раз»). Если вы понимаете игровой сюжет (в жизни — читали учебник по методу работы, регламент, инструкцию, протокол/процедуру, стандарт) и тем самым знаете, что делают разные роли в этом сюжете — вы можете абстрагироваться от того, кто из агентов и в какой конкретной ситуации (экземпляр ситуации) эти роли играет, вы сможете хотя бы немного предсказать происходящее. Если вы поняли, что идёт купля-продажа, то вы знаете, что есть роли продавца и покупателя, один из предметов интереса — цена, и стороны будут торговаться по поводу этого предмета интереса, причём покупатель будет стараться цену опустить, а продавец — поднять. Если покупатель цену задирает, а продавец — опускает, то вы должны догадаться, что идёт не купля-продажа, а выполняется какой-то другой метод с другими ролями (скажем, попытка обхода налогов по дарению, «я этот трёхэтажный дом не получил в подарок, а честно купил за один рубль, никакого дарения, никаких налогов» — купля-продажа тут ненастоящая, прикрывает другие роли в другой деятельности/«работе по другому методу»).

Программка в театре содержит важнейшую информацию: «действующие лица и исполнители»:

[205] http://progstone.narod.ru/reciprocality/r0/index.html

Главное управление культуры
Исполкома Моссовета

МОСКОВСКИЙ ДРАМАТИЧЕСКИЙ ТЕАТР
НА ТАГАНКЕ

ШЕКСПИР

ГАМЛЕТ

Перевод Бориса Пастернака

В спектакль включено стихотворение Б. Пастернака «Гамлет»

ДЕЙСТВУЮЩИЕ ЛИЦА и ИСПОЛНИТЕЛИ:

Клавдий, король Датский — В. Смехов, А. Пороховщиков
Гертруда, королева Датская, мать Гамлета — заслуженная артистка РСФСР А. Демидова, И. Афанасьева
Гамлет, сын прежнего и племянник нынешнего короля — В. Высоцкий
Полоний, главный королевский советник — Л. Штейнрайх
Офелия, дочь Полония — Н. Сайко
Лаэрт, сын Полония — В. Иванов
Горацио, друг Гамлета — Л. Филатов, Ю. Котов
Розенкранц, Гильденстерн, университетские друзья Гамлета — И. Дыховичный, И. Петров; А. Вилькин
Озрик — С. Холмогоров
Марцелл — В. Семенов, В. Спесивцев
Бернардо — В. Королев, В. Семенов
Рейнальдо, слуга Полония — А. Филиппенко
1-й актер — Б. Хмельницкий, В. Соболев
Мальчик-актер — Д. Межевич
Актриса — В. Радунская, Т. Иваненко
Актеры — А. Филиппенко, В. Спесивцев
Музыканты и придворные — В. Шаповалов, Ф. Антипов, Т. Додина, Д. Межевич, В. Матюхин, Д. Щербаков, С. Подколзин, А. Филиппенко, Л. Савченко, В. Соболев, А. Граббе, В. Королев, В. Семенов, В. Спесивцев, И. Афанасьева, И. Фролова, О. Школьников, Н. Чуб
Могильщики — Ф. Антипов, И. Бортник, В. Шаповалов, С. Л. Фарада, Р. Джабраилов
Призрак отца Гамлета — А. Пороховщиков, В. Смехов

Постановка Юрия ЛЮБИМОВА
Художник — Давид БОРОВСКИЙ
Композитор — Юрий БУЦКО
Ассистент режиссера — Ефим Кучер
Ассистент художника — Семен Бейдерман
Постановщик пантомимы — Валентин Манохин
Помощник режиссера — Е. Дроздова
Руководство худ. пост. частью — Ю. Хроменков, Б. Зазерский

Действующие лица/роли — это вдумчивый Принц Гамлет и безумная Офелия. У них есть своё назначение в пьесе, это ролевые/функциональные объекты.

Исполнители/агенты/оргзвенья — это весёлый актёр-стажёр Вася Пупкин в утренних спектаклях и мрачный народный артист Василий Петрович Черезколеноногузадерищенский в вечерних спектаклях, оба играющие Принца Гамлета, плюс педантичная Елена Ефимовна во всех спектаклях, и она не болеет и не замещается. Исполнители — конструктивные/материальные объекты, это системы-создатели, играющие роли в пьесе-проекте. Мышление тут то же самое, что про микроскоп, камень или молоток, которые играют роли или гвоздезабивального устройства, или пресс-папье. Конструктивные объекты играют роли функциональных объектов. Люди играют роли, роли — это функциональные объекты. Действующие лица — это функциональные объекты.

В ролевых играх всё то же самое, что в театре, но больше свободы. Такая же примерно «программка» называется **сетка ролей**[206], в ней важные замечания вроде «пока не будут заполнены прописанные роли, заявки на роли, придуманные самими игроками, не принимаются. Однако предложения игровых завязок и антуражных отыгрышей профессий и хобби приветствуются и принимаются по согласованию с мастерами». Если в ролевой игре, в которой есть гномы и пожарные, нет заполнения ролей гномов и пожарных — она провалится, но развитие сюжета может принять и какие-то дополнительные роли. Тут можно пообсуждать, надо ли это — принцип элегантности/lean наоборот, предписывает обходиться минимальным числом ролей, но какие-то другие соображения могут заставить думать и по-другому, например, если надо как-то отстроиться от конкурентов,

[206] https://vk.com/@-217236218-setka-rolei, раньше говорили и «раскладка ролей» https://hobbitsi.livejournal.com/3711.html

можно добавить к результатам проекта что-то такое, что могут добавить только дополнительные роли. Скажем, проект по созданию системы с AI требует хотя бы одного игрока/актёра/сотрудника, который умеет создавать системы с AI, но если все вокруг создают AI, может быть потребуется и какой-то другой игрок/сотрудник, который умеет отыгрывать роль дизайнера, создающего удобные интерфейсы к системам с AI, или умеет делать что-нибудь ещё, чего не умеют делать исполнители ролей в других проектах, и что может пригодиться для успеха проекта. Но бесполезно затевать проект по созданию системы с AI, если у вас есть хороший специалист, сыграющий роль разработчика интерфейсов, но никого в команде, кто отлично сыграет роль разработчика AI, нет.

Вопрос не в агентах::конструктивах, которые исполнители и по большому счёту смогут выучить работу по **методу/процедуре/«рабочему процессу»/протоколу/практике,** описанными **учебником/«учебным курсом»/регламентом/инструкцией/стандартом/руководством**, и получат мастерство, поддержанное **шаблоном/образцом/пустографкой/template чеклиста** прохождения состояния объектов. Регламенты и шаблоны чеклистов описывают метод работы, а вот конкретный экземпляр чеклиста будет отражать прохождение состояний конкретных объектов в ходе самой работы агентов по методу, это часть инструментария, усиливающего мастерство — внешняя память, которую использует в своей работе мастерство. Помним, что метод::поведение его мастерства с инструментарием обсуждается как:

- знания/теория/дисциплина/объяснения/алгоритм, описывающий метод,
- мастерство в методе — вычислитель, реализующий это знание/алгоритм в теле агента (человека, робота, организации) «аппаратно», «работающая в мозге или компьютере программа»
- инструментарий аппаратно усиливает мастерство, никто не работает «голым мозгом, голыми руками», мастерство безинструментального достижения результатов лучше приберечь для цирка, а не для работы.

Реализованные в компьютере или даже на бумаге чеклисты дают внешнюю память для мастерства метода, усиливают мастерство.

Главный вопрос в «действующих лицах» — в ролях, функциональных объектах, которые выполняют какие-то функции, изменения в окружении. Если не знаешь, что надо делать, то не можешь подобрать исполнителей (кастинг в театре — это как раз подбор аффордансов для ролей: агентов, которые смогут без особых проблем, без особого грима, без особого обучения сыграть требуемую роль).

В проектах создания и развития систем все роли заняты какими-то методами работы, а актёры (люди, компьютеры, коллективы людей и компьютеров с AI, а также с приданными им станками и материалами) важны лишь в той мере, в какой они могут сыграть нужные для проекта роли, то есть смогут работать по оговорённому методу, будут иметь мастерство и инструментарий, предусмотренные знаниями/дисциплиной метода. Причём надо обязательно проверить наличие мастерства и инструментария у агентов в проекте для выполнения культурно-обусловленных нормативных ролей.

Скажем, если в инженерном проекте не будет архитектора, то проект заведомо будет неуспешен. Дальше мы берём какого-нибудь «Серафима Владимировича, главного конструктора»::«агент в должности», который не знает о современном методе создания системной архитектуры и поэтому иногда принимает хорошие

архитектурные решения «по наитию», а иногда не принимает: просто не догадывается, что эти решения нужно обязательно принять и обязать разработчиков (с неминумемыми спорами, о которых сказано прямо в учебниках по методам архитектурной работы, а иногда и в инженерных стандартах, иногда и в регламентах) этим решениям следовать в порядке архитектурного надзора/governance. Может быть, Серафим Владимирович бы принял эти решения, если бы догадался этим заняться, а полномочий главного конструктора бы хватило для governance. Но он не знает, что должен играть роль архитектора, а также не знает современного содержания метода создания системной архитектуры «железной» или даже программной системы. А ведь в этом методе даже рассказывается, в какой форме документировать архитектурные решения (не диаграммами, а ADR, architectural decision records, подробнее смотри в руководстве по системной инженерии).

Ролевые/функциональные и выбранные аффордансами из числа доступных объектов физического мира конструктивные/материальные объекты, которые занимают в пространстве-времени одно и то же место — в 4D экстенсионализме это один и тот же объект, неважно, сознательный или несознательный, вменяемый или невменяемый, живой или неживой. Так что на момент исполнения роли в спектакле «Принц Гамлет»::роль, или в игре Принц::роль, или в рабочем проекте Архитектор::роль — это один и тот же объект, что и «Вася Пупкин::агент». Пахнуть Принц Гамлет будет запахом Васи Пупкина. Но мы их всё-таки не должны путать: Принц Гамлет существует только во время спектакля, когда Вася Пупкин играет его роль. Как и инженер-электронщик существует только во время работы, когда Ваня Щупкин играет его роль. А Вася Пупкин и Ваня Щупкин существуют независимо от игры своих ролей, и нас мало волнуют Вася Пупкин и Ваня Щупкин, когда они чихают или звонят по телефону подруге — это не Принц Гамлет или инженер-электронщик чихает и звонит по телефону, это Вася Пупкин и Ваня Щупкин в других ролях!

У Принца Гамлета (у роли!) свои предметы интересов и интересы в них — его волнует вопрос «Быть или не быть?». И у него есть предпочтение для состояния «быть» перед состоянием «не быть» по отношению к себе как системе. Он действует, исходя из этого предпочтения, а поскольку это театр, то действует, очень точно следуя предписаниям роли (пьеса тут — отраслевой стандарт) и указаниям режиссёра (стандарт предприятия, конкретного театра в конкретном варианте постановки каким-то режиссёром, то есть организатором). Действует/работает ресурс «Вася Пупкин»::актёр::агент, или «Василий Петрович Черезколеноногузадерищенский»::актёр::агент, это операционное менеджерское рассмотрение. Но вот функциональное рассмотрение — с точки зрения спектакля действует «Принц Гамлет»::роль. Агенты действуют/работают в проекте, играя роли, и только играя роли, то есть действуя/работая по каким-то методам! Если Вася Пупкин заболел, то это не Принц Гамлет заболел, но это Вася Пупкин стал играть роль пациента в проекте собственного лечения. И лечат пациента (Васю Пупкина, играющего роль пациента), а не Принца Гамлета. «Пациент» — это имя культурно-обусловленной роли в методе «лечение/therapy»!

В системном мышлении, когда говорим о ролях, то всегда имеем в виду действующее лицо, то есть лицо из пьесы/игры/деятельности — Принца Гамлета, роль, функциональный объект. Поведение какого-то агента (например, взрослого хорошо образованного человека) в системном мышлении — это игра его ролей (их

может быть несколько!) в игре/пьесе/методе, выполнение функции::поведение роли::«функциональный объект», поведение агента-в-роли.

Системный мыслитель всегда смотрит на метод работы и выполняющую этот метод работы роль агента, и пытается понять, в какой ситуации выполняется именно этот метод работы, что там за «ролевая игра», какие в ней нужны роли для каких необходимых работ, чего они там все пытаются добиться своей игровой импровизацией в рамках какого игрового сюжета, и уже только потом оценивает исполнителя работы — человека-актёра (или даже AI-агента, или целую организацию, необязательно актёр тут — именно человек), насколько он может выполнять требующийся для успеха метод работы.

Мы можем потребовать заменить актёра-исполнителя (безвестного Пупкина на талантливого народного артиста Черезколеноногузадерищенского), но обычно не можем потребовать заменить действующее лицо (вместо Принца Гамлета вдруг потребовать вставить в пьесу Бармалея и Бэтмена, или в проекте по созданию космического корабля потребовать вставить вместо инженера-электронщика кондитера или даже пилота-космонавта). Это огромное достижение цивилизации: роли культурно-обусловлены, они знакомы многим людям, про «игру по роли» (инженера-программиста, преподавателя физкультуры, врача-окулиста и т.д.) написаны учебники, а исполнители/агенты привносят в них собственное своеобразие (не всегда «личное», не все роли играются отдельными людьми-личностями!) — и это сливается в одно «исполнение роли».

Мышление об агентах (людях и AI): прежде всего они исполнители ролей

Конечно, в реальной жизни мы непосредственно видим в первую очередь исполнителей::конструктив — конкретных актёров::агентов, отыгрывающих роли, а не сами «роли». Но обсуждаем по ходу пьесы или ролевой игры мы исключительно роли, если только речь не идёт о квалификации агента, то есть качестве следования методу конкретного агента, качестве удержания себя на выполнении роли. Тут важна степень мастерства агента в выполнении метода, но также важна сила интеллекта, в том числе входящая в интеллект собранность для удержания себя на выполнении метода, которым должна заниматься в проекте играемая агентом роль.

Кто говорит фразу «быть или не быть»? Принц Гамлет, или Вася Пупкин? На момент отыгрывания/исполнения роли оба они — один и тот же объект, только называются по-разному и мы обращаем в зависимости от этого внимание на разные свойства этого объекта. Когда речь идёт о «действующем лице»/роли, то мы обращаем внимание на текст и сюжет (пьесы, ролевой игры, проекта создания какой-то системы, тут ведь не многое меняется при переходе от точной игры «по Шекспиру» к импровизационному отыгрыванию сюжета с Принцем в ролевой игре, или импровизационному отыгрыванию орг-архитектора в проектах организационного развития), а когда речь идёт об «исполнителе, отыгрывающем роль», то на качество исполнения и доступность исполнителя::ресурс в момент спектакля/«прохождения игры»/«течения рабочего проекта», то есть момент отыгрыша роли (поэтому все эти разные понятия оказываются по факту одинаковым указанием на время отыгрыша роли, «эксплуатацию/функционирование агента в качестве исполнителя роли»).

Мы всегда можем указать Васе Пупкину, что он плохо понял роль в игре или проекте (в спектакле жёстче — «плохо выучил роль»), или играет чужую роль и всяко по-

другому дать понять, что «ты не прав, Вася», если нам известна пьеса, или ролевая игра, или сюжет рабочего проекта, который он играет. Если сюжет и роль Васи неизвестны, то мы не можем понять — прав, или не прав Вася в своих действиях, «по роли» ли это, или «так нельзя!». Помочь человеку удерживать свою роль и отыгрывать её качественно — это лидерство. Если лидер не знает роль и её метод работы, то никакие уговоры «работать хорошо» Васю не исправят: он будет очень хорошо что-то делать, только совершенно не нужное для проекта, а лидер (который не знает, что нужно) этого не отследит. Если корабль не знает, куда плывёт, любой ветер не будет ему попутным. Если лидер не знает, на какую роль надо уговаривать Васю хорошо поработать, да и сам Вася не знает этой роли и не компетентен в не знаемом им методе работы, никакие уговоры не будут результативны. Сначала лидеру надо знать роль уговариваемого им актёра, понимать, какой там должен быть метод работы этой роли, лидерство по отношению к агенту-актёру (исполнителю роли) станет возможным только после этого.

Системное мышление тем самым требует деятельностного кругозора (минимальный кругозор даётся в руководствах по системной инженерии, инженерии личности, системному менеджменту) для распознавания методов работы и ролей исполнителей работ по этим методам в обстоятельствах рабочих проектов. В частности, обсуждение того, от какой роли лидер берёт понимание метода работы роли, на исполнение которой он «убалтывает» какого-то агента, какие вообще там должны быть роли — ровно в этих руководствах, а само лидерство как метод работы лидера обсуждается в руководстве по системному менеджменту, лидер как подроль преподавателя/наставника обсуждается и в руководстве по инженерии личности.

А.Тюков в 1997 году выделял такие сферы деятельности: политика, религия, философия, искусство, наука, образование, здравоохранение, физкультура и спорт, технология, проектирование, коммерция, финансы, право, армия, материальное производство и «пока не оформившееся просвещение»[207]. С тех пор прошло несколько десятков лет, появляются огромные вопросы к самой этой классификации. Хорошо видно, что она крайне антропоцентрична и отражает какие-то идеи о государственном устройстве. Раньше это было нормально, но уже появляется понимание, что основания рассмотрения методов в их дроблении на более мелкие и выделение альтернативных вариантов могут быть совершенно другими. Но выбор конкретной классификации (какие рода и виды методов бывают) — это «спор о терминах», это не важный вопрос, предлагать другие классификации не так уж продуктивно. Хотя мы и сами предлагаем какую-то «типологию»: классифицируем методы по типу эволюционной сложности систем, с которыми они имеют дело: инертное вещество, киберфизические системы, живые существа, агенты в узком смысле, организации (коллективы агентов), сообщества, общества. Тогда можно свести самые разные методы создания и развития систем к методам инженерии, найти много общего в методах самых разных проектов.

Например, можно подметить, что методы работы (деятельность/практика/труд) визионера в инженерных проектах, культуртрегера в проектах обучения, бизнесмена в проектах организационного строительства очень похожи, это всё методы, которыми пользуется теоретическая роль «шумпетеровский предприниматель» из учебников австрийской школы экономики (не путать с

[207] http://psyhoinfo.ru/programma-sozdaniya-obshchestvennoy-professionalnoy-sfery-prosveshcheniya

бытовым значением слова «предприниматель», это именно «шумпетеровский предприниматель»)! И тогда становится проще выявлять подобную роль и адаптировать её метод работы для самых разных проектов. Именно этот подход унификации всех методов как методов инженерии (создания и развития успешных систем) мы и применяем в наших руководствах.

Более важно для кругозора то, что «кругозор вчера» совершенно не означает «кругозора сегодня». Появилось и каждый день появляется множество самых разных новых методов достижения каких-то нужных нам состояний мира, а также исчезло и каждый день исчезает огромное число устаревших методов/практик/культур/деятельностей/видов труда. Скажем, много лет уже не видно кассиров в магазинах, которые считают на счётах с костяшками, а ведь это было когда-то обязательным! Да и деятельность кассиров исчезает как таковая, а кошельком стремительно становится смартфон. В инженерии всё трудней найти инженеров-конструкторов, которые работают с 2D-проектированием, это чёткий маркер заповедников старого производственного уклада, на такие производства надо ходить как в музеи. Современные предприятия давно работают по-другому. **Надо иметь не просто кругозор как «большой опыт работы со старыми методами работы, знание старого набора ролей», но кругозор современного состояния методов работы и действующих по ним ролей.**

Нет кругозора в методах/культуре/деятельностях/инженерии/видах труда — нет системного мышления, ибо оно трансдисциплинарно, оно призвано объединять ролевые мышления для этих отдельных сфер деятельности. Так что нужно уметь распознавать эти роли в проектах.

Мы рекомендуем иметь кругозорное знание хотя бы о следующих методах работы в современном многоуровневом мире, сортированных по системным/эволюционным уровням (в биологии их называют «уровнями организации»):

- Пока разговоры об «инженерии для всей вселенной в целом» носят сугубо теоретический характер (живых богов, творящих миры, пока не встречено и следов их не найдено), но приведём его для полноты картины.
- Цивилизационная инженерия во вселенной, включая нашу и инопланетные цивилизации в целом. Деятельность по изучению и даже прикидки по изменениям космических масштабов уже какие-то есть (например, активно обсуждается парадокс Ферми[208], есть проекты по обнаружению «техносигнатур»[209], и даже обсуждение технических решений по энергетике цивилизаций с использованием максимального количества энергии от энергии близлежащих звёзд — это вполне инженерия космических масштабов, строительство сфер Дайсона[210]).
- Развитие «антропосферы», а по-бытовому — развитие человечества, включая и самих людей, и их инструменты, и ещё всю среду обитания. Мы::люди пока тут одни на маленьком глобусе, и неплохо бы изменить всё человечество к лучшему (в том числе защитить от астероидных опасностей, вымирания от техногенных катастроф, барьеров роста из-за ограничений в экологии и т.д.). Увы, ничего хорошего по линии осознанного изменения мира тут продемонстрировать нельзя, несмотря на то, что вроде бы ООН именно

[208] https://en.wikipedia.org/wiki/Fermi_paradox
[209] https://en.wikipedia.org/wiki/Technosignature
[210] https://en.wikipedia.org/wiki/Dyson_sphere

этим занимается, но техно-эволюция, да и биологическая эволюция тоже идут — Просвещение, которое началось ещё во времена Вольтера и Руссо продолжается до сих пор[211]. И самое тут существенное изменение — это появление AI-агентов, статус участия которых в эволюции непонятен, но из-за которых уже требуется антропоцентричные/парохиальные описания мира заменять менее антропоцентричными. В нашем руководстве мы в 2023-2024 году аккуратно меняли «людей» на агентов, указывая на то, что в развитии антропосферы необязательно участвуют только люди.

- Общественная деятельность, которой занимаются общественные деятели, их часто называют «политиками». Речь идёт о методах «социальной» (то есть с сообществами и обществами) работы на базе нормативных[212] (то есть инженерных) экономики, социологии, политологии или вообще без опоры на эти теории, методом «проб и ошибок» (в руководстве по системной инженерии показано, что инженерия и научна, и ненаучна). Предметом метода работы (деятельности/практики/труда/инженерии) на этом организационном/эволюционном уровне является общество, в котором использующий метод деятель/инженер/практик не знает контрагентов в лицо, ибо их слишком много, не знает интересов этих контрагентов, ибо они могут играть самые разные роли. И всё же общество может быть достаточно организованным, то есть иметь какое-то управление в лице разделённых или слитых законодательной, судебной и исполнительной властей, а уж эти организации через подчиняющиеся им силовые/вооружённые организации могут принуждать всех остальных исполнять приказы нескольких агентов (не обязательно людей!), то есть могут задавать нормы общественной организации и требовать их исполнения. Могут быть самые разные цели инженерии общества: выставлять границы на «свою» (тут вопрос про собственность: бывает ли «народная собственность», или это собственность властного органа/организации, или частных лиц, необязательно «человеческих лиц» — но причём тут тогда «общество»?) территорию (не обязательно на Земле) и противодействовать своему исчезновению при агрессивных изменениях окружения (скажем, войны с другими обществами) или изменениях в своей организации (скажем, захват власти религиозным или санитарным фанатиком или диктатором и последующая сознательная или даже неосознаваемая их работа по уничтожению этого общества в силу каких-то иррациональных убеждений). Ещё и путаются сами объекты инженерии этого уровня: «общество», «народ», «нация», «страна», «государство», «граждане». Проекты строительства коммунизма, создания империй, просто попыток как-то обеспечить процветание на уровне хотя бы «у нас мало голодающих, в других местах их больше», попытки устроить разные варианты «демократии» с разным разделением властей — это как раз инженерные проекты этого системного уровня.
- Культурное строительство, инженерия сообществ, этим занимаются деятели культуры (в том числе инженерной культуры, политической культуры,

[211] https://ailev.livejournal.com/1574477.html

[212] https://www.encyclopedia.com/social-sciences-and-law/sociology-and-social-reform/sociology-general-terms-and-concepts/normative, https://plato.stanford.edu/entries/logic-normative/, https://plato.stanford.edu/entries/economic-justice/, https://plato.stanford.edu/entries/rationality-normative-utility/, и так далее.

менеджерской культуры, необязательно деятели искусств или хоббийной культуры). Сообщество как предмет инженерии/строительства/развития — определения subculture, counterculture, community, community of practice, learned society, school of thought из википедии (посмотрите их): все говорят об одном и том же, подчёркивая разные стороны разным языком, используя разные термины. И мы тоже можем говорить разным языком, подчёркивая разные стороны, но об одном и том же объекте «сообщество». У нас по-русски это будет субкультура, контркультура, сообщество практики, сообщество наученных (деятелей, «наученных/образованных в каком-то мастерстве», а не «учёных/исследователей»!), «незримый колледж». Но можно говорить и о племени (очень модно!), и о «муниципальном образовании» типа «деревня, где все друг друга знают», землячестве бывших (или даже нынешних) членов одного общества в каком-то другом обществе. Ориентируемся на число Данбара[213] (150, столько может удержать человеческая биологическая память), но с учётом современных компьютерных средств человеческая память быстро растёт, и рассуждения про это число становятся не очень надёжными. В Мордокниге принят лимит «друзей» в 5 тысяч человек как число знакомых, с которыми человек может хоть как-то быть знакомым и поддерживать отношения при помощи не только собственной памяти, но и компьютерной памяти. Но это может быть и много бóльшее число. Например, число системных инженеров киберфизических систем, которые вступили в Международный совет по системной инженерии на 2022 год — более 19000, это означает, что эти люди считают себя членами сообщества системных инженеров, тем самым примерно известны их ролевые интересы, но это не означает, что речь идёт о традиционной организации, которая создаёт какую-то целевую систему. Сотрудники штаб-квартиры, однако, занимаются построением сообщества как целевой системы. Думать о сообществах проще не как об организациях членов сообществ, но как об организациях, в которых члены сообществ играют свои роли — если у них всех одинаковая деятельность, то они в одинаковых примерно ролях участвуют в самых разных проектах, что и задаёт их общность. И тогда какой-нибудь «проектный офис» как «внутренняя консалтинговая организация» становится как раз организацией по развитию проектных менеджеров, развивает сообщество проектных менеджеров. И тогда число Данбара уже не имеет такой важности. Отдельно надо заметить, что продвижение товара, услуги, какого-нибудь предпочтительного метода работы с его знаниями и инструментарием — это создание сообщества. Продвижение (маркетинг, реклама, продажи) как создание клиентуры (сообщество клиентов) как раз тут. С одной стороны, для создания клиентуры надо выполнить инженерию личности, выдав знания по продвигаемым методам отдельным лицам, но с другой стороны — чтобы обратиться к конкретному лицу с методами инженерии личности, сначала надо это лицо найти, то есть обратиться к заранее неизвестному кругу лиц, то есть построить сообщество, и вот эта вторая часть с «выполнить инженерию личности для заранее неизвестных личностей — где их взять, заранее неизвестных?» тут и обсуждается, на уровне сообщества.

[213] https://ru.wikipedia.org/wiki/Число_Данбара

- Оргстроительство/менеджмент/«организационная инженерия»/«enterprise engineering»/«инженерия предприятий», которой занимаются менеджеры/организаторы/«организационные инженеры»/«enterprise engineers». Предмет их методов работы — организация, понимаемая как группа агентов (чаще всего людей, но мы помним о безмасштабности и AI-агентах) с понятными полномочиями по распоряжению работами и прочими ресурсами друг друга. Инженерия организации на стадии эксплуатации/работы/operations организации — это операционный менеджмент. Есть отдельное руководство по системному менеджменту, которое по факту руководство по инженерии предприятия, это прикладное кругозорное руководство (уровень метаУ-модели), конкретизация/заземление на предметную область менеджмента фундаментального безмасштабного эволюционного руководства по системной инженерии. Если создаваемая и развиваемая система — предприятие, штаб-квартира ассоциации, некоммерческая организация, фонд и т.д. (коллектива агентов), то надо работать по менее абстрактной версии руководства по системной инженерии, а именно — по руководству по системному менеджменту.
- Образование, психотерапия, коучинг, которыми занимаются педагоги, психотерапевты, коучи с таким объектом, как деятельная личность. Это всё «инженерия личности» (хотя так и не говорят, но мы советуем думать именно так!). Включать ли сюда разные варианты обучения «разумных роботов», то есть личностей не на базе homo sapiens, то есть личности как рациональные осознанные на сравнимом с homo sapiens уровне агенты-организмы (можно назвать их условно xGI, разные виды general intelligence)? Почему бы и нет. Поэтому тут можно рассматривать и проекты AGI (artificial general intelligence), и проекты создания киборгов (обычно людей и организаций, но тут больше людей) в части экзокортекса и экзотела/инструментов (внутриличных уровня частей личности, личных и коллективных — удерживаем, что разговор вроде как сейчас про один уровень, но его легко перенести и на другие. Скажем, «образование деятельной организации», а не только «образование деятельной личности», но исторически уровень одной личности выделяется довольно чётко). Грубо говоря, «добавка памяти и возможности вычислений», ибо добавка искусственного зрения, печени и т.д. — это уже более низкий системный уровень, «существа», им занимаются другие методы работы. В программе рабочего развития мастерской инженеров-менеджеров есть руководство по инженерии личности, где даётся кругозор по методам инженерии личности как прикладному варианту (конкретизации для системы «личность») методов системной инженерии.
- Методы/деятельность/практика/инженерия создания и разведения/«серийного изготовления» живых существ, «организмоводство». Ими занимаются деятели/практики/инженеры живой природы, такие как агрономы (растениеводство), фермеры (животноводство и ветеринария), лесничие (лесоводство), а также врачи (медицина). Люди — это тоже живые существа! Если нас волнует работоспособность человеческого организма/тела (разрабатывается руководство по валеологии[214]), а не только

[214] https://ru.wikipedia.org/wiki/Валеология

личность (инженерия личности), то это как раз методы инженерии уровня живого существа, живой природы. Эти методы создания и развития живых организмов/существ включают и «системную биологию» с «системными биоинженерами» как разработку и производство полностью искусственной жизни, и «генную инженерию» как модификацию генома уже имеющихся на планете живых существ.

- Классическая системная инженерия киберфизических систем, которой занимаются классические «железные» и «киберфизические» системные инженеры. Она работает с такими системами, в которых ещё нет интеллекта, но в которых уже есть универсальный компьютер-контроллер. Ракеты и самолёты, атомные и солнечные электростанции, умные (то есть обвешанные датчиками и кое-какими исполнительными устройствами) дома и прочие традиционные объекты для инженеров-технарей. Программная инженерия тоже тут. В какой-то мере руководство по системной инженерии выстроено не только как руководство по одному из методов интеллект-стека, но и содержит множество примеров методов прикладной классической инженерии, как «железной», так и программной. Это руководство как раз даёт кругозор по созданию и развитию систем этого уровня эволюционной сложности.
- Вещество (молекулы, простые физические детали): тут от органического синтеза инженеров-химиков до инженерии мыльниц и электролампочек инженерами-конструкторами, то есть инженерия простых/невменяемых (без сложной кибер/компьютерной/вычислительной) составляющей систем без сложного управления с контроллерами-компьютерами в их составе.

Конечно, в жизни вы будете встречать проекты, в которых нужно будет вместе рассматривать создание и развитие систем на самых разных системных уровнях — и это будут в том числе разные уровни эволюционной сложности этих систем. Противоречия между системами разных системных уровней проявляются как раз тут: клетки печени хотят, чтобы их было как можно больше, а организм считает это раком. Какие-то граждане не хотят поддерживать какое-то «государство» как организатора/инженера общества на территории их проживания, их посчитают «государственными изменниками» и накажут. Как решать такие противоречия? Изобретать новые и новые инженерные приёмы.

Но вам надо как-то понимать, какого сорта инженерные методы потребуются в вашем рабочем проекте. Самый простой способ это сделать — это выписать конкретные типы затрагиваемых систем из вашего проекта, присвоив им типы видов систем из нашего руководства. Например, если вы создаёте таблетку из лактобактерий как БАД (биоактивную добавку), вам потребуется как-то изменить (в том числе, например, уговорить принимать эту таблетку — изменить состояние личностей людей как частей клиентуры) системы на следующих уровнях стека:

- Потребнадзор::сообщество и общество
- Клиентура::сообщество (продвижение)
- Клиент::агент-индивид (личность и организм)
- Желудок и кишечник::существо
- Таблетка::вещество
- Лактобактерия::существо

Во-первых, часть этих объектов — это роли: они будут выполняться агентами, часть ожет быть «агентом на должности» (Потребнадзор, выполняет множество ролей — как минимум «медика/инженера безопасности тела» и «представителя общественности/общества потенциальных пациентов»), клиент может оказаться не ролью, а тоже связкой ролей (сторона контракта, плательщик, бенефициар, в данном случае ещё и потенциальный пациент, но для БАД это же не пациенты, а «приверженцы ЗОЖ», и т.д.).

Тут есть даже два перескока через уровень: лактобактерии производятся как живые системы, а затем высушенные становятся частью вполне неживой таблетки. Потребнадзор тут половинчат (обычное свойство госорганов): отражает мнение сообщества медиков с одной стороны и вроде как мнение общества в целом, со всеми их конфликтами. Его тоже нужно изменить (уговорить этого агента считать таблетку безопасной, то есть найти ответственных людей и AI, из которых он состоит и убедить их). Лактобактерии — это биоактивные добавки (грубо — сушёный кефир), они не требуют медицинского лицензирования. Если бы это было лекарство, то ситуация была бы более сложной.

Цель создания списка систем — достижение понимания создателями системы «таблетка из лактобактерий», что работа по каким-то инженерным методам будет на всех этих уровнях. Поэтому в проекте придётся обратиться к самым разным агентам, имеющим мастерство и инструментарий для методов работы с существами, методов работы с веществами, методов работы с личностями, методов работы с сообществами и даже обществом в целом (если выяснится, что это как-то затрагивает общественные интересы и нужно вести какие-то общественные дискуссии, менять законодательство и тем самым заниматься политикой). А ещё нужно будет создать организацию проекта, которая всем этим займётся, так что есть и роль организатора::менеджер, и возможно там будет не один человек её исполнителем (отдельно операционный менеджер, отдельно орг-проектировщик, отдельно лидер). Подробней про методы организовывания и их разложения на составляющие методы, роль организатора и её разложение на более мелкие составляющие роли говорится в руководстве по системному менеджменту.

Методологический (разнообразие методов работ, он же — трудовой/деятельностный/культурный/стилевой/практический/инженерный) кругозор важен, чтобы не запутаться в этом разнообразии культурно-обусловленных методов работы и исполняющих эти методы ролей: нужно примерно представлять, что скрывается за самыми разными действиями с самыми разными объектами, которые мы наблюдаем, за самыми разными словами, которые мы слышим и читаем от самых разных агентов в самых разных проектах. И после того, как будет понятно, какого типа упоминаемые агентами объекты (какой системный уровень, какой метод работы используется, какая проектная/деятельностная/функциональная/инженерная/трудовая/культурная роль выполняет используемый метод работы, как её называют такую роль, какой учебник или инженерный стандарт для метода работы говорит о SoTA в этом методе, что там в проекте целевая система, какие роли у систем создания этой целевой системы и какие интересы этих ролей, что там с графом создателей, что там функциональные объекты, а что там конструктивы, каким образом выбирались аффордансы для этих конструктивов, и т.д.), станет много проще: **обо всём этом «зоопарке» самых разных объектов и деятельностей вы будете думать примерно одинаковым образом и не упускать важного.**

В системном мышлении мы всегда должны думать:

- **От границы системы в её окружение во время использования системы сначала, а о составе системы внутри границы системы потом.**
- **О методах работы, ведущих к изменению состояния каких-то объектов в мире сначала, затем о роли исполнителя метода, и только потом о поиске аффорданса (агента, имеющего мастерство исполнения работ по методу и инструментарий для этих работ) как конструктива для роли.**

Это ещё один способ суперкратко обратить внимание на мантру системного мышления. Эта мантра задаёт постоянно действующий мыслительный очаг, гарантирующий проведение подобного рассуждения, гарантирующий постоянное мыслительное усилие — поначалу сознательное и трудное, а потом переходящее в мыслительный автоматизм.

Так, надо научиться видеть в агентах (людях, но также и AI, также и организациях) сначала действующих лиц, и только потом исполнителей ролей: должны сначала видеть Принцев Гамлетов, а только потом играющих их Василиев Пупкиных. Для этого нужно иметь методологический/трудовой кругозор, знать о существовании основных методов/способов работы (деятельностей/культур/стилей/стратегий/практик/«видов труда/инженерии/»), встречающихся в самых разных видах проектов и типовых для этих методов работы ролей.

Если мы не знаем про то, какая разыгрывается перед нами деятельность/пьеса/практика/культура/метод/способ/стратегия работы, то мы не можем оценить действия этих людей (они будут казаться личной прихотью), спланировать свои действия (они будут неважны для людей-в-ролях, ибо лежат вне их ролевых предметов интереса, буквально неинтересны, не будут иметь отношения к их предпочтениям), не можем сыграть свои роли в играемой ими пьесе (ибо будем бескультурными, «не выучили роль», книжек не читали, не знаем, как обычно это делается) — и без этого нас просто не поймут! И мы ни в коем случае не должны путать Принцев Гамлетов и Василиев Пупкиных! Мы не должны обращаться к Принцу Гамлету как к Офелии — исполнитель роли просто не будет знать, что делать и вместо работы по проекту будет «бытовой разговор» с выяснением отношений и (правильными) упрёками в непрофессионализме.

Это очень непростой навык (видеть в людях, AI и организациях роли и отмечать, по какому методу они работают, какая степень мастерства этих людей во владении методом, какой вариант метода из каких учебников или стандартов отыгрывает агент в роли), но он необходим. Это первое, с чего начинается системное мышление. Это означает, что **вы как системные мыслители должны в любой момент времени ответить — какая роль человека::актёра проявляется сейчас перед вами, в чём его ролевой интерес/предпочтение в какой важной характеристике, и отвечать этой роли (а не исполняющему роль человеку!) соответственно роли, заняв при этом какую-то свою роль в играемой им пьесе/ролевой игре — тоже став действующим лицом, а не оставаясь просто «безролевым недеятельным исполнителем».** Если не

будете играть свою роль в его пьесе/игре, то вы будете просто невидимы для него, ваши высказывания не будут иметь для него значения, вы будете для него «статистом из толпы»/NPC (non-player character, персонаж-робот из компьютерных игр) общаться с вами человеку-в-роли будет неинтересно, со статистами не разговаривают.

Особо сложно тут играть свою, а не чужую трудовую/деятельностную роль (мы уже говорили, что это отдельное мастерство удержания роли, **ролевое мастерство**, по аналогии с актёрским мастерством). Сидеть тихо и «всё понимать про другие роли» (обычно это работа аналитика: ничего не делать, только понимать, это маленькая часть работы инженера, которому надо делать) — это ведь только половина дела. **После понимания нужно ещё и обязательно действовать** (стать синтетиком, а не аналитиком — стать деятелем, меняющим мир, стать практиком/инженером/трудящимся/актёром-в-роли), представляя свой ролевой предмет интереса и ролевое предпочтение в нём, а также выполняя культурно-обусловленные или инновационные/фронтирные методы создания и развития системы, удовлетворяющей предпочтениям самых разных ролей.

Конечно, у вас может быть множество ролей (особенно в маленьких проектах, где вечно «театр одного актёра» или «игра одного игрока»), но лучше бы маркировать и для себя, и для окружающих, когда вы::актёр переходите из одной роли в другую. Это объявление своей роли для окружающих нужно тренировать, усиливать свой интеллект (осваивать методы мышления интеллект-стека, в том числе методы мышления, на которые наставляет наше руководство), а также тренировать ролевое мастерство (в котором можно выделить ещё мастерство собранности как мастерство надёжного управления вниманием), расширять свой методологический/трудовой/инженерный кругозор, чтобы не попасть впросак, не заметив какого-то культурно-обусловленного метода работы и не узнав тем самым культурно-обусловленной роли её выполняющего актёра. Для получения этого кругозора важно проходить прикладные курсы и стажировки по инженерии разных видов систем (у нас это прежде всего стажировки в работе по руководствам по системной инженерии, инженерии личности, системного менеджмента).

Трудностей в таком мышлении про методы работы и выполняющие их роли — множество. Например, когда на вас орут на каком-нибудь совещании, то просто невозможно сосредоточиться — видишь только человека (не человека::актёра, не исполнителя роли) и его к тебе отношение «вне роли», а как личности к личности. Но это «невозможно сосредоточиться» лишь означает недостаточный ваш уровень мастерства собранности и тем самым близость к мыслительной ошибке: вы перестали мыслить системно, вернулись к мышлению дикаря, которого ведут эмоции — видите в людях только эмоциональных животных и не учитываете знаний цивилизации о методах работы и ролях людей как агентов. Вы теряете доступ к даже имеющемуся у вас методологическому/трудовому/инженерному кругозору, вы не можете вписать ваше собственное прикладное мастерство в коллективные ситуации задействования мастерства множества агентов, вы не можете учесть ваш ролевой предмет интереса к каким-то системам в ситуации и удовлетворить интерес из-за того, что вы не замечаете ролевого поведения людей::актёров, позволяя эмоциям выключить рациональное рассуждение.

Важно, чтобы в проекте все обсуждения проходили в терминах «действующих лиц» и методов их работы, а не исполнителей/актёров с игнорированием знаний/теорий и инструментария методов, и никакие эмоции или другие отвлечения тут не должны

мешать. Сравните два диалога:

1. Обсуждение, в котором роли явно не упоминаются, вместо «действующих лиц» говорят имена актёров/исполнителей:

— Иванов опять чертежи испортил! Он присылает их в формате .dwg и ссылается на Петрова! Сидорова не может работать!

— А что думает об этом «Красшефмонтаж»?

— Его не волнует, лишь бы «Заготбазарбаза» не возражала!

Всё ли вам понятно, если вы случайно попали на совещание? Можно ли задать какие-то уточняющие вопросы по непониманию — и какие? Если вы хорошо знаете всех действующих лиц, то можете ли вы предсказать хоть как-то их предполагаемые реакции в данной ситуации?

2. Ролевое обсуждение, когда вместо имён актёров называют имена их персонажей/ролей (ровно тот же пример, что в предыдущем пункте):

— Конструктор опять чертежи испортил! Он присылает их в формате .dwg и ссылается на расчётчика! Архив не может работать!

— А что думает об этом технолог завода-изготовителя?

— Его не волнует, лишь бы поставщик корпусов не возражал!

Стало ли понятней, о чём идёт речь? Какие уточняющие вопросы вы бы задали?

Вот ещё пример, где мы назовём исполнителя по его должности, а не по роли:

-- Старший программист: когда я вчера смотрела на график нашего проекта, то поняла, что нам может не хватить времени на тестирование, поэтому неплохо бы озаботиться контрактацией дополнительных серверов для этой работы.

Варианты продолжения беседы (какой выберете?):

- Инженерный: какие ошибки требуют дополнительных серверов для их тестирования? Можем ли мы уменьшить объём тестирования? [и оцените вероятность, что «старший программист» не подумала об этих вопросах, когда делала своё заявление]
- Менеджерский: у нас предусмотрено это бюджетом, или как всегда? У вас есть проект контракта? Кто найдёт поставщика? [и оцените вероятность, что «старший программист» и впрямь не знает ответов на эти вопросы]

Можно ли верить, что «старший программист» желает обсудить инженерные аспекты дела, она и вправду выступила как программист? Или она выступила как менеджер?

Вы бы сами какую линию разговора повели: чем бы озаботились, о чём спросили?

Как кто вы поступили бы в этом разговоре (какую бы компетенцию вытащили в качестве основной — инженерную или менеджерскую)?

Вспомните, когда последний раз в разговоре вы проводили такой ролевой анализ «кто и о чём говорит — кто и о чём должен ему отвечать»?

Отслеживаете ли вы смену предметов разговора (смену предмета интересов/важных характеристик в обсуждении)? Не означает ли это для вас, что агент-собеседник поменял роли, или просто вытащил для обсуждения другой предмет интереса той же самой роли? Как долго (в часах разговора! собранность!)

можете удерживать внимание на диалоге так, чтобы в любой момент понимать «кто (какая роль) говорит — кто (какая роль) отвечает — предмет обсуждения/интереса, какие предпочтения в этом предмете у ролей (не у исполнителей ролей/актёров)»?

Обсуждение в терминах «действующих лиц» (понимание проектных ролей как ролевых/функциональных «деятелей», а не конкретных агентов-исполнителей, например, конкретных людей или конкретных людей-на-должности) крайне важно для коммуникации: такое обсуждение направляет мысль и позволяет понимать, какие «пьесы» сейчас обсуждаются — какие реплики могли бы следовать, а не только какие реплики следуют прямо сейчас. Если какой-то Принц Гамлет вдруг начинает давать реплики Офелии — то можно дальше обсуждать: спасает ли он пьесу ввиду неявки Офелии, или просто портит дело как некомпетентный актёр-исполнитель и нужно немедленно его заменить в роли Гамлета, уволить из спектакля и назначить на роль другого актёра, более грамотного/квалифицированного в своей роли или хотя бы более внимательного. То же самое в ролевой игре «Принц Гамлет», эта нелепость будет видна и при импровизационном характере отыгрыша ролей. Ровно то же будет и в рабочих проектах, **если финансист начнёт давать реплики разработчика чат-бота и наоборот, разработчик чат-бота вдруг начнёт давать реплики финансиста: надо сразу начинать разбираться, это спасение проекта, или порча дела.**

Когда идёт проектная деятельность, то людей (и искусственные интеллекты тоже!) лучше называть по их проектным ролям, а не по фамилиям или названиям организаций. Самый тяжёлый случай, это когда люди в проекте знают важность какого-нибудь Василия Петровича (он точно исполняет какую-то проектную роль! Он существенно влияет на проект! Он начальник!), но не могут назвать его проектную роль, не понимают каким методом он работает в этой роли, насколько он мастер в этом методе, насколько осознанно его мастерство, не понимают его ролевых предметов интереса, предпочтений, задаваемых методом работы.

Действия Василия Петровича поэтому для участников в проекта «невычислимы», так что они не знают, что от него ожидать, как реагировать. Чаще всего это бывает с начальниками, людьми-в-должности, про полномочия по распоряжению трудом и ресурсами которых можно долго и много рассказывать, но вот их мышление, их методы работы и задействующие эти методы работы роли и мастерство в этих ролях — вот этого сказать про них нельзя, знание производственной культуры тут бессильно, и нужно специально исследовать вопрос про профессиональные/трудовые роли этих людей в проекте, какие у них ролевые предметы интереса и предпочтения в них, что они умеют делать, кроме как что-то продавливать своими часто немаленькими полномочиями. Выглядят «начальники» часто как менеджеры (инженеры организации), но обычно оказываются очень узкими специалистами в какой-то специфической «железной» или «программной» инженерии, или коммуникациях как умении «договариваться и улаживать конфликты», и даже довольно редко в менеджменте (организационном и операционном), когда заходит речь именно об их ролевой работе. Всё остальное время они просто штампуют своими полномочиями чужую ролевую работу (проверяя формально, что у остальных участников проекта нет возражений — но это и голосование в чате может сделать, не нужно чату для этого быть начальником!), воспроизводя медленно работающий каскад «предложений-утверждений» в организации с организационной культурой «аналитиков и

начальников».

Вопросы. Проектные роли

В Центре медицинских разработок города Нью-Васюки есть идея разработать новый прибор для диагностики рака. Какие проектные роли было бы правильно учитывать в проекте?
--- Базовая больница N5 при Центре медицинских разработок города ### Нет, это конкретная больница, конструктивный объект, а не функциональный/ролевой — это не проектная роль. Можно ещё отметить, что в крупной организации обычно обнаруживается много разных проектных ролей, и «больница» тут говорить очень неспецифично. Больница тут оргзвено, конкретная организация, конструктив, а не роль.
--- Нью-Васюки ### Нет, город тут не проектная роль, это конструктивный объект.
--- Больница ### Это может оказаться функциональным/ролевым объектом, но слишком много проектных ролей может быть обнаружено в рамках больницы. Так что, скорее всего, речь тут идёт о больнице как конструктивном объекте-предприятии, без уточнения её назначения (а как только уточним функцию, то есть «что больница делает в проекте, зачем она нужна», то сразу всплывёт много проектных ролей в больнице).
+++ Врач-онколог ### Да, это проектная роль для онкологических методов лечения — интересы врача-онколога (профессиональная роль, которую будут играть потом какие-то конкретные врачи) нужно учитывать при разработке прибора для диагностики рака.
+++ Больной с подозрением на рак ### Да, это проектная роль для методов лечения онкологической болезни — интересы больного с подозрением на рак (роль в ситуации эксплуатации прибора для диагностики рака, которую будут играть потом какие-то конкретные больные) нужно учитывать при разработке прибора для диагностики рака.
+++ Пациент отделения онкологии ### Да, это проектная роль для методов лечения онкологической болезни — «пациент отделения онкологии» (без указания конкретной больницы и имени конкретного пациента!) указывает просто на интересы больного с подозрением на рак (роль в ситуации эксплуатации прибора для диагностики рака, которую будут играть потом какие-то конкретные люди-больные). Интересы больного нужно учитывать при разработке прибора для диагностики рака.
--- Директор Центра медицинских разработок города Нью-Васюки ### Нет, «директор» тут — указание на ответственность и полномочия по распоряжению ресурсами Центра медицинских разработок (он единолично представляет оргзвено «Центр медицинских разработок города Нью-Васюки»), он может выполнять самые разные проектные роли, так что у директора можно обнаружить самые разные предметы интересов самых разных проектных ролей и самые разные предпочтения в этих важных характеристиках.
--- Программист Центра медицинских разработок города Нью-Васюки ### Нет, это указание на должностную позицию, он может исполнять проектные роли и инженера-программиста проекта, и инженера-

разработчика аппаратуры, и операционного менеджера проекта, или вообще не участвовать в данном проекте.
+++ Программист### Да, это функциональная/проектная роль в методах программной инженерии, имеющая профессиональное отношение к предмету проекта.
--- Главный онколог больницы N 5 Валентина Ивановна ### Нет. Во-первых, указана должность (главный онколог больницы может и не быть уже лечащим врачом, а быть только администратором), и, во-вторых - указан конкретный исполнитель роли, а не сама роль.

Конструктор изделия Иван Петрович поругался с начальником цеха по поводу использования при производстве 3D-модели изделия. Какие проектные роли упомянуты в этой фразе?

--- Конструктор изделия Иван Петрович ### Нет, Иван Петрович — это исполнитель роли конструктора изделия, работающего по методам конструирования, а не сама роль.
+++ Конструктор изделия ### Да, это проектная роль, основной предмет интереса — конструкция изделия, метод «конструирование».
--- Иван Петрович ### Нет, это исполнитель роли конструктора.
--- Начальник цеха ### Нет, это должность ответственного и полномочного агента как распорядителя ресурсов цеха. Он может оказаться в разных проектных ролях («старшина гильдии актёров» может играть принца Гамлета, а может вообще не быть занят в спектакле).
--- В этой фразе проектные роли не упоминаются ### Нет, помянута проектная роль «конструктор изделия», проявляющая конструкторский предмет интереса, связанный с 3D-моделью изделия.

Иван Иванович — главный конструктор в отделе робототехники. Он заявил, что не допустит, чтобы в проекте использовались иностранные комплектующие. Интерес какой проектной роли он выражает в этой ситуации?
--- Ивана Ивановича ### Нет, это исполнитель роли, а не роль.
--- Главного конструктора ### Нет, «главный конструктор» тут не проектная роль (слово «главный» указывает прежде всего на должность, организационное место, возможности по распоряжению ресурсами и раздаче поручений подчинённым). Начальники могут играть самые разные роли, в том числе и не входящие в их традиционно понимаемые должностные описания.
--- Главного конструктора отдела робототехники ### Нет, «главный конструктор» тут не проектная роль (слово «главный» указывает прежде всего на должность, организационное место, возможности по распоряжению ресурсами и раздаче поручений подчинённым). Начальники могут играть самые разные роли, в том числе и не входящие в их традиционно понимаемые должностные описания.
+++ Протекциониста ### Да, «протекционист» - это роль, метод работы этой роли — протекционизм (метод назван, как обычно: по названию

теоретической экономической теории/объяснению этого метода, может быть вариант «протекционирование» с привычным отглагольным существительным для функции/культуры протекциониста::роль), его ролевой предмет интереса «иностранные комплектующие», ролевое предпочтение «отсутствие иностранных комплектующих», даже если это противоречит специфическим конструкторским предпочтениям в конструкторских важных характеристиках (например, предпочтения конструктора — повысить надёжность, уменьшить сроки конструирования и т.д.).

Конструкторы Вася и Петя подрались из-за старшего программиста Оли (они оба в неё влюблены). Как правильно назвать их проектные роли в этой ситуации?
+++ Дерущиеся ### Да, в этой ситуации у них есть интересы по драке — победить, не быть побитыми, метод достижения интереса — рукопашный бой.
+++ Влюблённые ### Да, в этой ситуации у них есть интересы в любви — выиграть объект любви, быть любимыми.
--- Конструкторы ### Нет, они играют роли конструкторов в другом проекте, в другой ситуации. В текущей ситуации их нельзя назвать конструкторами, в ситуации не участвуют конструкторские предметы интереса, метод конструкторской работы (включая конструкторские знания и конструкторский инструментарий) никак не связаны с дракой.

Начальник цеха заваривает чай. Работающий в цеху слесарь Иван Петрович подробно разъясняет ему, почему чай нужно заваривать в мягкой воде. Как правильно назвать проектную роль Ивана Петровича в этой ситуации?
--- Слесарь ### Нет, ситуация заваривания чая не включает роли слесаря, предметы интереса слесаря и метод слесарной работы в ситуации не затрагиваются.
--- Подчинённый ### Нет, ситуация заваривания чая не подразумевает каких-то отношений распоряжения ресурсами, в том числе человеческими. Обсуждение оргзвеньев тут ни при чём.
--- Работник цеха ### Нет, включение в какое-то оргзвено никак не подразумевает проектной/функциональной/деятельностной/орг роли в ситуации заварки чая.
--- Иван Петрович ### Исполнители ролей — не проектные роли, Вася Пупкин не Принц Гамлет.
+++ Повар ### Да, в ситуации заварки чая роль повара, его поварские предметы интереса и знания и инструментарий кулинарных методов определяющи.

Чашку поставили на блюдце и поместили в сушку, где их обдувает вентилятор. Какой из этих объектов в этой ситуации проектная роль?
--- Чашка ### Нет, чашка не агент/актёр в узком смысле (не автономна по части принятия решений), не может вести деятельности, у неё не может

быть ролевых/деятельностных предметов интереса, она не выполняет какой-то метод работы (но может выполнять функцию после изготовления).
--- Блюдце ### Нет, блюдце не агент/актёр в узком смысле (не автономно по части принятия решений), не может вести деятельности, у него не может быть ролевых/деятельностных предметов интереса, оно не выполняет какой-то метод работы (но может выполнять функцию после изготовления).
--- Сушка ### Нет, сушка не агент/актёр в узком смысле (не автономна по части принятия решений), не может вести деятельности, у неё не может быть ролевых/деятельностных предметов интереса, она не выполняет какой-то метод работы (но может выполнять функцию после изготовления).
--- Вентилятор ### Нет, вентилятор не агент/актёр в узком смысле (не автономно по части принятия решений), не может вести деятельности, у него не может быть ролевых/деятельностных предметов интереса, он не выполняет какой-то метод (но может выполнять функцию после изготовления).

+++ Проектные роли тут не упомянуты ### Да, проектных ролей как ролей интеллектуальных агентов (по-старому — стейхколдерских ролей или даже стейкхолдеров) тут нет, ибо в описываемой ситуации нет достаточно интеллектуальных агентов — некому тут заниматься трудом (но после изготовления могут заниматься выполнением функции), и уж точно некому иметь предметы интересов и предпочтения в них. Нет исполнителей проектных ролей, есть только неживые и неразумные предметы, которые не проводят ролевых рассуждений, не могут играть проектных ролей. При этом учитываем, что вряд ли речь идёт о достаточно разумных сушке и вентиляторе, чтобы считать их полноценными агентами, исполняющими какие-то проектные роли и с мнением которых надо будет считаться в проекте. Но в ближайшем будущем мы не знаем, изменится ли ситуация, конструктивы сушки и вентилятора могут оказаться удивительно разумными и с неплохим инженерным кругозором, а «сушка» и «вентилятор» могут быть ролями более-менее универсальных агентов-создателей (universal constructors из constructor theory).

На совещании программист прошивки гаджета Вася заявил, что цветовая гамма разрабатываемого гаджета ему не нравится. Как правильно назвать его проектную роль в этой ситуации?
--- Программист ### Нет, обсуждаемый интерес не программистский, программисты не обсуждают обычно цветовую гамму изделия как предмет их интереса, методы работы программной инженерии этого не касаются.
--- Вася, на совещании исполняющий роль программиста ### Нет, исполнитель какой-то роли (актёр) — это не сама роль. Тем более что тут не идёт речь о роли программиста, цвет гаджета явно не входит в набор предметов интереса программиста прошивки, методы программной инженерии этим не занимаются.

--- Программист, на совещании исполняющий подроль дизайнера ### Нет, программист ещё может выполнять подроль тестировщика и роль кодера, но давайте смотреть на методы работы программиста: методы дизайна не входят в методы работы, используемые ролью программиста. «Вася в должности программиста, на совещании исполняющий роль дизайнера и выполняющий работы методами промышленного дизайна» — вот так бы могло подойти.
+++ Дизайнер ### Да, это стейкхолдерская роль, соответствующая трудовому/деятельностному отношению к системе. Именно дизайнеры обычно обсуждают цветовую гамму, это их предмет интереса, у них есть предпочтения в этой характеристике системы и знание методов промышленного дизайна.
+++ Критик ### Да, это роль в дискуссии, проектная/деятельностная/функциональная роль критика выполняет в рамках метода критики в части приёмов выбора аргументов, обоснования каких-то предложений. Трансдисциплинарный метод мышления, который лежит в основе критики — логика (обсуждается в руководстве по интеллект-стеку). В описанной ситуации Вася исполняет в том числе и роль критика, критикуя (выполняя работы по одному из известных ему методов критики) текущие предпочтения в важной характеристике и обосновывая предложения дизайнера. Один исполнитель роли/актёр вполне может исполнять несколько разных ролей.

Научный руководитель проекта к.ф.-м.н. Василий Петрович составляет диаграмму Ганта в программе MS Project. Какая у него проектная роль в этой ситуации?
--- Учёный ### Нет, никакой деятельности/метода работы учёного тут нет: нет выдвижения гипотез, нет экспериментов по проверке этих гипотез, научных интересов тут нет.
--- Научный руководитель проекта ### Нет, «руководитель» — это явное указание на должность, ответственность и полномочия, это не про проектные роли, а про распоряжение какими-то ресурсами оргзвена (например, направляемыми на исследования ресурсами исследователей в проекте).
--- Василий Петрович ### Это исполнитель проектной роли, но не сама проектная роль.
+++ Менеджер ### Да, составление диаграммы Ганта — типичный метод работы для менеджера, помогает удовлетворению интереса в минимальных сроках выполнения работ и требуемых для этих работ ресурсов.
+++ Пользователь программного обеспечения ### Да, в проекте использования программы MS Project речь идёт о роли пользователя, задействующего метод «использование инструмента для прикладных задач» (предмет интереса — функционал кнопок программы).

На совещании инженерной группы сотрудник отдела кадров Люся много говорила про недопустимость опозданий инженеров при возвращении с

обеденного перерыва. Как правильно назвать её проектную роль в этой ситуации?
--- Инженер ### Нет, опозданиями занимаются не инженеры (у инженеров нет такого предмета интересов, это не предмет инженерных методов).
+++ Менеджер ### Да, опозданиями занимаются менеджеры (опоздания — это предмет менеджерского интереса, организация работы людей — метод работы агентов в роли менеджера).
--- Люся ### Нет, Люся — это исполнитель какой-то оргроли, а не сама роль. Вася Пупкин — это не Принц Гамлет.
--- Участник инженерного совещания ### Нет, участников совещания было много, но в ситуации обсуждается не опоздание как важная характеристика самого совещания, а опоздание как важная характеристика выхода на работу (опоздание тут явно на совещание, опаздывающий тут — роль участника совещания как выполняющего метод достижения каких-то результатов путём проведения совещаний, она не имеет отношения к культуре/стилю/способу/методу выхода сотрудниками на работу).
--- Сотрудник отдела кадров ### Нет, сотрудник отдела кадров — это вообще не роль, а организационное место — должность.

Агент определил, что при моделировании системы будут использованы уравнения Навье-Стокса. Как правильно назвать его в этой ситуации?
--- Менеджер ### Нет, проектная роль менеджера уравнениями для вязких ньютоновских жидкостей не интересуется.
+++ Инженер ### Да, уравнениями для моделирования вязких ньютоновских жидкостей обычно интересуются инженеры-технари.
--- Инвестор ### Нет, инвесторы обычно не интересуются уравнениями моделирования вязких ньютоновских жидкостей.
--- Спонсор проекта ### Нет, это менеджерская роль, а менеджеры обычно уравнениями моделирования системы не интересуются (если это не операционные менеджеры, но они не пользуются обычно уравнениями Навье-Стокса).
--- Директор предприятия ### Нет, директор предприятия вообще не проектная/стейкхолдерская роль, это должность.

Петя ставил хореографию для танцевального выхода, а затем танцевал перед зрителями и споткнулся. Как правильно назвать его проектную роль в этой ситуации спотыкания?

--- Постановщик танцевального перформанса (хореограф) ### Нет, при спотыкании не играет роль его деятельность как постановщика (время создания перформанса), речь идёт о ситуации исполнения, а не постановки (подготовки к исполнению) танцевального перформанса.
+++ Танцор ### Да, для танцора спотыкание во время исполнения важно, его предпочтение — не спотыкаться, Петя именно в этой проектной роли.

Петя попробовал себя в роли хореографа (постановщика танцевального перформанса) и споткнулся во время показа танцевальных движений танцорам. Как правильно назвать его проектную роль в этой ситуации?
+++ Постановщик танцевального перформанса (хореограф) ### Да, это деятельность по постановке (подготовке танцоров к исполнению какой-то хореографии как предзаданного набора танцевальных движений), предмет интереса тут — показ движений танцорам, предпочтение — точность показа (т.е. можно повторить показ, если споткнулся. А танцору уже ничего нельзя повторить — зрители будут смотреть его танцевание однократно), Петя тут в роли постановщика.
--- Танцор ### Нет, не идёт речи об исполнении танца, речь идёт только о показе в ходе постановки. То, какой Петя танцор, тут не имеет значения, у него нет в данном случае обычных интересов танцора (интерес к собственному качеству исполнения танца у него не такой большой, как интерес к качеству исполнения танца другим танцорам). Тренер по футболу необязательно хороший футболист, хореограф необязательно хороший танцор.

6. Интересы

Предметы интереса и интересы

В системном мышлении рассуждение обычно идёт согласно системной мантре, имеющей множество различных формулировок, но всё сводится к тому, что сначала рассматриваем работающую целевую систему в окружении (концепция использования), потом-— из чего её сделать (концепция системы) и какими методами можно сделать эту систему (методы инженерии), затем-— кто мог бы эти методы выполнять (создатели-инженеры), затем-— кто мог бы создать этих создателей (создатели-организаторы инженеров), и так далее по графу создателей. В любом случае, в какой-то момент при определении важных объектов внимания в проекте его участники::роли отталкиваются от культурно-обусловленного понимания, какие объекты наиболее вероятно будут менять своё состояние в ходе проекта-— исходя из знаний о том, как устроен типовой инженерный метод создания для тех или иных видов систем. Собственно, все наши руководства как раз наставляют в том, что в проект системный мыслитель входит отнюдь не с нулевым знанием о происходящем в проекте (который он ещё толком не видел) и что могло бы там вообще происходить.

Например, предполагается, что на каждом шаге роста функциональности развиваемой системы (непрерывное развитие подразумевает не разовое создание системы, а создание и развитие системы с постепенным наращиванием её функциональности) будет осознана возможность создания очередного инкремента системы, он будет описан в концепции использования, откорректирована будет концепция системы, затем концепция системы будет детализована в проекте/design с достаточной для изготовления инкремента точностью, затем изготовлены части нового инкремента, система собрана и отлажена «в физике», проверена, введена в эксплуатацию с новым инкрементом, если будут найдены проблемы, выведена из эксплуатации с переходом на предыдущую версию, отэксплуатирована (работа оператора, мониторинг), и так далее-— для систем разной природы будет что-то подобное, хотя терминология может существенно отличаться (например,

«изготовление мастерства» термин использован не будет, а будет говориться об «обучении мастерству»).

В любом случае, в проекте ожидается некоторый набор объектов, которые будут менять в ходе проекта свои состояния. Эта смена состояний проводится какими-то методами работы. Например, концепция системы делается с участием разработчиков (их предмет интереса-— функциональная декомпозиция и набор конструктивов в качестве аффордансов для подсистем как функциональных объектов) и архитекторов (их интерес в том, чтобы система была построена из конструктивов с оптимальной автономией, ибо это влияет на многочисленные архитектурные характеристики-— и это часто противоречит предложениям каких-то групп разработчиков). Интерес разработчиков обычно-— быстрый ввод новой функциональности сейчас («быстрое и грязное решение»), интерес архитекторов-— быстрый ввод новой функциональности в будущем (минимизация технического/архитектурного долга). Подробней это обсуждается в руководстве по системной инженерии, но сейчас важно, что есть методы работ по разработке и методы работ по принятию архитектурных решений, которые как-то согласованно двигают по состояниям объект «концепция системы», причём это движение происходит в ходе всего проекта, учитывая непрерывное развитие системы и выпуск очередных версий системы с новыми и новыми инкрементами, реализующими новую функциональность.

Само движение по состояниям каких-то изменяющихся в ходе проекта объектов, а также способы моделирования этого движения будут рассмотрены в руководстве по методологии. Какие объекты и как именно они движутся по состояниям-— это будет рассмотрено в инженерных руководствах (по системной инженерии, инженерии личности, системному менеджменту). В этих же инженерных руководствах будут рассмотрены роли, которые выполняют инженерные (менеджерские, учебные, продвиженческие и т.д.) методы работы. Каждый из этих методов работы заставляет агента, выполняющего соответствующую роль, преследовать какие-то ролевые интересы.

В конечном итоге нас интересуют не сами роли, а их методы работы (включая методы мышления/рассуждений) и важные/ключевые характеристики каких-то систем (concerns, системные/ролевые/трудовые/деятельностные предметы интереса, «озабоченности»). Важные характеристики — это характеристики системы, которые интересны ролям в связи с их методами работы. Значения этих характеристик роли меняют своим методом/способом, реализуя свой интерес/предпочтение в значениях этих важных характеристик. Несколько смежных важных характеристик иногда собирают в область интересов, area of concern, области объектов внимания ролей в проекте.

Например, для роли архитектора в современной архитектуре явно указываются характеристики, которые должен отслеживать архитектор (явно выбираются и документируются в проекте ведущие предметы интереса роли), и даже даются указания, какие лучше использовать метрики для отслеживания соблюдения этих характеристик. Например, для отслеживания меры удовлетворения интересов в поддержке принципа «непрерывного всего» (continuous everything) разработчиками::роль и «инженерами внутренней производственной платформы»::роль архитекторам::роль предлагаются ключевые четыре метрики (Four Keys of DevOps, предложены в книге "Accelerate: The Science of Lean Software and DevOps" by Nicole Forsgren, Jez Humble, and Gene Kim):

- Частота ввода в эксплуатацию
- Длительность прохождения изменений
- Процент сбоев для изменений
- Время восстановления обслуживания

Кроме этих характеристик/concerns архитекторов интересуют (их предметы интереса) операционные (то есть времени эксплуатации/использования) характеристики: доступность, непрерывность, производительность, возможность восстановления, надёжность и безопасность, устойчивость, масштабируемость. И ещё характеристики времени создания: конфигурируемость, расширяемость, устанавливаемость, повторное использование, локализация, сопровождаемость, переносимость, поддерживаемость, возможность апгрейда.

Общие для всех систем, эти ключевые архитектурные характеристики (они же-— архитектурные интересы, выражаемые самыми разными метриками) не относятся к функциональности, поэтому неудачные названия были non-functional requirements, quality requirements. Они хорошо известны системным инженерам как –ilities/-ости, и было понятно, что достигать приемлемых значений этих характеристик можно, изменяя конструкцию системы, то есть используя разные варианты сборки системы из конструктивов/модулей и разные варианты организации взаимодействия модулей через интерфейсы.

Эти характеристики и возможные метрики для них непрерывно обсуждаются сообществом архитекторов, в каждой инженерной организации эти характеристики имеют свои варианты, в каждом проекте у архитектора внимание уделяется только небольшой части этих характеристик. Но агенты, выполняющие роль архитекторов натренированы отстаивать свои интересы в достижении этих или подобных этим характеристик. Сам список этих характеристик-— хорошее начало для обучения архитекторов тому, что должно быть предметом их пристального внимания.

Подобный набор предметов интереса и какие-то идеи по приемлемым значениям достигаемых метрик для этих предметов интереса имеется и у всех остальных ролей, при этом ввиду огромного разнообразия видов систем сам набор этих «ключевых характеристик»/«предметов интереса» огромен, а в проекте участвует огромное число ролей, выполняющих работы по самым разным методам и поэтому затрагивающие вольно или невольно чужие предметы интереса: в проекте непрерывно идёт коллективное обсуждение таких решений по поводу целевой системы и всех остальных систем в графах создателей, которые ведут к успешности проекта: если почти все интересы соблюдены, то создаваемая и развиваемая проектом система успешна, проект её создания и развития успешен.

Предметом интереса может быть всё что угодно. Так, ISO 42010:2022 даёт следующий (тоже абсолютно неполный) примерный список этих предметов интереса для архитектора, причём требует, чтобы архитектор выявил и документировал их в явном виде: пригодность архитектуры для достижения целей целевой системы, возможности и потенциал предприятия для реализации целевой системы, осуществимость реализации и эксплуатации целевой системы, потенциальные риски и воздействие целевой системы на проектные роли в течение всего времени инженерного проекта, принесение пользы проектной роли (ролям), повторное использование известных архитектур, сопротивляемость разрушению, расширяемость, адаптивность, время отклика, использование ресурсов, эффективность, работоспособность, полезность, удобство использования,

совместимость, сложность, устойчивость и развиваемость целевой системы, воздействие на окружающую среду при разработке, использовании и выведении из эксплуатации целевой системы (suitability of the architecture for achieving the objectives for the entity of interest, enterprise capability and capacity to implement the entity of interest, the feasibility of realizing and operating the entity of interest, the potential risks and impacts of the entity of interest to its stakeholders throughout its life cycle, added value to the stakeholder(s), reuse of known architectures, resilience, extensibility, adaptability, latency, resource utilization, effectiveness, operability, usefulness, usability, interoperability, complexity, sustainability and evolvability of the entity of interest, environmental impacts of the development, use, and disposal of the entity of interest).

В предыдущей версии ISO 42010:2019 вместо entity of interest было system of interest, в текущей версии слово «система» заменили на более общее. В ISO 15288:2023 при ссылках на ISO 42010:2022 слово entity заменяют на system, так что мы последуем этому же примеру. Ещё интересно, что примерный список архитектурных характеристик в ISO 42010:2022 отличается от приведённого в предыдущих абзацах, основанных на новейших обсуждениях предмета системной архитектуры во времена перехода к «непрерывному всему». Обсуждения в среде архитекторов того, что является предметом их интереса не прекращается, и у каждого архитектора предполагается своё мнение о том, какой его ролевой предмет интереса (архитектурная характеристика), и какой его ролевой интерес (каково значение архитектурной характеристики нужно получить) в каждом конкретном проекте, причём в ходе проекта приоритетные интересы архитектора могут меняться. Именно это и указывается сейчас в учебниках для архитекторов: вы не сможете удовлетворить все возможные архитектурные интересы, поэтому выберите лишь несколько главных предметов интереса (архитектурных характеристик), выпишете их явно-— и профессионально оптимизируйте метрики только для них, а остальные держите на уровне «гигиены», то есть удерживайте на минимально приемлемом уровне, чтобы не вызывать замечаний.

Предметы интересов определяют вопросы одних ролей к другим ролям: их интересует прогноз развития ситуации по поводу реализации/соблюдения их предпочтений в значениях этих важных характеристик. Это не вопросы исполнителей/агентов, это вопросы трудовых/профессиональных ролей/«действующих лиц». Это не вопросы Васи Пупкина, это вопросы его роли Принца Гамлета. Важные характеристики и предпочтения — они ролевые, нужны для выполнения работы этой ролью по методу работы этой роли.

Агент/актёр, исполняющий какую-то назначенную ему роль, может использовать средства удовлетворения своего ролевого интереса, далеко выходящие за пределы исполнения работы по методу назначенной роли. Например, актёр-1 может неожиданно для актёра-2, который играет роль с другим предпочтением, чем у роли актёра-1, позвонить своему другу, а друг его-— начальник актёра-2. Актёр-1 дальше просит друга «надавить» на этого исполнителя, чтобы было реализовано его предпочтение/интерес, а не предпочтение/интерес роли актёра-2. Итогом может быть не рациональная аргументация в пользу изменения мнения роли актёра-2, а банальный шантаж: угроза не роли, а самому актёру (начальственная угроза директора театра лишить Васю Пупкина премии, если он будет настаивать на произнесении «быть или не быть», когда Вася играет роль Гамлета-— и пусть выкручивается как хочет перед Шекспиром, режиссёром, зрителями, театральными

критиками).

Это всё чистая импровизация актёра-1, но никто не сказал, что реализовывать свои предпочтения люди будут, просто заявляя о них на совещаниях другим ролям. Нет, люди/актёры очень изобретательны, когда реализуют предпочтения своих ролей, они могут и роли для реализации этих намерений пару-тройку раз сменить, и других людей поднанять для исполнения недостающих ролей. Методы работы, предметы интереса и интересы ролей, то есть действующих лиц, а не исполнителей/актёров!

Самим исполнителям/актёрам как ролям агентов (можно и так рассматривать: актёрский труд тоже ведь труд! Входить в роль, играть пьесу, приходить вовремя на спектакли/игры, не путать слова) зарплату хочется получить, сладко поспать перед выходом на работу, развлечься да развиться для прихвата более интересных ролей. Вася Пупкин или Мирослава Эриховна ничем особо не выделяются, своих собственных предпочтений у них в текущих проектах обычно нет, но когда они начинают играть свои роли, то есть выполнять труд артиста (с отыгрыванием на зрителя роли Принца Гамлета), инженера (с разработкой CAD-моделей), менеджера (с уторговыванием плана проекта по ресурсам и срокам), маркетолога (с оценкой объема рынка) — вот тут и просыпается инициатива и изобретательность.

Это «активная деятельная позиция»/предприимчивость/инициативность/пытливость и творчество, но не какой-то узкой роли, а актёра, причём поддержанное интеллектом и дальше имеющимся у агента всем набором мастерств исполнения работ по прикладным методам.

Актёр разрабатывает стратегию (**стратегия** — это специализированный метод для получения результатов в специфической ситуации), на основании её планирует свою работу (уточняет доступность ресурсов), а потом действует, на ходу стратегируя (переопределяя стратегию) и перепланируя работы, исходя из изменившихся обстоятельств.

Почему актёр действует? Он просто считает, что вложиться в дополнительные действия отыгрывания ролей по дополнительным методам работы «здесь и сейчас» принесёт ему выгоду в будущем, «там и тогда». Он действует абсолютно инженерно, то есть в порядке стратегирования на основе своей модели мира генерирует множество альтернатив каких-то методов достижения своих целей в интересах профильной роли, оценивает выгоду этих действий, сравнивая вложения в них и результирующую воображаемую картину мира будущего (классическая роль предпринимателя по Йозефу Шумпетеру), а потом планирует с учётом наличных ресурсов действия/работы по выработанной стратегии-— и действует, непрерывно уточняя состояние дел с удовлетворением интереса своей основной роли и меняющейся ситуации с готовностью агентов, играющих другие роли, как-то поддаваться его давлению.

Системному мыслителю как деятелю/актору нужно уметь поддерживать разговор на темы предметов интереса в проекте (у многих ролей могут быть одни и те же предметы интереса, но разные интересы-— и это требует согласования интересов), давать для каждой роли ответы по предметам их интереса достаточные, чтобы каждая командная и внешняя роль могла оценить реализуемость своих предпочтений/интересов.

В проекте важно документировать описания распространённых областей интересов, чтобы облегчить достижение договорённостей. Например, если предмет интереса в

цене уникальной системы, то нужен какой-то вариант сметы, или протокола о договорной цене, или экспертной оценки по ценам аналогичных продаж на открытом рынке, или ещё какой-то вариант описания. Нужно будет выбрать то описание, которое позволит достичь соглашения. Это документированное описание цены::«ключевая/важная характеристика» будут изучать и продавец::роль, и покупатель::роль. Описание цены будет их основным коммуникационным инструментом для достижения договорённости по их совершенно разным предпочтениям в цене как важной характеристике. И этим описанием цены будут интересоваться не только продавец и покупатель, оно может потребоваться для согласования своих предпочтений большому количеству других ролей. Например, операционный менеджер может интересоваться ценой, когда будет оценивать максимально возможный набор ресурсов для выполнения работ по проекту: он должен иметь максимум ресурсов, но не больше, чем позволит согласованная между покупателем и продавцом цена.

Если вы встретили человека, у которого нет ярко выраженного интереса (интересуется всем подряд, или не интересуется вообще, не имеет ярко выраженных предпочтений, хотя вроде и заявляет какой-то предмет интереса) — значит это не исполнитель роли в проекте, а кто-то просто «любопытненький», «из статистов», «проходящий мимо зевака», случайно зашедший на сцену во время спектакля зритель, забежавший случайно в ролевую игру не-игрок. Не тратьте на него время до тех пор, пока он не проявит предметов своего интереса и предпочтений! Но после того, как вы поймёте, что он будет делать::метод с получаемой им информацией о предметах его интереса в проекте, ровно наоборот — вы должны на него тратить время, согласовывать с ним принимаемые решения, как бы ни хотелось его проигнорировать. **Создание систем — это непрерывный поиск договорённостей по поводу проекта. Успешный проект — это когда все договорились, а системный мыслитель должен сделать так, чтобы всех вокруг себя договорить!**

Если роль (Принц Гамлет, не Вася Пупкин) беседует о чём-то, что не входит в его деятельностный интерес, то вы беседуете в этот момент с актёром (возможно, играющим какую-то другую роль в данный момент, вам неизвестную), а не с действующим лицом вашей «пьесы», вашей деятельности. **Для того, чтобы надёжно различать людей в их различных ролях и определять их методы работы, предметы интереса и предпочтения, нужно иметь трудовой кругозор — в общих чертах представлять, как устроена человеческая деятельность с разделением труда в его разных изводах на самые разные методы работы, какие виды/варианты методов есть в каждом классе методов (например, в системной инженерии сразу рекомендуют обращать внимание на такие методы работы разных ролей, как визионерство, разработка, архитектурная деятельность, инженерия производственной платформы, а в менеджменте это бизнес, организовывание, орг-архитектурная деятельность, администрирование), с системами какой природы эти методы работают, какие предметы интереса этих методов работы, какие предпочтения в этих предметах.** В предыдущем разделе про методы и роли мы уже это обсуждали, но тут добавляем, что в кругозор входит и понимание предметов интереса типовых ролей. В проекте недостаточно будет распознать метод работы и выяснить культурно-обусловленную роль, выполняющую метод. Надо ещё и понимать, в чём интересы этой роли, чего она будет добиваться!

Не склеивайте важную характеристику и предпочтение

Слово «интерес» в английском будет interest, это «предмет коммерческого интереса», «что нас интересует в проекте». В системном мышлении просто исторически называют interest немного другим словом: **concern**[215], что на русском в более точном переводе звучит как «**озабоченность»: важная характеристика системы, предмет интереса к системе.** Предметы интереса — это характеристики системы, к которым приковано постоянное внимание разных ролей в проекте, это не ролевые хотелки, куда двигать предмет интереса. Предмет интереса многих ролей и интерес какой-то роли как «ролевая хотелка куда двигать предмет интереса» — разное. Предмет интереса, характеристика — не «цена повыше», а «цена», а вот интерес/предпочтение — как раз «цена повыше», куда двигать предмет интереса! Для цены может быть и другое предпочтение: цена пониже, но «предмет интереса»/«ключевая характеристика» будет тем же — цена.

По предметам своего интереса проектные роли как внешние, так и роли в команде постоянно задают самые разные вопросы. ISO 42010:2022 приводит пример таких вопросов (помним, что это стандарт документирования архитектурных описаний, но и все остальные описания будут отвечать на похожим образом сформулированные вопросы, только их предметы могут быть совсем не похожи):

- Как поддерживается система?
- Какое поведение системы критично для безопасности?
- Может ли целевая система достичь соответствия обязательным национальным стандартам?
- Какова стоимость эксплуатации?
- Каковы риски, возможности, удовлетворённость, выживаемость, согласованность, доступность по ресурсам, сложность и доверие предлагает данная архитектура?
- Какова прозрачность распределения, описанная в эталонной модели открытой распределённой обработки[216]?

Команда проекта описывает целевую систему и её окружение, а также системы создателей в их сложно устроенном графе так, чтобы и роли в команде проекта, и внешние проектные роли имели внятные ответы на эти вопросы. А ещё все участвующие в обсуждении роли предпринимают действия, чтобы реализовать свои предпочтения по теме интересов (иногда актёры, чтобы добиться предпочтений одних своих ролей включают и выполнение методов работы каких-то других ролей, помним, что агенты могут проявлять недюжинное творчество).

Роли действительно озабочены каким-то предметом, в их мышлении главенствуют эти трудовые/деятельностные «озабоченности», concerns — и они действуют исходя из намерения реализовать свои предпочтения в целевых характеристиках. Продавец будет пытаться поднять цену, покупатель-— опустить цену.

Каждая трудовая роль в зависимости от своей функции/вида труда имеет один или больше интересов (предметов интереса с предпочтениями в них) — при этом вполне возможно, что разные проектные роли/stakeholders имеют одни и те же предметы интереса и одни и те же, или разные предпочтения в них. Это очень удобно: если

---

[215] Stakeholders have interests in a System; those interests are called Concerns — http://www.iso-architecture.org/ieee-1471/cm/

[216] https://www.geeksforgeeks.org/types-of-transparency-in-distributed-system/

даже у одной роли два-три-пять предметов интереса, то общий список этих характеристик систем не будет вдвое или впятеро длинней списка проектных ролей, он будет много меньше. **Если нужно готовить для уторговывания предпочтений в интересах какие-то материалы/описания/модели, то их число будет пропорционально числу важных характеристик, а не числу предпочтений, не числу проектных ролей. И уж дальше предпочтения большого количества ролей будут уторговываться при помощи немногих описаний, отражающих небольшое количество предметов интереса.**

Принципиальная схема электросамоката как описание, отражающее некоторые его важные характеристики, будет использована многими и многими ролями: и теми, которые выступают за повышение::предпочтение его мощности::характеристика, и теми, кто выступает за ограниченное::предпочтение энергопотребление::характеристика, и теми, кто выступает за максимальную::предпочтение скорость разгона::характеристика, и теми, кто заботится о скоростной его зарядке. Это описание/модель позволяет уторговывать множество предпочтений для самых разных характеристик электросамоката как предметов интереса внешних и командных ролей проекта создания электросамоката.

При одинаковых предметах интереса/важных для ролей характеристиках **предпочтения** в значении этих характеристик для разных внутренних/командных и внешних ролей могут крайне различаться. Если встречаются роли «безопасник» и «продавец», то предметом их интереса у электросамоката наверняка будет «максимальная скорость», но вот предпочтения/оценки интереса будут разниться: для одного максимальную скорость нужно будет максимально опустить для достижения максимальной безопасности, а для другого максимально поднять, чтобы убедить в покупке, иметь преимущество в скорости перед самокатами конкурентов. Конечно, совсем необязательно появление пары ролей с такими разными предпочтениями для одного интереса, но такие пары встречаются более чем часто, и отнюдь не только в части интереса «стоимость» или «максимальная скорость».

Переговоры по значениям важных характеристик (иногда их называют **производственным конфликтом**) идут с любыми предметами интереса. Так, «вес электросамоката»::характеристика имеет предпочтение «поменьше» у потребителя, а у конструктора может быть интерес «вес побольше», чтобы разместить более ёмкую батарею, или использовать более прочную и в силу этого более тяжёлую конструкцию, и он признаёт, что «вес побольше» это плохо, но для него это «необходимое зло», для него это хорошо! Если отдать решение конструктору, то самокат может оказаться запредельно тяжёлый для потребителя, хотя и исключительно хорошо выглядящий по многим другим характеристикам. Лучше при этом будет интересы потребителя для веса::характеристика отстаивать маркетолог/продавец как член команды, чем принимать «взвешенное решение» сам конструктор.

Помним, что если обе проектные роли с конфликтующими предпочтениями по одной характеристике играет один актёр/исполнитель роли, и производственный конфликт происходит в одной голове, то говорят о **конфликте интересов**[217]. Разводить конфликт интересов по разным людям::актёрам полезно, ибо при

[217] https://en.wikipedia.org/wiki/Conflict_of_interest

продуктивном производственном конфликте роли контролируют ошибки друг друга, а когда этот контроль ошибок идёт внутри одной головы, он затруднён: актёр может предпочесть решение одной из ролей даже в том случае, если это решение ошибочно. При разведении конфликта интересов на роли для разных актёров, дискуссия по предпочтениям в одном предмете интереса идёт публично, и поэтому ошибки в рассуждениях могут найти и какие-то третьи лица, и они же могут дать предложения по решению производственного конфликта, принятие решения может быть документировано и проверено независимыми экспертами. Тем самым при отсутствии конфликта интересов (производственный конфликт внутри головы одного актёра) и переводе конфликта интересов в производственный/продуктивный конфликт могут проводиться более качественные согласования предпочтений в рамках производственного конфликта в разных головах разных актёров, ошибок будет меньше, предпочтения разных ролей будут учтены более качественно, система будет более успешна.

Но если пытаться это разделение разных конфликтов интересов по разным головам (если это люди, или даже если это подразделения) делать буквально со всеми выявленными конфликтами, то ничего не получится: конфликтов интересов огромное количество, и даже при учёте того, что одна голова может участвовать во множестве конфликтов интересов нужно всё-таки следить за тем, чтобы роли, занимающиеся интенсивными переговорами, находились организационно поближе друг ко другу-— переговоры были не всех со всеми, а внутри каких-то оргзвеньев (это будет обсуждаться дальше: хорошая модульность важна и целевых системах-продуктах, и в организациях создателей), поэтому в пределе-— внутри головы одного агента.

Но если просто считать, что тогда мы поднимем число голов в подразделении (грубо-— вместо одного человека поставим двоих, чтобы развести конфликт интересов, классический пример такого-— развести роли кассира и билетёра), то это не только удорожит, но и замедлит работу в больших проектах. Чем больше голов в проекте, тем меньше времени будет уходить не на работу (выработку решений, «выбор из разных альтернатив в одной голове»), а на бесконечные переговоры по конфликтам, что резко замедлит проект согласно закону Амдаля[218] (увеличение числа процессоров/голов для ускорения работы нельзя вести бесконечно, ибо всё больше и больше времени будет уходить не на собственно работу, а на то, чтобы работу разделить, потом согласовать, потом собрать результаты, в итоге производительность на одного работника/процессор существенно падает). Аккуратней с ростом числа работников и замедлением работы, это ведь верно и в случае «бесконечного числа работников, получаемых из AI-агентов».

Но если уж найден «козёл, охраняющий огород», то лучше бы функции охраны огорода и поедания капусты в таком важном случае отдать разным работникам, причём дело пойдёт немного медленнее, но если этого не сделать, то результаты будут быстры и хорошо предсказуемы в части непредвиденных дополнительных расходов на капусту.

Конфликты интересов могут быть не только в одной голове, но могут быть целые сообщества с конфликтом интересов. Не будем тут обсуждать Биг Фарму, которой выгодно выпускать лекарства, не вылечивающие пациентов, а принимаемые для

[218] https://en.wikipedia.org/wiki/Amdahl's_law

облегчения симптомов вечно, а также конфликт интересов врачей, которым выгодно назначать ненужные процедуры. Давайте возьмём пример из промышленности, картель Феб.

В мае 2022 года в США анонсировали (внимание! Отсутствует агент, который это сделал! Вы обратили внимание, что это безличное предложение? Какая роль того, кто или что обычно занимается таким «анонсированием» и что это вообще значит? Отлавливайте безличные предложения! В них нет ролей, действующих лиц, с ними трудно дальше работать!) законопроект, по которому будет запрещено (опять пропущен объект: кто::роль кому::роль запретит) торговать лампами накаливания (вот он: тип системы, с которым работаем, уровень вещества, даже ниже уровня киберфизической системы). Это делается (кем?! Опять пропущена роль!) в порядке стимулирования перехода на светодиодные лампы (кого стимулируем? Кто выигрывает? Но это точно выражение намерения). В этой связи в статье агентства Блумберг авторы (тут они выступают как критики предлагаемого регулирования: критики проекта социальных инженеров, ратующих за запрет лампочек накаливания) рассказывают историю картеля Феб (Phoebus, он же Аполлон - он был и богом света)[219].

Там была социальная инженерия уровня сообщества. Конкуренция производителей ламп накаливания с вольфрамовой нитью быстро привела к тому, что лампы стали более-менее вечными (время жизни лампы::важная характеристика/предмет интереса). Налицо конфликт интересов: производителям хорошо, когда лампы перегорают быстро::предпочтение (их будут покупать вечно!), и когда не перегорают долго::предпочтение (их будут покупать у тех, кто их умеет делать именно такими). И тогда была предпринята «социальная инженерия» среди производителей ламп: они организовали картель::«организационная система», который с огромными усилиями (помним, что при конфликте интересов действуют силы, тянущие одновременно в сторону разных предпочтений — и поэтому всегда есть и силы, заставляющие объединяться в картель, и силы, заставляющие выходить из картеля, отсюда поговорка, что «картели работают, картели работают недолго») заставил всех производителей делать лампочки накаливания с вольфрамовой нитью, не перегорающие до 1000 часов, но после этого таки перегорающие. Для этого была создана рабочая группа «1000 Hours Life Committee», организована лаборатория в Швейцарии, которая проверяла партии ламп на «правильное перегорание». И если перегорали лампочки раньше, то рынок таких убирал (покупатели умные! плохое не берут!), а если перегорали позже — то картель назначал штрафы. Если образцы ламп из партии горели без перегорания между 800 и 1750 часами, производитель мог продать остаток партии без штрафа. Но если лампы ухитрялись продержаться от 1750 до 2000 часов, виновная в этом нарушении компания платила штраф в 20 швейцарских франков за каждую тысячу проданных таких ламп. Штрафы повышались, если лампы горели дольше: если лампочка держалась 3000 часов, то это обходилось производителю в 200 франков штрафа на тысячу проданных. Система работала: перед началом работы картеля лампы держались в среднем 2500 часов без перегорания, а через десятилетие картель Феб сумел снизить эту величину до 1200.

Авторы статьи задаются вопросом про то, как намерение запретить продажу лампочек накаливания (которые до сих пор перегорают примерно так же за 1200

[219] https://www.bloomberg.com/opinion/articles/2022-05-04/cheering-for-led-bulbs-read-this-cautionary-tale-about-the-phoebus-cartel

часов, хотя картеля давно нет) связано с интересами разных ролей в проекте, и какой встречной социальной инженерии можно тут ожидать: при переходе на светодиодные лампы заявлено, что они почти вечные (ровно как это было с лампами накаливания в момент организации картеля Феб). Авторы интересуются: когда законодательно запретят конкурирующие с ними лампы накаливания, которые до сих пор перегорают через те же 1200 часов, не окажется ли, что у светодиодных ламп точно так же съедет срок службы с «почти вечного» на привычные 1200 часов? Ведь ситуация со светодиодными лампами и чистотой помыслов их изготовителей (неустроенность/frustration от взаимодействия желания потребителей иметь вечную лампочку и желания производителей ограничить срок службы лампы) ровно такая же, а конкурентное преимущество в долговечности перед лампами накаливания показывать будет не надо, обычных ламп накаливания не будет в продаже. Оставшиеся производители светодиодов договорятся: вечные светодиоды этим производителям будут поперёк горла, они начнут производить перегорающие.

Про антимонопольное законодательство, конечно, в тексте статьи ничего не говорится (кем? Авторами статьи! Ловите безличные предложения!) — оно ж про сговор по ценам, а не про время службы лампы! Социальная инженерия интересна тем, что все эти «социальные инженеры» работают друг против друга, конкурентно. Картели обязательно будут появляться и обязательно будут распадаться, но в техно-эволюции, некоторые находки будут удачными, и несмотря на то, что они не всем могут нравиться и явно не будут рациональными, они будут выживать и их будет много разных вариантов с примерно одинаковой ценой (проявление неустроенности/frustrations в силу попыток решить конфликты системных уровней). Это физика[220], объясняющая рост числа возможных примерно одинаково квазиоптимальных конструкций самых разных систем и общий рост сложности систем (рост числа системных уровней) по мере эволюции: на решение проблем нижних уровней влияют обстоятельства более высоких уровней, и наоборот: оптимизация решений в проектах должна быть многоуровневой.

Примеры типа консорциума Феб показывают, что **вполне можно разобраться в ситуации и на уровне выше уровня организации/проекта (сообщество, общество и даже «агентечество», поскольку агенты сегодня уже не всегда люди). Если использовать набор методологических понятий «агент», «актёр как агент-исполнитель роли», «роль», «предмет интереса/важная характеристика системы или создающего и развивающего её проекта», «интерес/предпочтение», «намерение», «стратегия»::метод, «план работ», то их должно хватить для подробного разбирательства с самыми разными ситуациями, связанными с поведением агентов, демонстрирующих в проектах иногда весьма извращённую «рациональность».**

Список методов описания из CPS PWG Cyber-Physical Systems (CPS) Framework

Предмет интереса/важная характеристика нам нужен, чтобы мы потом смогли сказать, как его описывать — **предмет интереса отражается/framed by методом описания/viewpoint.** Например, метод описания цены::характеристика может быть правилами составления сметы в рублях, а может быть правилами

[220] https://elementy.ru/nauchno-populyarnaya_biblioteka/434505/Konflikty_kak_osnova_slozhnosti, научный текст в https://www.pnas.org/doi/full/10.1073/pnas.1807890115

указания произвольно определённой цены, но в долларах или «условных единицах». Этот метод описания нужно будет использовать, чтобы описанный при его помощи предмет интереса (цена) был потом обсуждён разными ролями (продавцом и покупателем). Тем самым **дискуссия о методе описания цены** (как моделировать цену, как лучше её описывать, каким способом отразить знания о цене как характеристике системы) будет отличаться от **дискуссии о значении цены**, то есть о том, «как снизить цену» и «как поднять цену», хотя обе эти дискуссии оказываются существенно связанными (выбор метода описания обычно влияет на переговорную позицию одной из сторон: разные методы описания интереса удобны для продвижения разных предпочтений).

И когда мы только что **разделили одну сложную дискуссию по важной характеристике на две более простых (по методу описания и по уторговыванию предпочтений)**, задача стала немного проще — это и есть цель системного мышления. **В ряду понятий «метод работы», «роль», «предмет интереса», «предпочтение» нужно поставить ещё одно понятие: «метод/способ описания» (viewpoint).**

Разделение в мышлении методов, выполняющих их ролей, предметов ролевого интереса, предпочтений в значениях характеристик систем, метода описания предмета интереса и документирование состояния всех этих объектов-— упрощает удержание внимания в достижении договорённостей между ролями в проекте, облегчает коллективное выполнение проектов по созданию систем.

Мы не можем после составления списка предметов интереса/concerns и подготовки описаний/views по каким-то согласованным методам описаний/viewpoints забыть о том, чьи именно эти предметы интереса: сами интересы/предпочтения разных ролей для найденных общих предметов интереса/характеристик могут существенно отличаться, поэтому подготовленные описания системы — это лишь материалы для дискуссий.

Системное мышление коллективно — оно подразумевает непрерывные согласования предпочтений для самых разных характеристик, важных самым разным проектным ролям. Системное мышление предназначено для структурирования этих обсуждений и требует различать в них актёра, роль, предмет интереса, интерес/предпочтение (куда двигать характеристику, какое состояние предпочесть), метод описания, описание, значение характеристики как результат уторговки предпочтений.

В публичном документе (публичный документ отличается от стандарта только тем, что не предусмотрена процедура проверки на соответствие) CPS PWG Cyber-Physical Systems (CPS) Framework Release 1.0[221] приведена более полная, чем в ISO 42010 таблица предметов интереса для киберфизических систем, то есть систем, в составе которых есть датчики, эффекторы и управляющий ими универсальный-по-Тьюрингу компьютер. В этом публичном документе (выпущен в мае 2016 года) ввиду большой длины списка характеристик, они разбиты на группы, отражающие аспекты/aspects: функциональный, организационный, человеческий, доверия, времени, данных, границ, состава, жизненного цикла (functional, business, human, trustworthiness, timing, data, boundaries, composition, lifecycle).

[221] https://pages.nist.gov/cpspwg/

Аспекты/aspects/area of concerns/«области интереса» отражают какие-то содержательно сгруппированные предметы интереса. Длинная таблица CPS Framework в документе выглядит так (приведём её начало и конец):

Table 4: Concerns

Aspect	Concern	Description
Functional	**actuation**	Concerns related to the ability of the CPS to effect change in the physical world.
Functional	**communication**	Concerns related to the exchange of information internal to the CPS and between the CPS and other entities.
Functional	**controllability**	Ability of a CPS to control a property of a physical thing. There are many challenges to implementing control systems with CPS including the non-determinism of cyber systems, the uncertainty of location, time and observations or actions, their reliability and security, and complexity. Concerns related to the ability to modify a CPS or its function, if necessary.
Functional	**functionality**	Concerns related to the function that a CPS provides.
Functional	**manageability**	Concerns related to the management of CPS function. For example, Managing Timing in complex CPS or SoS is a new issue with CPS that did not exist before. It is being developed with new standards
Functional	**measurability**	Concerns related to the ability to measure the characteristics of the CPS.

•••

Lifecycle	**maintainability**	Concerns related to the ease and reliability with which the CPS can be kept in working order.
Lifecycle	**operability**	Concerns related to the operation of the CPS when deployed.
Lifecycle	**procureability**	Concerns related to the ease and reliability with which a CPS can be obtained.
Lifecycle	**producibility**	Concerns related to the ease and reliability with which a CPS design can be successfully manufactured.

Как использовать таблицу? Эта таблица CPS Framework работает как чеклист: все ли аспекты/«области интереса» систем (целевой и дальше систем создания) вашего проекта вы обсудили, в них нашли все ли предметы интереса?

ISO 42010:2022 говорит, что аспекты — это отражение «лучших практик» (best practices, иногда good practices, лучшие или просто хорошие, известные на сегодняшний момент методы работы) в том, чем должны интересоваться специалисты в своих предметных областях, по ним как менее детально определённым предметам интереса проще договориться, проще их вспомнить в ходе проекта-— и уже углубляться в отдельные предметы интереса, исходя из особенностей конкретного проекта. Помним, что «объективность-— это хорошо организованная субъективность», так что аспекты просто отражают мнение

экспертов о том, что важно при создании и развитии каких-то систем, а «лучшая практика»-— это «лучший на сегодняшний день метод работы». Список аспектов из CPS Framework был «лучшим» (хотя и с этим можно было бы поспорить) на май 2016 года. С тех пор, например, вместо понятия «жизненный цикл» всё чаще используется «создание и развитие системы», ибо однократное прохождение разработки «от рождения до смерти» заменилось на создание MVP, то есть minimal viable product, минимально жизнеспособный продукт, и последующее непрерывное развитие без явного учёта окончания этого развития.

После того, как вы нашли предметы интереса, надо подобрать для каждого предмета интереса его метод описания (ссылка на какой-то стандарт или метод описания, созданный и документированный самостоятельно). Дальше вам надо определить: где::софт самые разные проектные роли могут найти актуальное на текущий момент проекта документированное описание состояния дел по поводу выбранного предмета интереса.

Скажем, для функциональной области интереса (functional aspect) важны функции целевой системы (functionality). Это вы вряд ли забудете выяснить. В руководстве по системной инженерии будет указано, что такое описание (view)-— концепция использования. Но вам нужно обязательно пройти дальше: какой будет метод описания функциональности? Скажем, вы выбрали для небольших проектов JTBD или customer journey (если не знаете, то гуглите, или спросите AI-агента! Если вы профессионально занимаетесь созданием концепций использования, то есть описываете функциональность систем, то такое нужно знать без заглядывания в гугл или вопросов к AI-ассистенту, надо иметь профессиональный кругозор). Теперь надо документировать выбранный метод описания и указать место (моделер-— и адрес, где в нём найти), где будут лежать самые свежие версии описания функциональности. Заодно придётся ответить на вопросы вроде «а если две команды разработчиков занимаются разными сценариями использования и делают независимые модули для их поддержки-— это должно быть одно место или два?».

И эти вопросы задаются по всем остальным аспектам (они же области интереса) и в них всем ключевым характеристикам (они же предметы интереса). Например, для области интереса создания и развития системы (заменяем термин lifecycle из CPS Framework) выбираем предмет интереса «эксплуатируемость» (operability, архитектурная характеристика). Если вы инженер-профессионал, то вы знаете, что именно вас должно там интересовать. Если не профессионал, то будете «изобретать велосипед». Но лучше уж изобрести велосипед, чем забыть! Предупредим: это очень непростой вопрос! Скажем, вам придётся как-то включить в рассуждения цифрового двойника-— это ж как раз для улучшений в эксплуатируемости? Теперь вам надо выбрать метод описания («предметно-ориентированный язык»/DSL/«подход к моделированию»/framework/онтологию-— или взять из литературы, или создать самостоятельно для целей проекта). Определить софт (не ваш, а тех агентов, которые будут зачем-то изменять эти описания и зачем-то смотреть эти описания) как место, в котором будет храниться документированное описание. Затем документировать это описание, стараясь учесть все предпочтения по поводу эксплуатируемости-— а у проектных ролей (внутренних и внешних) будут самые разные предпочтения. И проследить, чтобы это описание как-то претворилось в жизнь, отразилось в физически воплощённой системе (имеется ввиду не появилось внутри документированным, а отразило то, что вы описывали: воплотить в жизнь описанный трактор — это не документировать описание

трактора, а сделать работающий железный трактор). Заведомо не надо порождать оторванные от реальности описания, если только вы не писатель-фантаст.

Разговаривайте не с людьми, разговаривайте с ролями

Вы должны по действиям и высказываниям актёра/«агента-исполнителя роли» определять его метод работы и по этому методу работы определять роль. Это делается независимо от того, как называется этот исполнитель проектной роли в жизни — какая у него должность, как он называет себя сам, как он именуется в каких-то документах проекта, как называют его окружающие его люди, и вообще человек ли это или организация, или робот какой-то организации. Роза пахнет розой, хоть розой назови её, хоть нет. Если человек ремонтирует унитаз, то это роль сантехника, даже если ремонтирующий унитаз человек называет себя хирургом-кардиологом. Надо смотреть на то, что делают агенты, а не на их названия — названия могут быть выбраны из самых разных соображений.

Дальше вам надо обязательно понять, какой в данный момент предмет интереса у роли и выяснить ролевые предпочтения для этого предмета интереса. После этого вы поймёте, какие у роли союзники и противники (тоже роли), а также определитесь сами (у вас же тоже есть роль, в которой есть свои предметы интереса и предпочтения, вам же тоже надо качественно выполнять какой-то метод работы этой роли!)-— вы союзник или противник.

Затем вам надо найти подходящий метод описания/viewpoint для обсуждения предмета интереса. **Метод описания (viewpoint)** — это как легенда карты, показывает то, какие объекты и их отношения будут изображены на карте-описании. Как делать и документировать описания разной степени формальности, вы уже узнали из руководства по рациональной работе.

Потом в своих высказываниях, документах, действиях приводить описания (саму карту в соответствии с выбранной легендой), чётко придерживаясь предмета интереса вашего собеседника::роль и показывать, что будет происходить в проекте с реализацией его предпочтений в значении важной для этой его роли характеристики, а ещё проводить переговоры с другими агентами, ибо нужно будет учитывать интересы и других ролей.

Если вы услышали что-то про финансы::«предмет интереса», то вы должны понять, что перед вами финансист::роль, при этом нужно абстрагироваться от того, как выглядит актёр, реализующий::отношение финансиста, то есть проигнорировать для текущего рассуждения рост, возраст, пол, вероисповедание, интонации и степень озлобленности, хотя нужно понять мастерство этого актёра в финансовых рассуждениях. Вы должны понять дальше, какая важная характеристика интересует сейчас финансиста, а затем выбрать адекватно отражающий её метод описаний: баланс, отчёт о прибыли и убытках, кэш-флоу — это три разных метода описаний для ответа на разные предметы интереса в рамках общего предмета интересов «финансы». **Не ошибитесь, когда выбираете метод описания: если вопрос финансиста о кассовом разрыве, то бесполезно разговаривать о балансе! Поэтому надо иметь какой-то методологический/трудовой/инженерный кругозор по методам работы многих ролей, которые нужны для успеха проекта. Минимально—освоить работу по инженерным руководствам (по системной инженерии, инженерии личности, системному менеджменту) программы рабочего развития мастерской инженеров-менеджеров, хотя**

этого совсем недостаточно, но лучше, чем вообще не иметь кругозора по методам инженерии.

Это всё верно не только при общении с актёром-человеком, но и с роботами, и с организацией (когда вы задаёте вопросы через чат и вам присылают ответы, или наоборот — вам задают вопросы, а вы присылаете ответы, то неважно, кто там на другом конце чата: сколько людей разговаривают, и люди ли это или робот).

Разговаривайте всегда не с агентом (человеком, фирмой, их компьютером, их анкетой), а с ролью, хотя и отслеживайте эмоциональное состояние агента (если это человек) и ситуацию с ресурсами разговора. Скажем, если вы едете в лифте, то не стоит заводить разговор на пару часов, это бесполезно, а ещё бесполезно беседовать с Васей Пупкиным о его отношении к Дездемоне в роли Отелло в тот момент, когда Вася полностью погружён в отыгрывание Принца Гамлета: ответом вам может неожиданно быть «я спрошу Дездемону, быть или не быть»-— и не удивляйтесь, это вы плохо разобрались с ролями и смутили Васю своим вопросом.

Важно не столько давать ответ на задаваемый актёром в непонятной роли конкретный одиночный вопрос, сколько в целом отвечать на интерес его текущей проектной роли, давая развёрнутое описание предмета интереса по выбранному (и, желательно, согласованному и объяснённому) методу описания так, чтобы актёру были понятны шансы реализовать предпочтения его роли. Отвечайте не на сам высказанный вопрос, а прямо на интерес! Вас спрашивают: «вы использовали золото в составе вашей системы?», но после понимания «откуда такой вопрос» (в чём интерес роли) правильный ответ не «использовал/не использовал», и даже не «золота там полтора грамма на три тонны системы», а «цена экземпляра системы будет не более $300 при заказе от 1000 штук». И это сократит время разговора в разы!

Если интересы/предпочтения у разных ролей даже одного актёра по поводу одного интереса разные, и тем более разные предпочтения разных ролей, исполняемых коллективом актёров, это повод для переговоров по поводу уторговывания 1. сначала метода описания, это одна тема переговоров и 2. самих значений важной характеристики, о которой нужно договариваться. Это повод для изобретательской деятельности в ходе этих переговоров (методы конфликтологии[222], изобретательства[223] тут могут помочь очень существенно).

Это типичный способ задействования методологического (в данном случае) мышления: важные объекты и их связи из мета-мета-модели дают чеклист того, о чём нужно обязательно подумать, и что вам обязательно нужно сделать в ходе разговора:

- Увести ваше внимание с актёра на его метод работы, определить метод по делам и словам.
- Переобозвать агента по роли его метода работы.
- Понять предмет ролевого интереса собеседника::роль.
- Понять предпочтение собеседника::роль в предмете интереса (скажем, какой-то архитектурной характеристике).
- Разделить переговоры по отражающему предмет интереса методу описания

[222] https://en.wikipedia.org/wiki/Conflict_resolution
[223] https://ru.wikipedia.org/wiki/Теория_решения_изобретательских_задач

и значению важной характеристики (одни сложные переговоры по двум темам сразу много хуже двух более простых по разным темам).

- Учесть в этих переговорах наличие других ролей с аналогичными предметами интереса, включить актёров, играющих эти роли, в переговоры. **Нельзя «договориться со всеми»! Любая потом договорка этих «всех» между собой сорвёт их договорённости с вами. Единственный выход-— договорить всех вокруг себя!** Предмет интереса может быть у этих «всех» один и тот же, но предпочтения — разные! И ещё по ходу проекта могут меняться приоритеты в интересах: в какой-то момент «цена решает всё», но потом через месяц окажется, что «за ценой не постоим, но вот вес системы решает всё!», будьте к такому готовы.

Этот чеклист — просто представление в другой форме материала предыдущих абзацев. Метод/способ его применения нарабатывается в ходе каждого разговора, участия в каждом совещании, работы в каждом коллективном проекте. Это типовое использование материала по фундаментальным методам мышления интеллект-стека (понятизация, семантика, онтология, этика, риторика, методология, системная инженерия тут только некоторые из них): объекты дисциплин/теорий этих фундаментальных методов мышления позволяют рутинно рассуждать о том, что важно в текущей ситуации, игнорировать неважное и тем самым существенно экономить силы и время. Так, в ходе переговоров кроме вот этого ролевого рассуждения вы будете задействовать и культуру риторики (убеждающей речи), в которой обратите внимание и на обращение к нейросети вашего собеседника, а не к сухому компьютеру (то есть будете управлять уровнем формальности высказываний, осознанно использовать разные теории понятий в ваших высказываниях), будете учитывать эмоциональное состояние агента, и оцените его текущий уровень знания (возможно, что-то потребуется объяснять дополнительно, особенно в случае низкой квалификации актёра-собеседника в исполнении его проектной роли). И учитывать, с людьми ли вы вообще говорите, и знакомы ли вы лично, или общаетесь «с неопределённым кругом лиц» или даже компьютерной «нежитью».

Зачем согласовывать интересы? Помним: системное мышление говорит, что создаваемая система и её проект будут успешны только при согласованных значениях важных характеристик, при согласованных интересах/предпочтениях по поводу системы. Существенно уменьшают время согласования интересов, уменьшают сложность этого труда:

- Разговоры не с людьми::актёрами, а с их ролями по поводу предметов их интересов и методов описаний этих предметов интереса, самих интересов как предпочтений в значениях важных характеристик системы,
- Предсказание намерений актёров и возможных их стратегий и планов по реализации намерений в своих явно и неявно проявляемых ролях,
- Разделение сложной коммуникации на отдельные вопросы (принцип «разделения интересов»/«separation of concerns» будет обсуждён чуть позже),
- Использование системного мышления для нахождения необходимых в обсуждении предметов интереса (это мы тоже затронем чуть позже в нашем руководстве),
- Задействование полного интеллект-стека (например, риторики как метода убедительной речи).

После перехода на такого сорта «ролевые разговоры» (вместо разговоров с актёрами) с вами будут разговаривать те люди::актёры, с которыми вам раньше поговорить не удавалось. Они раньше не понимали вашей пользы для прояснения ситуации с областями их интереса, реализации их предпочтений в важных характеристиках участвующих в проекте систем (в том числе характеристиках многочисленных систем создания), поэтому им было **неинтересно** с вами общаться. А всего-то нужно признать, что **люди-исполнители ролей действуют не по своим прихотям как «люди» или AI-агенты, а действуют в соответствии с диктуемыми их ролями предпочтениями в диктуемых ролями важных характеристиках. Людям-актёрам интересно не всё, но от предпочтений в их ролевых предметах интереса они не откажутся, о своих ролевых интересах не «забудут» — эти ролевые интересы/предпочтения культурно-обусловлены, и в их учёте нужно опираться не столько на личные особенности агента, особенно если это живой человек, сколько на наработанную человечеством культуру труда играемой актёром роли.**

Моделирование: использование чеклиста для отслеживания ролей

Участвуйте в трёх совещаниях, используйте в разговорах на этих совещаниях чеклист из предыдущего моделирования «внешние проектные роли и успешность системы и её проекта», где говорилось «Выпишите от 15 внешних ролей разных агентов (людей и организаций) вашего текущего проекта (если один человек или организация играют несколько ролей — выписывайте все) и их интересы. Проставьте ваше ожидание (по десятибалльной шкале) по выполнению выписанных ролевых предпочтений по итогам реализации проекта». Положите текст чеклиста перед собой и следуйте ему для каждого участника совещания, включая себя. От совещания к совещанию ваше мастерство в удерживании внимания на ролях, их предметах интереса и самих интересах, методах описания предмета интереса должно увеличиваться. Опишите ваши впечатления в таблице.

Номер совещания	Роли, отслеживаемые по чеклисту	Результаты	Трудности	Что можно было бы сделать лучше
1.				
2.				
3.				

Вопросы. Проектные роли и их предметы интереса

Дима определил, что устройство будет состоять из держала, крепила и мерила, соединяемых друг с другом винтами с левой резьбой. Какая проектная роль Димы в этой ситуации?

--- Менеджер ### Нет, обычно менеджер не определяет конструкцию изделия.

+++ Инженер-технарь ### Да, именно инженер-технарь определяет конструкцию изделия. При этом в руководстве мы обычно используем слово «инженер» много шире, в том числе менеджер — это «инженер

организации», а ещё используем часто неологизм «инженер-менеджер», чтобы указать на частое совмещение агентами в коллективных проектах ролей прикладных инженеров и менеджеров.
--- Инвестор ### Нет, обычно инвестор не определяет конструкцию изделия.
--- Студент ### Нет, обычно студент не определяет конструкцию изделия. Если сказать, что «студент мог бы это делать в учебном проекте», то агент Дима выполнял бы роль студента в учебном проекте (были бы совсем другие слова — «пара», «учебник», «упражнение»), но вот это действие он бы делал (уж как мог) не как «студент», но как «инженер-технарь».
--- Директор предприятия ### Нет, директор предприятия вообще не проектная роль, это должность.

Директор дивизиона гаджетов задал, что разработка устройства будет приоритетом в работах второго квартала. Какая его проектная роль в этой ситуации?
+++ Менеджер ### Да, приоритеты по срокам — обычно интерес подроли операционного менеджера («оператора организационной системы в момент её эксплуатации»). А «директор дивизиона» в отрывке — это актёр, названный так по имени его должности. Но он играет роль менеджера.
--- Инженер-технарь ### Нет, инженеры-технари обычно не занимаются расстановкой приоритетов в работах, это не их интерес.
--- Инвестор ### Нет, инвесторы обычно не занимаются расстановкой приоритетов в работах, работы — это не их предмет интереса.
--- Аналитик ### Нет, аналитик обычно ничего задать не может, может только предложить (но если он предлагает, то это уже не анализ, а синтез, то есть он тогда не в роли аналитика, а «аналитик» — это тогда должность, но всё равно не роль).
--- Директор дивизиона ### Нет, директор вообще не проектная роль, это должность. В задаче актёр назван по имени его должности, а спрашивают по роль.

Автандил Николаевич заявил, что его интересует эргономика системы и её размеры. Как правильно назвать его в этой ситуации?
+++ Пользователь ### Да, обычно пользователей интересует хорошая эргономика, и ещё их вполне могут интересовать размеры системы.
--- Финансист ### Нет, их интересуют совсем другие вопросы, но не эргономика и размеры.

--- Логист ### Нет, размеры ещё могут как-то интересовать логиста, но уж точно не эргономика.

--- Пофигист ### Нет, пофигистов вообще ничего не интересует. Это не проектная/функциональная/деятельностная роль, не функциональная позиция, его интересы можно в проекте не учитывать. А в задаче Автандил Николаевич заявил свои предметы интереса, он не пофигист.

--- Плательщик ### Нет, плательщика интересуют совсем другие вопросы, и это не эргономика и размеры.

--- Ремонтник ### Нет, ремонтников интересуют совсем другие вопросы, не эргономика и размеры.

Тихая блондинка сидела молча всё совещание, а потом заявила, что её интересует возможность получения оплаты за разработку системы четырьмя траншами по 25% каждый. Какая её проектная роль в этой ситуации?
--- Пользователь ### Нет, пользователя не интересуют особенности оплаты за разработку, его интересует польза от системы.

+++ Финансист ### Да, особенности оплаты как раз предмет/область интереса финансистов.

--- Логист ### Нет, это не вопрос логистики — ничего никуда перевозить и вовремя доставлять не нужно.

--- Пофигист ### Нет, пофигистов вообще ничего не интересует. Это не проектная роль, так что пофигиста можно вообще в проекте не учитывать.

--- Плательщик ### Нет, плательщик платит, а не получает оплату.

--- Ремонтник ### Нет, вопросов/интересов/предпочтений в ремонте тут нет.

Директор завода спросил: «каким транспортом будет доставляться система с завода в пункт установки». Как правильно назвать его в этой ситуации?
--- Пользователь ### Нет, пользователя интересует нанесение непоправимой пользы/использование, а не доставка.

--- Финансист ### Нет, это не финансовый вопрос/предмет интереса, а логистический (способ, которым осуществляется доставка).

+++ Логист ### Да, это вопрос логистики, занимающий должность директора человек::актёр высказал предмет интереса логиста::роль.

--- Пофигист ### Нет, пофигистов вообще ничего не интересует, так что он не может быть проектной ролью. Пофигистов в проекте игнорируют, они обычно вне проекта.

--- Плательщик ### Нет, это не интерес оплаты, актёра на должности директора не интересует такая важная для плательщика характеристика как «оплата», его интересует «транспорт для доставки».

--- Ремонтник ### Нет, интересов к каким-то вопросам ремонта тут нет.

Конструктор сказал, что он послал один и тот же обновлённый эскиз троим агентам-в-ролях, заявившим интерес к внешнему виду изделия: пользователю, маркетологу, промышленному дизайнеру. Мог ли он так сделать, или нужно было для каждой проектной роли подготовить специальный вариант эскиза?
+++ Мог, их может удовлетворить один и тот же эскиз ### Да, несколько проектных ролей могут иметь один и тот же предмет интереса (хотя в этих предметах интереса они могут иметь разные предпочтения), поэтому они могли удовлетвориться одним и тем же эскизом для рассмотрения (но по поводу этого эскиза могут иметь самые разные мнения, когда они его рассмотрят).

--- Не мог, каждому нужно что-то своё ### Нет, это совсем необязательно: несколько проектных ролей могут иметь один и тот же предмет интереса, обсуждаемый при помощи одного и того же описания.

Позиция

Когда человек-исполнитель долго удерживает какую-то одну роль, иногда действуя согласно интересам этой роли даже в ситуациях, когда от него ожидаются исполнение других ролей и тем самым преследование других интересов (т.е. на первом плане в любых ролях оказываются предметы интереса и предпочтения этой роли в соответствующем методе работы), то это называется — **позиция** (понятие системодеятельностных методологов[224], оно почти эквивалентно понятию методологической/проектной/деятельностной/организационной роли, но имеет свои особенности). При этом не путаем **«1. позицию»** как термин надёжного удержания роли (осознанного или неосознанного) и другое словарное значение-— **«2. позиция» как должность в организации. Должность не обсуждаем, продолжаем обсуждать роли.**

Когда агент-актёр не только исполняет роль, но и занимает ролевую позицию «менеджер», то у него менеджерские ценности/предпочтения в каких-то важных для его роли характеристиках начинают проявляться и тогда, когда он организовывает инженеров-технарей в рабочем проекте, и когда он воспитывает детей, и когда он сидит в парламенте. Когда этот актёр в позиции «родитель», то у него воспитательные (скажем, привитие каких-то моральных ориентиров окружающим) предметы интереса/важные характеристики и предпочтения/интересы и дома среди детей, и в рабочем коллективе, и на шумной вечеринке.

Позиции чаще всего занимают неосознанно (и тогда такими **позиционерами** как агентами, занявшими позицию, легко манипулировать: любые их действия легко вычислимы, ибо действуют уже не они сами, у них вполне предсказуемая ролевая/функциональная позиция и её ценности, и они её сменить не могут, не могут выйти из роли). Реакция исполнителя такой «застрявшей деятельностной роли» на явное указание его неосознанно занятой позиции бывает разная: «что-то застряла роль в сознании, спасибо, что обратили моё внимание» или наоборот «какая такая у меня позиция? как так у меня не меняются в разных делах роли? я ведь такой спонтанный и разный, чем горжусь!».

---

[224] http://praxos.ru/index.php/СМД-методология

Но длительное пребывание в роли у агента может быть намеренным и осознанным — осознанное удержание внимания на интересах роли, собранность, «удержание позиции». Если это «неосознанная собранность», то это, скорее, «разобранность» (утеря управления своим вниманием), «неосознанное застревание в роли», «неосознанное удержание позиции». В любом случае, **позиция — это длительное и явное удержание роли, намеренное или не намеренное.**

Тут нужно обратить внимание и на сам термин проектная/функциональная/деятельностная роль: «проектная роль» означает роль в каком-то проекте, но сама роль берётся не из проекта, а из какого-то культурно-обусловленного (часто ассоциируемого с «профессией», хотя тут речь идёт просто о квалификации в каком-то мастерстве) методе работы Деятельностная роль — роль в деятельности/ «виде труда», это всё синонимы с незначительными нюансами смысла, но это всё отсылки к общеизвестному разделению труда. Роль «плотник» берётся не из проекта «создание собачьей будки», а из человеческой культуры, это функциональная роль — но в проекте по созданию собачьей будки это будет проектная роль! Чтобы не запутаться, всегда можно проверить: если обе эти роли играются одним и тем же физически существующим Степаном Никаноровичем в один и тот же промежуток времени, то речь идёт об одной и той же роли — просто разные названия подчёркивают разные аспекты этой роли: проекта, где эта роль задействована, и деятельности, в которой эта роль определена.

Можно и дополнить ряд синонимов: **методологическая/проектная/функциональная/деятельностная/практическая/трудовая/культурная/организационная/инженерная роль**. Если хотите подчеркнуть то, что следуете системной инженерии как нормативному методу, то это будет **инженерная роль** — и даже какой-нибудь «политик» тут будет в инженерной роли (социального инженера). Если эти роли длительно кем-то удерживаются — это будут уже не роли, а методологические/проектные/деятельностные/практические/трудовые/инженерные/культурные/организационные позиции.

Занятие позиции связано больше с «методом работы»/«видом деятельности»/«видом труда»/«видом инженерии»/«видом субкультуры» (иногда говорят про метод работы в целом, деятельность в целом, труд в целом, но иногда подчёркивают, что у одной роли чаще всего задействуется только часть всех вариантов её возможных методов/деятельностей/труда/инженерии/практик, тогда просто вставляют «вид» или «вариант»-— «вариант метода», «вид труда», «вид культуры» или даже «вариант субкультуры», чтобы подчеркнуть прикладность рассуждения для какой-то разновидности метода), но меньше связано с проектом. Занявший позицию человек будет обычно занимать её осознанно и не осознанно не в одном проекте, а во многих проектах — это что-то типа «выбора профессии» (осознанный выбор), или «обнаружения себя в профессии» (неосознанный выбор), а не чисто ситуационный выбор «в проекте создания конкретной системы». Агент в роли «плотник»-— он во всех проектах будет «плотник проекта», если удерживает позицию, но позицию мы его будем называть при этом деятельностной/трудовой, а не проектной. Это, конечно, связано с тем, что роль выбирают и позицию занимают не любую, а только ту, для выполнения методов работы которой у агента есть мастерство. Если хочется занять роль парикмахера или инженера-теплотехника, то без мастерства в методах стрижки волос или методов инженерии теплообменников

это не получится.

Можно и нужно занимать позицию осознанно: «сейчас стратегически займу такую-то позицию», то есть «выберу себе понятную роль в понятной пьесе/деятельности/практике/игре/культуре, и буду удерживать внимание на предметах интереса и предпочтениях этой роли в самых разных проектах, пока не передумаю и самостоятельно явно не выйду из этой роли» (и где-то там рядом будет обучение мастерству выполнения метода работы выбранной роли). Стратегически-— это означает, что в ходе стратегирования было принято решение придерживаться какого-то метода работы и удерживать его во внимании (подробности по стратегированию как выбору метода работы смотри в руководстве по методологии, а в руководстве по системному менеджменту есть подробности по организационному стратегированию, причём причём стратегирование оказывается равноприменимым и к агентам-людям, и к агентам-организациям).

Осознанный выбор агентом (например, человеком) роли/позиции в ходе стратегирования обычно называется **самоопределением**[225] и очень ценится: **вы становитесь предсказуемым в своём поведении для людей, обладающих трудовым кругозором. Им становится понятно, как с вами сотрудничать, что вы можете делать, а что делать не будете. Самоопределение значительно облегчает вхождение в различные проекты, улучшает работу в команде. Помним, что мало просто самоопределиться, нужно ещё иметь мастерство в выполнении метода занятой позиции, а ещё и иметь собранность для удержания этой позиции в ходе проекта.**

Когда исполнитель скачет по разным ролям в одном проекте, как зайчик, то с ним очень трудно договариваться: внешний эффект при этом такой, будто он непрерывно меняет свои предметы интереса и предпочтения характеристик в них — что было для него ценным в его предыдущей роли пять минут назад, вдруг перестаёт быть значимым, но зато появляются какие-то новые претензии к давно уторгованным всеми остальными ролями значениям характеристик. **Человека без явно предъявляемой роли и ролевых предпочтений можно назвать «какой гибкий человек, никто его подловить не может», а можно (и чаще всего так и называют!) — «какой скользкий». Поэтому лучше бы быть собранным и как-то пытаться занимать позицию, а не просто «играть роли» по одной в каждую секунду.**

Люди как актёры относительно редко сознательно и намеренно избегают чётко занимать позицию, затем выполняя работы в каком-то чётко понятном методе, чтобы сознательно стать непредсказуемыми, чтобы их ролевые предметы/зоны/области интереса были скрытыми, намерения и стратегии (методы/способы работы) их как агентов были неявными, и поэтому вытекающие из них действия нельзя было просчитать. Нет, в подобных случаях люди::актёры просто плохо ориентируются в своих ролях, не осознают их (не знают, что от них ожидают игры по роли, а не «общего активного участия» в проекте), у них нет никакого «намеренного избегания в удержании роли», но их внимание блуждает по разным предметам обсуждения, у них отсутствует собранность, и они этого сами не замечают.

Бывает, что необходимость игры по роли актёрами осознаётся, но они просто не имеют мастерства, необходимого для исполнения своей роли: плохо знают роль, не

[225] https://shchedrovitskiy.com/nabroski-k-bazovoy-sheme-samoopredeleniya-1/

знакомы с трудовой культурой, т.е. не читали учебников и не получали иного образования; не имеют опыта игры в роли, ибо не накапливают время нахождения в роли; не имеют трудового кругозора, и поэтому не понимают, о чём договариваться с актёрами, исполняющими другие роли в этом проекте.

Несобранные люди будут неустойчиво удерживать позицию, неустойчиво воспроизводить ролевое поведение — и это будет проблема для проекта. Функциональная/профессиональная/деятельностная/трудовая/проектная/инженерная/практическая/методологическая/организационная/культурная роль может меняться у человека даже в ходе произнесения одной фразы — начало фразы будет, например, от роли Принца Гамлета или операционного менеджера, а конец фразы от роли Отелло или стяжателя, желающего урвать бонус к своей зарплате.

Перескоки между ролями возможны на роли из совсем другой пьесы или игры (как в случае Принца Гамлета и Отелло, операционного менеджера и стяжателя) — и да, это может быть и проявлением конфликта интересов. Или вдруг вылезает роль партийного функционера, который решил провести политику своей политической партии прямо на рабочем месте. Системный мыслитель должен отслеживать такие ситуации и использовать методы лидерства (в театре это по отношению к актёрам делает режиссёр), помогая исполнителям ролей освоиться со своими ролями и фокусируя их на выполнении ролевой работы в проекте, а не просто произвольной «жизни в проекте».

Многие люди воспринимаются как надёжные (и реально ими являются) потому как «удержание ими роли»/«занятие позиции» происходит у них уже автоматически, как привычка их мышления. Они автоматически придерживаются «набора ценностей» (то есть интересов, предпочтений в значениях важных характеристик, мы сознательно редко используем слово «ценности», потому как все его понимают по-разному и оно сбивает с толку, иногда означает value/пользу, а не «значения важных характеристик», иногда речь вообще идёт о морали) своей роли, в которой они застряли, ценностей методов работы, которыми долго занимаются. Поэтому они выглядят как принципиальные люди, отстаивающие какие-то свои принципы. Но если у них своего дела/деятельности/методов работы и выполняющей эти методы работы позиции/роли нет, то они могут так же бессознательно «не держать позицию», и выглядеть поэтому скользкими и беспринципными: они никогда не «Принцы Гамлеты», они всегда Васи Пупкины, с ними невозможно играть ролевые игры для нескольких ролей, то есть участвовать в разделении труда (работе разных агентов по разным методам работы, в которых они имеют разное мастерство). А проектов без разделения труда сегодня и не бывает.

Люди (мы говорим тут главным образом о людях, а не об AI или организация), которые осознают выполнение ими своих (профессиональных, социальных, семейных) ролей, могут выбирать — занимать ли им надолго какие-либо позиции, а также наоборот, расставаться ли с занятыми позициями тогда, когда потребность в них прошла. Люди, которые осознают чужие роли, часто могут понять предметы интереса и ролевые предпочтения/мотивы/ценности тех или иных действий и высказываний людей, ровно потому что они думают не про самих людей, а про исполняемые ими роли. Про самих людей::агенты, их намерения, личные стратегии, намерения и планы им мало что известно, а вот про деятельностные роли в культуре обычно известно довольно много, плюс всегда можно уточнить свои знания (например, погуглить или спросить у AI, если роль редкая).

В большом числе случаев «позиция» определяется профессией, часто используемым в работе надёжно освоенным мастерством. **Названия распространённых ролей в проектах и организациях — это очень часто названия профессий (каких-то занятий/методов/«видов труда», всё более и более узких по мере углубления разделения труда): менеджер и более узкий в части решаемых задач операционный менеджер, классический системный «железный» инженер и более узкий инженер-архитектор, продавец и более узкий аккаунт-менеджер (занимающийся крупными клиентами и возможностями по заключению сделок с этими клиентами). Углубление разделения труда как раз про это. Артист, более узкий артист цирка и даже ещё более узкий дрессировщик, специализации становятся всё более и более узкими. Роли — это никогда не названия должностей, не названия «руководства и подчинения», никаких «ведущий артист» или «главный артист», роли указывают на используемый метод работы, они не указывают ни на степень мастерства агента-исполнителя роли в работе по этому методу, ни на полномочия агента-исполнителя роли выполнения такой работы с задействованием каких-то ресурсов.**

В самых разных инженерных («железной» инженерии, инженерии личности, инженерии организации, инженерии сообществ) проектах необходимо по возможности письменно моделировать позицию и степень мастерства всех агентов — компетентность исполнителя роли в проекте может как соответствовать требуемой роли, так и не соответствовать. «Беда, коль пироги начнёт печи сапожник, а сапоги тачать пирожник» — беда, коль исполнитель с мастерством сапожника в вашем проекте таки будет печь пироги, ибо «других исполнителей у нас нет». Вкус и вред таких пирогов вы легко можете себе представить.

Всегда разделяем человека и его роль в проекте: оцениваем, подходит ли человек-исполнитель на конкретную роль по степени его мастерства/компетентности в методах работы роли, но ещё и собранности (ролевое мастерство, ибо что толку с умеющего играть на рояле, когда он не может за этим роялем усидеть и пяти минут?). Поэтому на всех совещаниях и при прочтении всех документов проекта нужно отдельно понимать:

- какой метод должен применять исполнитель роли, где этот метод описан,
- какова его культурно-обусловленная роль, как её можно назвать,
- занимает ли исполнитель роли позицию, или «ускользает» от позиционирования, так что по отношению к нему требуется лидерство как метод помощи в удержании роли/занятии позиции,
- какова степень его мастерства в выполнении метода работы по роли, которую он должен играть и насколько вы сами мастер в применении этого метода, чтобы оценить его уровень мастерства.

Это понимание должно быть абсолютно осознанным и его желательно документировать, ибо удержать это всё в уме для множества ролей сложно. Затруднения с документированием вашего понимания чьей-то и даже своей роли и выбранного варианта метода работы часто показывают недостаточную продуманность вопроса — «собака всё понимает, но сказать не может», рука зависает над клавиатурой, но не пишет! Помним о методе описания (какую табличку с указанием типов в колонках надо заполнить), это может помочь.

Системный мыслитель должен также чётко понимать, что обычно и он сам в проекте играет какие-то роли, у него есть какая-то профессиональная/компетентностная позиция, он не насквозь «системный нейтральный человек над схваткой». Нет, он тоже занимает какую-то ролевую позицию, но как системный мыслитель он делает это осознанно (занятие позиции в СМД-методологии называют «самоопределением», мы это назвали бы личным стратегированием) и надёжно удерживает эту позицию. Системный мыслитель самоопределяется/стратегирует свою работу по роли. Он в том числе планирует ресурсы на то, чтобы проявлять в этой позиции (то есть в устойчиво и долговременно удерживаемой роли) мастерство. Он собран после занятия позиции, открыто предъявляет её для удобства организации с ним сотрудничества, открыто предъявляет свои интересы как зависящие от занятой позиции.

Отдельная проблема — это переход к практически непрерывному личному стратегированию (самоопределению), ибо пожизненный однократный выбор позиции::роль для работ какой-то профессии::метод сегодня невозможен, и даже не удаётся это сделать считанное число раз, переопределяться/стратегировать приходится постоянно, постоянно доучиваться новому прикладному мастерству, постоянно усиливать интеллект, то есть заниматься образованием.

Ещё одна проблема — это удержание множественного самоопределения/удерживание множественности стратегий, ибо ролевых позиций в самых разных жизненных контекстах, самых разных жизненных ситуациях приходится занимать много. Что делать с самоопределением/личным стратегированием в современности, а не в прошлом веке «наследственных профессий», подробно рассказывается в разделах по стратегированию руководств по методологии и по системному менеджменту, ибо в сущности своей стратегирование (определение метода/способа достижения каких-то результатов как основа планирования работ) имеет одну и ту же природу для индивидуальных и коллективных агентов, то есть для людей и организаций. Непрерывное в ходе жизни и множественное стратегирование/самоопределение с регулярной сменой методов работы, в которых имеется мастерство, ролевых позиций и интересов для успешной работы по этим методам там называется **бесконечным развитием,** оно необходимо для соответствия агента окружению непрерывно эволюционирующей современности.

Заявить о том, что вы заняли позицию — это ещё не значит оказаться мастером в заявляемом методе работы. Можно занять позицию саксофониста, но очень плохо играть на саксофоне, не знать нот. Поэтому аккуратней с занятием позиций по итогам стратегирования, не забывайте, что для занятия позиции надо сначала научиться в этой позиции действовать, научиться работать не абы как, а по какому-то методу. «--Эй, на барже, вам муханничка не нужно?-— Не нужно!-— Ну и правильно, а то б я вам намуханничал!».

Лидерство

Чтобы люди::агенты устойчиво находились в требуемых от них ролевых позициях в организации, существует метод **лидерства/**leadership: он помогает людям-исполнителям занять устойчивые ролевые позиций в проекте (в том числе, если надо, поощряет доучивание этих людей-исполнителей до нужной степени мастерства в выполнении метода работ по роли), а также помогает сотрудничеству

их друг с другом в этих занятых позициях. Методы лидерства главным образом уговаривают, «убалтывают» агента на занятие позиции, помощь в обучении для достижения мастерства редко рассматривается в составе лидерства, чаще предполагается, что «агент может, но не хочет, надо ему помочь захотеть». Если агент «хочет, но не может» — это другая ситуация, ситуация обучения (но чтобы агент начал учиться какому-то мастерству, тоже требуется лидерство, ибо учиться агент обычно «может, но не хочет, надо ему помочь захотеть»).

В менеджменте как инженерии организации лидерство эквивалентно методам производства, то есть методам изготовления/воплощения целевой системы по известному проекту. На производстве мы покупаем или изготавливаем конструктивные части, а затем берём их и соединяем их так, чтобы они вместе выполняли нужные функции в момент эксплуатации. В лидерстве всё не так просто, начиная с того, что «известного орг-проекта» часто нет, его надо сделать явным. Затем откуда-то надо получить людей. Это делает администрация, но она часто может взять каких-то людей без нужного уровня мастерства, ибо с нужным уровнем мастерства это «слишком дорого, мы не потянем», так что надо людей доучить, а иногда и образовать (усилить интеллект, ибо нет не просто людей, но ещё и нет образованных людей и с этим все согласны. Скажем, усилить интеллект путём стажировки в мастерской инженеров-менеджеров, но это где-то год (480 часов по паре часов в день на пятидневке по программе рабочего развития).

И вот дальше надо «уболтать» людей занять позицию, научить выполнять свою роль и быть собранным, не отвлекаться. Уболтать надо на занятие не должности, не позиции в штатном расписании, не вакансии, а на занятие ролевой позиции!

В лидерстве не занимаются тем, что люди берут себе полномочия по распоряжению своим и чужим трудом и имуществом/оборудованием/капиталом компании и затем используют эти полномочия. В лидерстве занимаются именно проблемой агентов, которые не горят желанием играть по роли, даже если они имеют нужное мастерство. Часто так и говорят, «да, я хорошо умею организовывать мероприятия, но очень хочу научиться работать с 3D-моделями в САПР», дальше организация из хорошего менеджера мероприятий получает плохого инженера-проектировщика.

Лидерство часто называют «катализацией сотрудничества» именно потому, что разделение труда — это разделение прежде всего деятельности по разным ролям, раздача разных ролей разным исполнителям ролей (не все роли отдаются одному исполнителю), и если какая-то роль в её исполнении пропущена, то проект/пьеса/организация труда не идёт, сотрудничества не получается. Нужно, чтобы были исполнители всех нужных ролей в проекте, и они хорошо исполняли эти роли, не сачковали. Например, если никто не играет роль Офелии, а собралось четыре крайне занятых актёра, играющих Принца Гамлета в одном коллективе, то никакого сотрудничества в части роли Офелии нет, его нужно организовывать специально — либо нанимать дополнительных людей на роль Офелии (или уже умеющих играть эту роль, или придётся нанять и затем научить), либо переучивать кого-то из исполнителей роли Принца Гамлета на исполнение новой для них роли Офелии, при этом организация на это время останется без нужного числа Принцев Гамлетов. Поскольку все текущие исполнители ролей заняты, они выполняют работы этих ролей и у них просто нет свободного времени, которое можно было бы инвестировать в обучение новым методам их работы, то редко переучивают старых исполнителей роли. Кто-то ведь должен делать работу исполнителей старых ролей, пока идёт обучение новых людей на новые роли, вот исполнители старых ролей и

будут продолжать работать с полной занятостью по старым ролям, времени на переобучение у них не будет. Новую работу по новым методам работы в новых ролях будут учиться делать вновь набираемые люди.

Если люди устойчиво занимают какую-то культурно-обусловленную позицию, профессионализируются в её методах работы, преследуют интересы этой позиции, то их жизнь наполняется смыслом, ибо они участвуют в культуре, а ещё и участвуют в команде и очевидно приносят пользу в рамках разделения труда.

Устойчиво удерживающие свою позицию люди способны очень эффективно играть свою роль в коллективном/проектном разделении труда, то есть распределении выполняемых работ по разным методам по разным людям-агентам, которые имеют высокий уровень мастерства в выполнении тех или иных методов.

Лидер — это же роль! Роль лидера играют отнюдь не только начальники, её играют по случаю все сотрудники. **Лидер не столько «ведёт за собой» или «даёт пример» или «указывает куда идти» (стратег), сколько помогает людям занимать и удерживать ролевые позиции в организации, он режиссёр-постановщик, назначающий людей-актёров (исполнителей) на проектные/организационные/деятельностные/профессиональные роли и помогающий потом им эти роли успешно освоить, удержаться в этих ролях в суете корпоративной жизни.**

Лидерство::метод — это мостик, который стягивает бездушный мир разработки и проектирования организации (знаний, схем, ролевых/функциональных объектов) и живой мир людей-исполнителей-ролей с их личными (а не профессионально-ролевыми) предпочтениями.

Неформально говоря, лидер убалтывает (в руководстве по инженерии личности это будет «обучает», опора на learning sciences) какого-то исполнителя играть в проекте какую-то роль, занять ролевую/профессиональную позицию. При этом лидер не сам определяет, какую! Её он узнаёт из орг-проекта! И если орг-проекта нет, или он есть, но не документирован, то ждите проблем.

Скажем, в спектакле (который ведётся по пьесе, она будет орг-проектом) не хватает Офелии (роль!), а из наличных актёров в труппе остался только Пётр Николаевич. И Петру Николаевичу совсем не улыбается играть Офелию. Лидер может провести с Петром Николаевичем ряд бесед: рассказать о том, что актёрское мастерство — это искусство перевоплощения, что нужно приобретать новое мастерство (непрерывное образование, бесконечное развитие), про сложность перевоплощения мужчины в женщину и поэтому ровно это будет тестом актёрского мастерства, про древние традиции театра Кабуки[226], где потомственные актёры-мужчины играют роли одновременно как мужчин, так и женщин. И вот уже Пётр Николаевич вышел как-то вечером из дома в юбке, чтобы попробовать, признал, что актёрски это неимоверно трудно, и это «настоящий тест его мастерства», как и говорил лидер, а через месяц он уже с огромным успехом играет Офелию. Труппа счастлива, Пётр Николаевич счастлив, зритель доволен. Это и есть лидерство: с одной стороны катализирование сотрудничества и осмысление жизни людей («официальное» современное понимание лидерства), с другой стороны — убалтывание людей хорошо играть нужные для организации (определяемые орг-проектом) роли (неформальное современное понимание лидерства).

---

[226] https://ru.wikipedia.org/wiki/Кабуки

Лидерством как методом «убалтывания агентов на надлежащее исполнение роли» занимается обычно весь коллектив, и каждого исполнителя той или иной роли в проекте направляют на устойчивое занятие его позиции буквально все члены дружного коллектива, поэтому каждый в коллективе играет в том числе и свою профессиональную/проектную роль, и роль лидера. Лидерство по факту распределённое, а не личное.

Дружные коллективы отличаются ровно этим «распределённым лидерством», ибо никакой один исполнитель-руководитель-заодно-и-лидер не сможет провести всей необходимой лидерской работы в большом коллективе — у него просто не хватит времени на душевные разговоры с сотнями и даже тысячами сотрудников (со-трудников, совместно трудящихся), если только один этот человек-исполнитель будет этим лидерством заниматься. Грубо говоря, лидерство в хорошей команде проекта есть, а явных лидеров обычно нет — каждый занимается немножко лидерством и по отношению к другим, и (главное!) **осознанным лидерством по отношению к себе, демонстрируя ролевое мастерство.** Если лидерство в игровой/театральной метафоре — это метод, которым работает мастер игры/режиссёра, то ролевое мастерство — это то, чему учат игроков/актёров. И, конечно, всех игроков/актёров учат быть немножко и мастерами игры/режиссёрами (а в жизни — менеджерами).

Про лидерство рассказывается в руководстве по инженерии личности, а также ещё более подробно-— в руководстве по системному менеджменту.

Моделирование: роли коллег по организации

Опросите пятерых ваших коллег, заполните табличку, после этого заполните строку для себя.

Номер по порядку	Актёр-человек (ФИО)	Что делает (методы работы в порядке убывания затрачиваемого времени)	Список выполняемых ролей для этих методов работы	Должность	Какие из роли хочет выполнять (включая те, которые не выполняет сейчас)	Какие текущие роли не хочет выполнять	Ваш прогноз его карьеры (в свободной форме)
1.							
2.							
3.							
4.							
5.							
Вы.							

Внешние и внутренние роли

Условно можно разделить функциональные/проектные/stakeholder роли на **внешние** и **внутренние** по отношению к команде инженеров, непосредственно создающих систему (с командами создателей по всему графу создателей сложнее, там каждый раз надо разбираться, какой именно из проектов имеется в виду).

Внутренние роли — это роли **команды проекта** (классические инженеры или не слишком классические — врачи, преподаватели, разные специализации менеджеров, маркетологи и т.п.), а внешние — это все остальные, которые в команду проекта не входят, но на которых влияет или которые влияют на проект. Типичная внешняя роль-— «пользователь», **которого лучше бы называть по его роли в его предметной области** — игрок, пациент, едок, а ещё плательщик, подрядчик, инспектор и т.п.

Хороший анализ видов внешних ролей в крупных продажах (например, инженерного оборудования — в отличие от розничной продажи игрушечной машинки) дан в книге Neil Rackham «Стратегия работы с клиентами в больших продажах»[227]. В этой уже очень старой (1989) книге говорится, что в крупной организации за простым словом «клиент» могут скрываться самые разные роли — и со всеми ними нужно работать по-разному. Так что «нашими клиентами являются поликлиники» уже больше тридцати лет лучше не говорить, это не лучшая практика. В реальной жизни внутри этой поликлиники обнаруживается много разных ролей — и врач, и медсестра, и менеджер, и айтишник (да ещё несколько разных!), и лаборант, и пациент, и невидимый обычно инвестор-владелец. Когда вы говорили «нашими клиентами являются поликлиники», то кого из них вы имели в виду? Для каждого из них нужно уметь отвечать на разные вопросы, подавать материал на разном уровне детальности, хвалить систему за разное, по-разному отстраиваться от конкурентов, по-разному вести переговоры на разных стадиях продажи. Поэтому **пытайтесь называть внешние роли не по отношению к команде (тут вырвется заведомо сверхобщее слово «клиент», как «все эти которые снаружи»), а как они называются у себя там в своих деятельностях, в своих командах.**

Мы рекомендуем табуировать некоторые слова, которые очень любят использовать в подобных случаях, и которые скрывают за собой целые букеты ролей с самыми разными ролевыми предпочтениями в самых разных важных характеристиках систем:

- **Пользователь** — называйте так, как они называют себя в проектных ролях их проектов. Если речь идёт об игре, то это «игрок» (хотя профи-игроделы могут и «игрока» разбивать на составляющие более узкие роли). Если корпоративный софт, то там обычно много самых разных людей, играющих самые разные роли — и для этих разных ролей придётся учитывать самые разные важные им характеристики, их нельзя объединять словом «пользователь». Ключевой ориентир тут — роль называется по ключевому методу работы, которым занимается исполнитель роли. Если играет в компьютерную игру — игрок, если врачует — врач, если думает о финансах — финансист. Простая проверка: есть ли в названии какая-то предметная область, специализация метода? Не просто «пользователь», ибо

[227] http://www.koob.ru/rekhem_nick/strategiya_raboti_s_klientami_v_bolshih_prodazhah

для «пользователя» предметная область непонятна, метод для «пользователя» не прикладной из какой-то предметной области, а просто «метод вообще», это слишком абстрактно для продуктивных рассуждений! Заземляйте! Должны звучать слова из какой-то предметной области, а не общеметодологические слова!

- **Покупатель** — обычно там тоже отдельно бенефициар (тот, кто получает выгоды от купленного), приобретатель (юридическая сторона договора), плательщик, доверенное лицо (например, курьер, получающий покупку), эксперт (тот, кто выбирает товар) и т.д. И их всех нужно учитывать в разных ролях, учитывать их разный ролевой интерес к самым разным важным характеристикам.
- … вы уже поняли идею. Слово «заказчик» или «клиент» также является подозрительным — часто там есть люди с противоположными предпочтениями. Финансист-заказчик хочет минимальную цену и поэтому минимальные возможности, инженер-заказчик хочет максимальные возможности и поэтому максимальную цену — и они там имеют противоположные предпочтения в цене и возможностях, в случае объединения их в «заказчика» мало что с этим можно будет сделать, это всё учитывать в проекте нужно отдельно.

Часто внешние люди-в-роли/стейкхолдеры недоступны. Например, у вас 10 тысяч игроков вашей игры на телефоне, у всех них одна роль «игрок». Но пока игровая программа ещё не написана и в эту игру никто не играет, то и игроков нет. Помним: если какой-нибудь «Вася Пупкин»::оргзвено/«конструктивный объект» не играет хоббита::роль/«функциональный объект», то никакого «хоббита» в этот момент в физическом мире нет! В таких случаях, когда «игроков роли нет, спросить некого, они будут когда-нибудь», внешние роли поручают представлять членам команды.

Четверть века назад для этого использовался **метод персон**, описанный в книге Алан Купера «Психбольница в руках пациентов»[228] в 1999 году. Там моделировались не роли, а исполнители ролей — персонажи/персоны (persona). В этом методе предлагалось составить типовой портрет *пользователя* продукта, и кто-то из команды должен был играть его или её роль, как в театре. Например, «мать-одиночка, 32 года, живущая на окраине небольшого городка, пользующаяся своим планшетом для ведения домашних финансов». Но в последние годы прошла волна критики метода персон, ибо фокус его был направлен не на моделирование роли и методов работы роли, чтобы выявить интересы роли, но на моделирование несущественных характеристик агента-исполнителя роли, которые слабо связаны с сутью дела, то есть с предметами интереса персоны-в-роли. Это типичная ошибка: работа с агентом, а не выявление метода работы агента, его роли, интересов роли.

Это как если бы мы советовали представить в методе персон Принца Гамлета, предлагая точнее описать его вес, рост, пищевые привычки, предпочтения в одежде (тем самым описывая не роль, а агента, который мог бы играть Принца Гамлета) и надеясь при этом, что это даст нам более точный ответ о его деятельностных предпочтениях в моменты, когда он задаёт свой ролевой вопрос «быть или не быть». Понятно, что это психологически удобно команде. Крайне важно, чтобы команда проекта разрабатывала систему не как удобную «для себя», ибо это будет конфликт интересов-— им будет удобна та система, которую легче разработать, а

[228] http://www.koob.ru/cooper_a/psih_v_rukah_pacientov

как удобную «для других». И разрабатывать систему для «толстого Гамлета» много лучше, чем для «меня, хотя я ни разу не Гамлет, но готов его изобразить, уж как понимаю». А вот с точки зрения моделирования ролевых интересов персоны-— это тупик, хотя для успеха системы нужно удовлетворить именно ролевые интересы!

Лучше бы абстрагироваться от того, что «мать-одиночка живёт на окраине небольшого городка» и вместо «пользователя» назвать её «домашним финансистом» — и это отсечёт в обсуждениях все области интереса матери-одиночки, все области интереса жителя небольшого городка, и это правильно, они будут только лишними, отвлекать внимание от важного. Если нужно учесть какие-то особенности исполнителя/актёра, то их учитывают как особенности другой роли этого актёра, а не самого агента-актёра. Так, для плохо видящих (важная характеристика: accessibility, доступность интерфейса) нужно предусмотреть крупный шрифт в интерфейсе (предпочтение). Но при этом мы понимаем, что роли «домашний финансист» и «плохо видящий» имеют разные важные характеристики, и эти характеристики будут отражены в функционале и конструкции системы по-разному.

Может даже быть выявлен конфликт интересов (разные предпочтения по одному предмету интереса/важной характеристике у одного исполнителя роли). Домашний финансист хочет увидеть сразу много на маленьком экране, а слабовидящий — крупный шрифт, и пусть с этим шрифтом на экране поместится не всё, листать будет неудобно, зато всё видно! Какой в итоге будет размер шрифта — как для слабовидящего, или как для домашнего финансиста? Кто победит в этом конфликте интересов? Теперь можно предлагать разные решения. Например, экранную лупу для шрифта (на экране всего много, но лупа даёт возможность посмотреть что-то из этого в нормальном для глаз размере), регулируемый размер шрифта (каждый исполнитель роли домашнего финансиста и плохо видящего настроит себе размер шрифта по-своему) и т.д.

Все современные методы моделирования внешних ролей в проекте (например, метод Jobs To Be Done[229]) пытаются одновременно как поднять точность содержательного моделирования роли и метода работы этой роли, чтобы точнее определить ролевые предметы интересов и предпочтения (нужен успех, поэтому удовлетворить эти предпочтения-— важно!), так и поднять психологическую достоверность этого представления для команды проекта, ибо «помочь человеку» психологически легче, чем «удовлетворить тому, что описывается моделью». Для этого в команду на личные встречи приглашают различных экспертов по предметной области, устраивают фокус-группы для потенциальных исполнителей нужной внешней роли, члены команды сами пробуют набрать необходимый опыт и сыграть за внешнюю роль.

Методы моделирования и «личного представления живыми людьми» разных внешних ролей в проекте обсуждают при разработке **концепции использования** системы (описание функциональности такой системы, которая приносит пользу в ходе её эксплуатации). Разработка концепции использования системы ведётся по методам разработки, которые задействуют разработчики::роль. Кроме концепции использования разработчики::роль разрабатывают также **концепцию системы**, выполняют проектирование/design с детальностью, достаточной для изготовления, изготавливают систему (задействуя внутреннюю производственную платформу,

[229] https://medium.com/no-flame-no-game/что-такое-jobs-to-be-done-и-job-stories-4c57c1dc84cf

«завод», «конвейер», который строят инженеры этой платформы). Разработчики/developers::роль много ещё чего делают, и их обычно считают главными, кто озабочен удовлетворением интересов внешних проектных ролей, особенно пользовательских ролей (мы говорили уже, что этих ролей может быть множество, и называться они будут не «пользователь», а как-то более специфично для их предметной области).

Тем не менее, удовлетворением интересов внешних ролей будут озабочены не только разработчики, но и все командные роли, участвующие в проекте. Так, архитекторы::роль тоже моделируют внешние проектные роли, ибо они озабочены удовлетворить их интересы в части специфических **архитектурных характеристик**, которые могут не учесть разработчики. Скажем, разработчики серверной системы могут быть озабочены какими-то возможностями продукта, которые требуют слишком большой вычислительной мощности для одного «игрока», задача архитектора-— помнить, что этих игроков будет много. Поэтому архитектор идёт моделировать то, что ему расскажут люди из службы продвижения (ожидается этих «игроков» сто, сто тысяч, или сто миллионов одновременно). И далее архитектор и разработчики будут находиться в продуктивном конфликте, а у разработчиков, скорее всего появится технический (архитектурный) долг, то есть работы, которые они должны сделать, чтобы удовлетворить интересы игроков не только по функциональности, но и интересы игроков («игровой аудитории») по доступности и производительности серверов (эти «доступность» и «производительность» как раз будут архитектурными характеристиками), а представителем «игровой аудитории» (все игроки в целом, разница с одним игроком как разница клиентуры с одним клиентом) будет служба продвижения. Архитектора команда разработчиков может считать внешней проектной ролью, менеджер команды считает разработчиков и архитектора (или целую команду архитекторов) вполне себе членами команды.

Показателен пример одной из команд инженеров, которые вдруг попросили включить в состав команды местного юриста. Мотив: на переговорах агент, играющий роль юриста, вёл себя так, как будто работал против команды инженеров, а не был членом команды! Расчёт команды: после формального включения в члены команды исполнитель роли юриста будет учитывать прежде всего интересы инженеров, а не интересы каких-то там своих «надзоров» или кого ещё он там учитывает, представляя и продвигая интересы не инженеров по отношению к этим «надзорам», а интересы «надзоров», игнорируя интересы инженеров.

«-- Сколько будет дважды два?-— А мы покупаем или продаём?», это всегда надо помнить. В паутине организационных отношений легко запутаться без **понимания (то есть акцент на моделировании, причём модели документированы)** методов работы, ролей, предметов интересов и интересов, кто::агент чьи::роли интересы представляет.

Важны оказываются не столько «внешнесть» или «внутренность», сколько **учёт (то есть акцент на документирование результатов моделирования)** всех найденных предметов интересов и далее выполнение работ по согласованию и удовлетворению интересов на основе этого списка предметов интересов как чеклиста-— и при этом надо помнить, что интересы происходят от культурно-обусловленных ролей, а не конкретных агентов, исполняющих роли «носителей интересов», многие интересы не будут предъявлены публично, но могут появляться

в частных переговорах (и это хорошо, ибо если все говорят со всеми, то это плохая архитектура, плохая модульность проекта, подробней об этом в руководствах по системной инженерии и системному менеджменту).

В любом случае, недостающие и внешние, и внутренние роли всегда нужно как-то отыгрывать в проекте, организовывать для этого каких-то компетентных, имеющих хоть какую-то квалификацию, а не случайных доступных исполнителей таких ролей, иначе успешность системы будет под вопросом. Помним, что успешность системы определяется согласием ролей (что-то про датское королевство понимает «принц Гамлет»), а не согласием исполнителей ролей (про датское королевство Вася Пупкин, которому поручили играть принца Гамлета ничего не знает)!

Если исполнителем роли ценителя музыки выступит глухой человек, то роль он исполнит плохо — и его согласие с какой-то музыкой ничего значить не будет, успеха музыки это гарантировать не будет. Если исполнителем роли инженера-архитектора выступит профи в операционном менеджменте, то его согласие с конструкцией системы административно значит много, а вот в системной инженерии — ничего не значит, ибо согласие нужно квалифицированного в системной инженерии актёра-исполнителя роли. Согласие опытного исполнителя роли операционного менеджера, самозванцем выступающего в роли инженера тут в зачёт не идёт.

Неудачи/неуспех многих и многих проектов объясняются тем, что решения принимают неопытные исполнители профессиональных ролей (зато полномочные принимать решения за любую роль, но это объясняет происходящее, но не оправдывает).

Если кому-то сложно представить абстрактного «Принца Гамлета» (но именно это и нужно делать!), то представляйте хотя бы персону: придумайте типичного исполнителя этой роли. В любом случае, избегайте считать, что все люди-в-ролях похожи на именно вас-в-роли. Все люди-в-ролях уникальны, а опытные исполнители этих ролей имеют большой опыт игры в соответствующей роли, плюс они часто имеют для выполнения своей роли в вашем проекте больше времени, чем вы — если вы, конечно, не профессионал именно в этой роли. Но если вы разработчик, то ваш шанс хорошо исполнить чью-то чужую внешнюю роль близок к нулю («лучшая модель кошки — это другая кошка, желательно та же самая», брать надо кого-то, кто имел похожее мастерство, а не кругозорное представление о мастерстве). Нужно предпринимать специальные усилия, чтобы представлять в проекте внешние роли: проводить фокус-группы, беседовать с позиционерами, изучать литературу и т.д.

О вас можно сказать то же самое: вы в ваших традиционных ролях, в которых у вас есть мастерство, будете иметь больше времени для их выполнения, и у вас больше опыта их отыгрывания, чем у исполнителей других ролей. Если это не так, и в вашем проекте «пироги печёт сапожник, сапоги тачает пирожник» про вас, то проект ваш в опасности, и эта опасность исходит от вас.

Начальник — это не роль, это полномочия по принятию решений. Решение будет принято, но оно будет плохим, если у начальника нет мастерства в методе работ роли. Проверяйте мастерство начальников в выполнении какой-то деятельности, в занятии если не позиции, но хотя бы в мимолётном занятии роли и проявлении мастерства. Задавайте им простые вопросы для проверки (например, «каким методом вы получили этот результат? По какому учебнику я могу повторить

рассуждение, приведшее вас к этому решению?»). Конечно, при этом вам придётся определиться: вы любите больше проект или начальника. Если начальника любите больше, то чёрт с ним, с проектом, заваленным из-за ролевой некомпетентности/недоквалификации агента-начальника (агента, занимающего организационное место начальника).

Организационные места, организации, оргзвенья

Чтобы было понятно, кто распоряжается ресурсами организации (помещениями, оборудованием и в особенности трудом других людей), в организациях вводится структура ответственности: одни люди являются начальниками для других. Но, конечно, правильно тут думать об агентах, а в случае юридического «начальствования»-— «лицах». Например, вы не можете уволить агента (человека) в должности «генеральный директор акционерного общества», даже если являетесь собственником контрольного пакета акций. Вам надо собрать общее собрание, которое переизберёт нового генерального директора. Но вы можете назначить этим самым собранием акционеров на должность «генеральный директор акционерного общества» какую-то свою организацию. Генеральный директор в акционерном обществе называется «единоличный управляющий орган», но никто не сказал, что это должно быть физическое/человеческое лицо! Так что вы назначаете генеральным директором акционерного общества свою организацию, а потом извещаете сотрудников акционерного общества, что исполнять роль генерального директора от имени вашей организации будет ваш младший уборщик Вася Пупкин-— Вася Пупкин предъявляет сотрудникам АО ваш приказ и начинает выполнять свои служебные обязанности младшего уборщика вашей организации, то есть выполнять работу генерального директора акционерного общества.

Зачем так сложно? Если вам нужно заменить Васю Пупкина какой-нибудь Дашей Миркиной, то это вы делаете в два часа ночи, издав собственный приказ по вашей организации. Приходит затем Даша Миркина в акционерное общество, добирается туда к трём часам ночи-— и объявляет, что им надо выполнять теперь не приказы Васи Пупкина, а её приказы, поскольку теперь она единоличный управляющий орган акционерного общества, она теперь генеральный директор. А Вася Пупкин? Ну, он продолжает работать в вашей организации. Выгода в том, что вы меняете генерального директора акционерного общества за пару минут, а не через многомесячную процедуру созыва общего собрания акционеров. Это абсолютно рабочая схема, когда-то РАО «ЕЭС России» управляло так деятельностью своих региональных энергокомпаний, которые тоже были акционерными обществами, но дочерними и зависимыми. Если кто из директоров не был согласен с мнением управляющей компании, вылетал со своей должности за несколько часов, а не за несколько месяцев, которые занимает организация внеочередного собрания акционеров, даже если у вас 100% акций.

Так что запоминаем: должности-— это не про роли, и даже не про людей. Это организационные места, на которых работают самые разные агенты, исполняя самые разные роли. И это про назначения, увольнения прежде всего, а затем про подчинение в смысле поручения работ. И всё, если не считать, что название должности ничего не стоит, но носящему какое-то название должности человеку оно может быть приятно. Скажем, малое предприятие «Империя», единоличное управляющее лицо-— не генеральный директор, а «император». Затрат ноль, если вы владелец предприятия, но носителю должности «император» это может быть

приятно.

Организация (organization)-— это организованная (то есть с понятными полномочиями и ответственностями по распоряжению трудом и капиталом) группа агентов с находящимися в их распоряжении зданиями, сооружениями, оборудованием, расходными материалами, сырьём, денежными средствами и т.п. Но главное — это распоряжение ресурсами: принципиальная известность всем организованным агентам о том, кто::агент-на-оргместе/должности кого::агент-на-оргместе/должности может о чём-то::работа-по-практике попросить, и эта просьба должна быть выполнена, а отказ приводит к разбирательствам и может служить основанием для увольнения.

В организациях вводятся **организационные места/должности/вакансии** (иногда их называют и позиции в штатном расписании — не путать с ролевыми позициями из предыдущего раздела!). Структура должностей/оргмест определяет не ролевую структуру, а структуру подчинения, подотчётности, возможности кого-то о чём-то просить, кому-то что-то заплатить, кому-то что-то помешать сделать, кому-то что-то разрешить сделать. В театре это актёры, они ответственны за игру в каких-то ролях, которые будут им назначены потом. Сначала Василия Пупкина принимают на должность актёра и он тем самым получает полномочия играть роли в спектаклях, а уже потом актёра назначают на роль Принца Гамлета в дневных спектаклях, потом также назначают играть Отелло в вечерних спектаклях (и исполнитель на должности актёра играет две роли), потом Василий Пупкин увольняется, и на его должность (после увольнения пустую — вакантную, оргместо) принимают Петра Сидорова, и эти роли Принца Гамлета и Отелло играет уже Пётр Сидоров.

Люди-в-должности и люди-в-роли-— это про разное. Люди-в-должности-— про подчинение-отчётность, в том числе «обещание выполнять работу» и «актуальное выполнение работы», люди-в-роли-— про метод работ. По должности нельзя обычно понять, каким методом работают люди на этой должности, хотя называться должности могут очень похоже на названия ролей, сразу и не поймёшь, о чём речь. Должность «программист» может оказаться занимаемой Богданом, который исполняет роль технического писателя англоязычной документации, совершенствуется в методах технического писательства и считает, что написание программ на языках программирования для него уже скучно, и что он никогда больше уже не будет работать в роли (а не должности, должность в этом плане его не волнует — она про зарплату и название, а не про содержание работы) программиста.

Сотрудников в организации можно попросить не обо всём, но если просить у сотрудника выполнения какого-то действия по ожидаемой роли в проекте, то просьба будет выполнена. Суть тут в том, что должности и роли не зависят друг от друга: они назначаются отдельно, часто роли по нескольку штук назначаются для исполнения их агентами-на-должностях/организационных местах, а уж затем агент попадает на эту должность и тем самым от него будут ожидать выполнения всех ролей, приписанных этой должности. Если к должности «главный альпинист» привязаны роли «спелеолог», «химик-аналитик», «лётчик вертолёта», то замена Павла Андреевича на Илью Никаноровича на этой должности означает, что Илья Никанорович начинает выполнять работы по методам спелеологии, химического анализа и полётов на вертолёте (и от него, соответственно, ожидается мастерство выполнения этих методов). При этом абсолютно очевидно (из структуры

подчинения), кто будет начальником «главного альпиниста», и тем самым-— начальником того агента, который будет занимать должность/оргместо.

Нужно чётко разделять:

- должности-— это про просьбы и долженствование (отсюда «должность», но часто и «место») в их исполнении работ по привязанным к должностям ролям. Про работы ролей (связанные с затратами ресурсов работы с экземплярами каких-то объектов, для чего нужны полномочия), а не сами методы/функции/способы/культуры работы (для чего нужно мастерство).
- роли-— про методы работы прежде всего, профессиональную компетентность/квалификацию/«степень мастерства» выполнения работ по методу этой роли агентом, который будет играть роль (займёт должность, к которой привязана роль), ролевые/трудовые/деятельностные важные характеристики каких-то затрагиваемых методами работы роли систем (предметы интереса) и предпочтения в значениях этих важных характеристик (интересы).

С людьми всё сложно, если мы говорим о том, что должности-— это прежде всего про ресурсы, то есть ещё и понятие «ставка», что означает нормативную длительность оплаченной работы на этой должности за период (чаще всего за месяц, но иногда это неделя или даже день, или даже час-— говорят тогда о недельной ставке, дневной ставке или почасовой ставке, скажем, $300 в час может быть почасовой ставкой преподавателя в зарубежном вузе). «Работать на полторы ставки»-— это обычно тратить в полтора раза больше времени, чем обычно на оговариваемой должности, получать в полтора раза больше оплату. Можно, конечно, занимать несколько должностей сразу («сидеть на двух стульях сразу»-— помним, что должность и оргместо-— это синонимы).

Заполненная человеком вакансия/должность/оргместо даёт нам **оргзвено/организационную единицу**. Минимальное оргзвено/организационная единица состоит из одного человека, причём если речь идёт о «полставки» (а в вузах для того, чтобы дать какие-то формальные полномочия по чтению лекций, но не платить зарплаты, дают иногда 1/8 ставки), то это не «полчеловека», а «один полный человек, но в плане выполнения работ можно рассчитывать на половину обычной продолжительности работы в неделю, по всем приписанным должности ролям».

Оргзвенья могут объединяться, формируя как **подразделения** (постоянно действующие организационные единицы) так и **коллегиальные органы** (временные, собирающиеся лишь время от времени советы, комиссии, рабочие группы), так и **проектные группы**, занятые получением какого-то результата.

Обычно коллегиальных органов и проектных групп в штатных расписаниях (списках вакансий) нет, как раз чтобы подчеркнуть их относительную автономность в подчинении, а также нестандартный способ выплат вознаграждения — хотя это совершенно не означает, что никто не вправе поручать им какие-то дела, требуя выполнения каких-то деятельностных ролей и ожидая от них проявления ролевых интересов/предпочтений, а также что за работу в составе такого оргзвена не будет выплачено вознаграждение. Но повторим: отношения подчинения/руководства отдельно, ролевое/предметное/профессиональное отношение к деятельности — отдельно.

Минимальное оргзвено состоит из исполнителя, рассматриваемого одновременно

как

- **агент-в-должности** (и это означает, что понятна его подотчётность/подчинение и права по руководству/распоряжению трудом и имуществом других оргзвеньев, а также как производится оплата его труда). **По этой линии работает должностное подчинение, management, начальствование. Это линия конструктивного устройства организации.**
- **агент-в-роли** (и это означает, что понятно, каков его метод работы, есть ожидания по поводу его профессионального мастерства для работы по этому методу, а также понятны его предметы интереса и сами интересы, и тем самым можно строить ожидания, каковы будут его действия по роли). По этой линии работает профессиональный/функциональный надзор, governance (это переводится как «поднадзорность», «присмотр», но не «управление») и не принято говорить о подчинении. Но если по этой линии следование методу будет выполняться криво, то агент вылетит из должности-— и никакой начальник ему не поможет.

И тут хорошо бы понимать, что язык этого разговора про организацию (когда понятно, кто кому может поручить что-то сделать и почему эти поручения будут выполняться) не очень применим для разговора на уровне сообществ, обществ и человечества в целом: в сообществах непонятно, кто кому что может поручить, поручения (если они есть) выполняются сугубо добровольно. В обществах может существовать организация (государство), которая своих граждан считает разве что не собственностью, выполнение поручений там обычно связано с рэкетом: прямая угроза насилия, если будет отказ в выполнении поручений. Если поручено прийти в военкомат, а вы не хотите-— то вариантов у вас «уволиться» нет. Если поручено отдать 13% дохода, то вариантов не исполнить поручение нет, и уволиться тоже не так просто, «заплати налоги и спи спокойно» неотличимо в этом плане от рэкета, при неподчинении к вам применят силу, чего не будет в бизнесе или сообществах.

Поэтому будьте аккуратней в использовании наставлений нашего руководства на этих уровнях: сообщества и общества. Системное мышление, конечно, на этих уровнях вполне применимо, но не факт, что вы сходу будете успешны в этом применении. Системные мыслители ещё не так хорошо проработали эти уровни в части методов беглого мышления о них и надёжно работающих инженерных методов создания и развития обществ и сообществ, чтобы можно было быстро и эффективно научить вас работать с сообществами и обществами. Ну, и надо всегда помнить, что если вы хотите что-то с сообществами и обществами сделать, то каждый там человек (и AI-агент) вовсе не мечтает, чтобы вы им поуправляли, и наоборот-— с удовольствием поуправлял бы вами, если бы у него была такая возможность. Поэтому такие проекты чаще всего утопичны. Да, некоторые люди становятся президентами и премьер-министрами и как-то влияют на жизнь миллионов людей, но прикиньте процент этих небожителей от общего населения Земли и посчитайте вероятность, что вы в своей жизни попадёте в это число. Не занимайтесь утопическими проектами, не тратьте на них время. Вы вполне можете улучшить жизнь миллионов и миллионов людей, никак не имея их в непосредственном подчинении-— ни должностном, ни по линии государственной власти.

Начальники: это джокеры, ожидайте от них исполнения любых ролей

Особенно часто путают деятельностные/проектные/организационные роли и оргзвенья/заполненные организационные места при взгляде на начальников — потому как возможности начальников по распоряжению ресурсами очень важны. Вы будете согласовывать какие-то предпочтения по предметам вашего интереса с игроком по профессиональной роли (содержательное обсуждение) или начальником (разрешит-не разрешит, исходя из непонятно каких соображений)?

Начальников по отношению к ролям нужно рассматривать как карточных «джокеров»[230], которые могут стать любой картой по желанию игрока. Начальник пытается играть лично те роли, которые он считает недостаточно представленными в проекте, или пытается выяснить ситуацию, чтобы поручить решение каких-то вопросов тем оргзвеньям (отдельным людям или коллективам), которые в команде есть и которые имеют достаточную степень мастерства в исполнении работ по методу, нужному для решения трудного вопроса. Для этого он активно ищет неучтённые ещё интересы/предпочтения, а также занимается лидерством, то есть ставит в нужную роль одного человека-в-должности/актёра как минимальное по размеру оргзвено, или даже на эту нужную роль назначает оргзвено из нескольких человек как соорганизованных актёров с их распределением оргролей внутри команды-оргзвена. Это лидерство агент-начальник делает, чтобы был качественно выполнен какой-то определённый метод работы, ведущий к учёту каких-то определённых интересов. За речью начальников нужно следить особо внимательно: их предметы интереса обычно не определены, не предъявлены, плюс они их регулярно меняют в ходе разговора.

Помним: «слова-термины важны, и не важны», поэтому не нужно смотреть на названия должностей и титулы. Нужно внимательно слушать прямо сейчас произносимое начальником и/или внимательно читать начальственные тексты в чатах и документах, не обращая особого внимания на упоминание должностей (например, в приказах и распоряжениях), похожих на названия ролей. Должность — это не про содержание работы, то есть методы/стили/культуры работы! Это про полномочия распоряжения ресурсами, поручение работ! Это «логистика», менеджмент, рациональное использование чьих-то ресурсов. Содержание/методы работы нужно восстанавливать самостоятельно из речей, текстов, моделей агентов, участвующих в проекте.

Например,

- первые пять минут «селекторного совещания» какой-нибудь «начальник цеха» будет в роли операционного менеджера (предпочтения: уменьшить сроки работ, уменьшить использование ресурсов),
- потом пару минут инженером (например, инженером-разработчиком концепции использования: предпочтение выявить и согласовать дополнительные сценарии использования, позволяющие предложить новые фичи/возможности целевой системы),
- потом до конца разговора инженером по станкам с ЧПУ (предпочтение низкой сложности обработки деталей со стороны инженера производственной платформы, поэтому намерение выяснить особенности обработки деталей в связи с изменившимися вариантами использования станка и ставшими

[230] https://ru.wikipedia.org/wiki/Джокер

известными новыми сроками).

При разговоре нужно вовремя распознавать момент переключения с роли на роль и отвечать каждый раз правильной роли, а не «начальнику цеха». «Начальнику цеха» отвечать надо только тогда, когда вопрос будет про полномочия распоряжения ресурсами. Только тогда важно, что речь идёт о «начальнике оргзвена/цеха», но и тут легко перепутать операционного менеджера, который вычисляет оптимальное распределение ресурсов, хотя и не трогает полномочия по распоряжению ими (скажем, операционный менеджер выдаёт расписание работ, а конкретных людей для этих работ выдают начальники этих людей) — и поэтому легко перепутать один разговор про ресурсы с другим разговором, «выделение ресурсов» с «наделением полномочиями по поводу распоряжения ресурсами».

Директор театра в пьесах сам не играет, а если даже и играет в ситуациях острой необходимости (кто-то из актёров заболел, или в отпуске, или бюджет не позволяет нанять актёра и надо играть «хоть кому-то»), то **нельзя понять по наименованию его должности, какую роль он играет, в каком методе работы у него высокая степень мастерства.** Директор, как и любой начальник, может вмешаться в ход исполнения любой роли, а также сам сыграть любую роль, в любой момент, просто потому как у него есть для этого полномочия (но не мастерство в этой роли!). Директор театра — не Принц Гамлет, это просто должность в штатном расписании, и даже не актёрская.

Менеджер-начальник и «операционный менеджер»/«менеджер проекта» в классическом проектном управлении — их не нужно путать, хотя слово используется одно и то же, manager/«управляющий». Менеджер-начальник — это указание на ответственность и полномочия (должность), а не на деятельностную/профессиональную роль с требуемым от исполнителя каким-то уровнем мастерства. Операционный менеджер — это человек, выполняющий работы методов операционного управления, то есть увеличивающий скорость прохождения работ через организацию.

Должности — это не роли, это **организационные места**. Агенты как исполнители ролей занимают организационные места, становясь при этом «оргзвеном». К оргместу/должности приписаны:

- имя этого места (имя «должности»)
- полномочия по распоряжению трудом и ресурсами
- обязанность выполнять работы по каким-то методам в каких-то ролях
- обязанность выполнения распоряжений их начальников из вышестоящих оргзвеньев.
- Ставка (какая-то часть вознаграждения за работу, получаемая в силу занятия должности).

Агенты, занимая оргместа, наследуют эти полномочия и обязанности.

Если исполнитель роли ещё и начальник, то он произвольным образом может выбирать свою роль — пытаясь собой заместить отсутствующего исполнителя роли. Конечно, начальник играет эту роль (как и любой нетренированный и непрофессиональный исполнитель роли) часто плохо — занимающий оргместо начальника исполнитель роли инженера-теплотехника может быть очень плохим инженером-теплотехником, или очень хорошим инженером-теплотехником, но вместо того, чтобы потратить на проблему два дня, как это он сделал бы, если

реально выполнял роль, он потратит на проблему десять минут, то есть это как актёр в пьесе Шекспира потратил бы на роль Гамлета минуту-другу, а потом пошёл заниматься другими делами. Но этот агент-начальник может считать, что решения любой подчинённой ему роли должен принимать именно он, ведь он начальник!

Такому положению дел способствует организационная культура «инженеров как аналитиков, готовящих предложения» и «начальников, утверждающих предложения». Один из главных конструкторов машиностроительного завода (обратите внимание, слово «главный» к конструктору безошибочно указывает не на роль, а на должность) сообщил недавно автору руководства, что вечером он обнаруживает в своей информационной системе примерно двести документов на утверждение. На вопрос о качестве принятия решений по утверждению всех этих инженерных «предложений» (двести за вечер!) он только рассмеялся. Такая система будет существовать годами, если не начинать разбираться с агентами (людьми, AI, организациями), должностями, оргзвеньями как агентами-в-должности, ролями (и в руководстве по системному менеджменту будут ещё рассмотрены **оргвозможности/capabilities** как отражение того, что оргзвенья имеют полномочия и ресурсы на выполнение работ по какому-то методу. Ибо можно иметь мастерство работы по методу, занимать должность, но не иметь возможность работать по этому методу. Приняли на работу пианиста::«агент с мастерством игры на рояле» на должность «пианист», поручили играть сонаты в холле отеля, он же умеет это делать. Но маленький нюанс: рояля не выдали. Оргзвено есть, мастерство есть, а вот оргвозможности играть сонаты-— нет, в холле тишина, «нет оргвозможности». Ситуация ничем не отличается от конструктора, которому не выдали средств автоматизации проектирования.

Плохо путать лиц (в том числе людей) как занимающих оргместа и тем самым находящихся на должности и играющих организационные роли уж как могут («не стреляйте в пианиста, он играет как умеет») с «профи», которые заведомо имеют высокую степень мастерства в выполнении работ по методам ролей. В случае обсуждения ролей у исполнителя роли обсуждается профессионализм и «степень мастерства»/квалификация/компетентность в методе работы этой роли, а для занимающих должность — совершенно другие характеристики (ответственность, лояльность, честность, не склонность к воровству ресурсов).

Слово «менеджер» путает всех: это может быть:

- должность «менеджер», понимаемая как «просто начальник», хотя бывает и «менеджер по продажам», представляющий наоборот, нижнюю ступеньку в иерархии работников,
- роль «менеджер» в значении «организатор», понимаемая как совокупная роль орг-проектировщика (проектируют ролевую структуру в работах предприятия и предлагают организационную структуру руководства/подотчётности), лидера (помогающего людям занять организационные места/должности и приступить к качественному выполнению закреплённых за этими должностями ролей — стать организационными звеньями),
- роль «менеджер» в значении только одного узкого «операционного менеджера», которая занимается деятельностью по максимизации логистической производительности организационной системы, «оператор организации», «инженер по эксплуатации организации». У такого менеджера есть чёткий предмет интереса: поток материалов, информации, работ через

рабочие места организации, и интерес в максимизации прохода («проход/throughput» — объем потока за единицу времени), теория/дисциплина метода операционного менеджмента — «исследование операций» (математические расчёты производительности), рассуждения по этой теории/дисциплине дают (в данном случае — операционному) менеджеру::роль возможность оптимизации загрузки работой имеющихся ресурсов (буквально, он::роль следит, чтобы не было «пробок» — заторов в потоке полуфабрикатов через предприятие). И если в организации найдена поломка (проток через организацию нарушен, «ничего не работает»), нужно звать менеджера-организатора::роль. Да, человек::агент может быть один и тот же, и даже роль одна и та же «менеджер», поэтому звать может и не надо, но виды ролей разные, методы работы разные, и их хорошо бы различать. А в крупных организациях так точно и люди будут разные (причём роль операционного менеджера будет быстро отходить компьютеру, эксплуатация чего бы то ни было сегодня интенсивно автоматизируется).

- Модные самые разные должности и роли, которые подразумевают принятие хоть каких-то решений — как административных (должности), так и содержательных (роли). Например, product manager[231], который принимает решения по рыночным свойствам продукта и очерёдности запуска новых фич в разработку (и часто путается с project/program manager, product owner, и даже с stakeholder representative). Это некоторая сложная роль (подразумевает выполнение многих методов работы), но как обычно, эта роль может выполняться одним человеком-исполнителем, многими людьми, и даже многими оргзвеньями/подразделениями. В каждом конкретном случае вам придётся разбираться, с чем именно вы столкнулись: и работать дальше всегда с методами, по которым работает набор ролей агента-на-должности, хотя и учитывая идущие от должности полномочия.
- Основатель предприятия/founder, но по факту он тоже «инженер предприятия» и поэтому эту роль считают менеджерской. Обучение всяких «технологических предпринимателей» организуют главным образом как интенсив по организационному менеджменту, то есть учат основателей предприятий agile-методам (например, канбан) в части операционного менеджмента, но дополнительно ещё обучают стратегированию как согласованию самых разных предпочтений самых разных ролей при выполнении роли бизнесмена, предмет интереса которого-— рыночная цена самого предприятия. Из специфики тут можно указать только на ведение переговоров с внешними проектными ролями для самого предприятия, то есть с инвесторами и контролирующими органами (регистрационная палата, банк).

В наших руководствах термин «менеджер» используется главным образом как инженер организации (чаще всего-— предприятия, но может быть и ассоциация, и холдинг, и даже кружок плавания). «Менеджер» в наших руководствах — это не должность начальника/руководителя, а профессиональная роль инженера организации, которая ещё и бьётся на многочисленные составляющие роли (как и роль любого инженера), каждая из которых работает своим каким-то методом! И есть отдельное руководство по системному менеджменту, где об этом рассказывается подробней,

[231] https://en.wikipedia.org/wiki/Product_manager

даются основные методы менеджмента как инженерии организации и указываются типовые подроли менеджмента, «роли менеджеров».

Моделирование: роли начальника

Запишите для вашего начальника роли, которые вы знаете, что он выполнял в последние несколько месяцев (по возможности, упорядочите их по проценту времени их выполнения от общего времени работы начальника). Не забывайте о таких, например, ролях, как «стратег» (то есть не забывайте о работах по методам стратегирования), «лидер», «организатор» и прочих менеджерских и предпринимательских ролях как ролях инженеров организации (инженеры-технари в отличие от инженеров предприятия/enterprise engineers часто не считают это работой, но это не так — это работа, она требует мастерства и немалых затрат времени и других ресурсов). Агента и должность достаточно вписать в первой строчке таблицы. Если решите, что у вас несколько начальников, продолжайте таблицу, указывая новые ФИО и должности.

Агент (ФИО)	Должность	Роль	Что именно делается в роли	Степень мастерства актёра в этой роли	Процент времени нахождения в роли

Звание и степень мастерства

Ничего нельзя сказать про трудовые интересы начальника: они блуждают, в меру понимания агентом-в-должности своих полномочий. Аналогично ничего нельзя сказать про то, какие трудовые интересы у агента (чаще всего человека) как носителя звания. Слова «профессор», «кандидат наук», «полковник» или «краснознамённая дивизия» ничего не говорят нам, кроме того, что у человека или дивизии есть какой-то опыт в непонятно какой деятельности. Цех повышенной культуры производства, как и краснознамённый полк: нельзя сказать о методах работы (чем занимается) цех или полк, какие они выполняют роли в этих методах, предметная область остаётся неизвестной (Генетика? Артиллерия? Авиаперевозки? Подавление восстаний?). Звание «народный артист» не проясняет, идёт ли речь об исполнении роли Гамлета или Петрушки в совершенно разных спектаклях. Это какая-то общая характеристика агента-актёра, но не уточнение предметной специфики самой роли, не указание на метод работы роли.

А вот **«степень мастерства»**/квалификация/компетентность исполнителя какой-то роли важна. Мы специально говорим про степень, а не уровень мастерства, чтобы не путать с самыми разными другими уровнями (онтологическими в иерархии типов/мета-моделей, системными в иерархии размеров систем с учётом их вложенностей). **Уровни-— это обычно про иерархии, степень-— про проявленность какого-то свойства.** Хотя язык не так строго следует этим различениями, например, «уровень громкости», хотя никаких иерархий по громкости нет, или «уровень воды», там тоже без иерархии. Но всё-таки мы будем говорить про степень мастерства, а не уровень, понимая, что в жизни это будет синонимами.

Сегодня квалификация рассматривается не как набор каких-то отдельных знаний,

навыков и умений, а как набор **компетентностей**, достаточных для игры в какой-то роли. Компетентность означает, что какие-то дела исполнитель роли с этой компетентностью, может успешно выполнить не только «на экзамене», но и в реальной жизни, в реальном проекте. Компетентность в каком-то методе включает в себя не только знания-умения-навыки, но и личное отношение к этому методу, принятие на себя профессиональной ролевой ответственности, самоопределение в роли для этого метода.

По факту, традиционно использующиеся в российской и советской педагогике критерии знаний-умений-навыков/**ЗУН** не включают в себя навыков постановки задач в нетепличных условиях реальной жизни, а компетентность включает в себя постановку задач в реальной жизни для их последующего решения — нахождение в жизни предметов интереса роли, мастером метода для которой является агент, умение удерживать роль среди мельтешения и обилия неважных для игры по роли деталей в жизни. **Мастерство**-— это часть личности. **Личность** -— это вся сумма мастерства агента, который кроме личности («мастерство в каких-то методах работы») имеет ещё и организм («аппаратуру для личности как набора мастерства») и даже можно рассматривать инструменты, усиливающие возможности личности в части каких-то методов (скажем, блокнот как усилитель памяти для мастерства мышления).

Личность проявляет себя поведением расширенного инструментами организма агента. Конечно, мастерство «выполнения какого-то метода»/«игры какой-то роли по сценарию/алгоритму метода» в частности и личность в целом как сумма всех таких мастерств-— это функциональные объекты, которые представляют собой реализуемых конструктивом агента (мозгом, компьютером и телом человека или робота в целом, плюс дополнительный инструментарий метода — агента рассматриваем с его инструментами) создателей/constructor, действующих по алгоритмам методов своей работы как «алгоритмам преобразований» (помним, что constructor theory даёт нам теоретическое обобщение универсальных компьютеров/вычислителей информационных результатов по каким-то алгоритмам до универсальных преобразователей физического окружения до каких-то материальных результатов по каким-то алгоритмам). Подробней про личность и её мастерство, обучение мастерству (развитие личности) и оценку мастерства будет в руководстве по инженерии личности.

Актёр, имеющий мастерство степени знания, умения и навыков (ЗУНы) для роли Принца Гамлета вполне сможет играть Принца Гамлета на репетициях, и максимум-— на выпускном спектакле со строго отобранными зрителями. Мы можем только гадать, как поведёт он себя на каких-нибудь гастролях в сельской глубинке, причём необязательно в своей родной стране. Компетентный в исполнении роли Принца Гамлета актёр сможет сыграть Принца Гамлета где-нибудь в густой толпе посреди суматохи (обычные условия в проектах!), а когда нужно будет общаться с черепом Йорика, за неимением черепа приспособит к этому какую-нибудь картофелину, или хотя бы бумажку с нарисованным черепом — и изготовит этот «череп» себе сам, придумает, как это сделать. Но самое главное — имеющий мастерство достаточной степени найдёт себе в толпе остальных недостающих участников пьесы и сумеет организовать с ними свои ролевые диалоги, договорит их всех вокруг себя!

Он умеет вписать свой труд в роли в труд всего играющего пьесу/ролевую игру коллектива (мы называем такую квалификацию/компетентность «практик»), а в

пределе-— организовать коллектив на выполнение новых для него дел. Мы называем такую квалификацию «мастер», хотя это и узкое значение термина «мастер». Более широкое значение-— «носитель мастерства». Больше значений слова «мастер» в наших руководствах будет описано в руководстве по инженерии личности.

Мастерство включает в себя некоторый прикладной/узкий интеллект для предметной области мастерства. Мы отмечали уже, что фундаментальный интеллект не зависит от предметной области, но в достаточно широких предметных областях число возможных объектов внимания и выполнимых с ними действий достаточно велико, чтобы для выполнения метода у агента-в-роли была общая способность разбираться с ситуацией, «узкий, но интеллект». Мастерство в какой-то широкой области обычно включает в себя способность справиться с задачами в предметной области какого-то метода, с которыми ещё не встречался ни мастер как носитель мастерства, ни его учитель. Грубо говоря, если в пьесе/игре возникнет ситуация, где потребуется дать реплику, которой не было в сценарии для роли (в проекте-— выполнить операцию, которой не было в регламенте/руководстве/стандарте), то мастер выдаст эту реплику (в проекте-— выполнит операцию, которой его не учили), даже если придётся для этого поменять вид мастерства (если у вашего автомобиля заглох мотор или поломался его электропривод, то мастерство собственно водителя вам не поможет, надо сообразить переключиться на мастерство механика или вызвать какого-то агента, у которого есть такое мастерство). В прикладном мастерстве интеллект (и прикладной/узкий, и фундаментальный/универсальный) играет огромную роль, ибо надо всегда быть начеку: в жизни при выполнении метода регулярно что-то идёт «не так», как в знаниях/теории/объяснениях этого метода, и нужно быстро соображать, что же в этом случае делать.

Дальше можно вообще уходить от педагогического сленга (как старого со знаниями-умениями-навыками, так и нового «компетентностного») и говорить просто о мастерстве выполнения метода, мастерстве игры какой-то роли, мастерстве на занятой (профессиональной) позиции. Начальник, начавший (часто незаметно для себя) играть какую-то командную деятельностную/профессиональную/функциональную роль (например, инженера-архитектора, ибо он заметил, что эту роль никто не выполняет, а уже есть несколько групп разработчиков, работу которых нужно срочно наладить в части организации интерфейсов между автономными модулями создаваемой системы) может быть или мастером в этом деле, или хорошо обученным новичком, или вообще не разбираться в системной архитектуре. Важно понимать не только какую роль человек играет, но и какая степень его мастерства в выполнении метода этой роли. Ибо если мастерства нет, то в проекте придётся найти того, кто таки умеет выполнять работы по методу и поэтому сыграет эту роль качественно, иначе не будет успешного проекта. Нельзя искать «исполнителя нужной роли», надо искать «умеющего работать по нужному методу, владеющего высокой степенью мастерства работы по нужному методу»-— затем его назначать исполнителем роли, да ещё и задействовать лидерство::метод для того, чтобы назначенный исполнитель роли реально занимался работами по методу и сотрудничал в этом с остальными исполнителями ролей. Вам надо получить результаты, это достигается исполнением работ по каким-то методам, а не абы какими работами. **Вам не надо просто иметь «исполнителя роли», надо, чтобы он умел работать по методу работы этой роли!**

Вопросы о методах и подборе людей-исполнителей ролей для этих методов — это разные вопросы! Сначала нужно сказать, что «нам нужные точные измерения, эти методы практикуют в метрологии», затем «тем самым нам нужен метролог», а потом только подбирать людей на исполнение этой роли по степени их мастерства в выполнении точных измерений. Так, «степень мастерства»/квалификация/компетентность Маши в точных измерениях, то есть как метролога — нулевая, Петя никогда не выполнял работ по методам метрологии, но прошёл курс метрологии в институте пять лет назад (то есть у него есть небольшое ненулевое мастерство в метрологии), а вот Аристарх Никанорович защитил месяц назад докторскую диссертацию по метрологии — у него мастерство в метрологии выше, чем требуется для выполнения работы метролога в текущем проекте.

Сколько всего проектных ролей

Нужно запомнить простой принцип: **важных ролей в проекте всегда на одну больше, чем вы выявили**. Роли с их интересами по отношению к вашей системе и проекту всегда есть, если есть проект. И вы их не «разрабатываете/изобретаете/выдумываете». Вы их «выявляете», «находите» (discover).

Если брать отдельно внешние, а не командные/внутренние роли, то начинать нужно не с двух-трёх ролей, а примерно с **15 (пятнадцати).** Помним при этом, что если пятеро человек по поводу проекта играют одну и ту же деятельностную роль, то это всё-таки одна роль. Помните «Танец маленьких лебедей» из Лебединого озера? Там четыре исполнителя, но культурно-обусловленная роль «маленький лебедь» по факту одна, хотя работ по этой роли в спектакле будет четыре. Игроков у смартфонной игры может быть сорок тысяч человек, но роль у них всех одна — «игрок». Так что 15 внешних ролей по факту могут оказаться исполняемыми довольно большим числом людей. Но верно и обратное: один агент может играть множество ролей, так что пять человек в проекте могут оказаться на поверку десятком самых разных ролей. Агент-игрок может быть ещё и плательщиком за право играть, и покупателем (регистрация покупки идёт на него)-— хотя все эти роли могут играть и разные люди, например, если играет ребёнок, регистрация покупки идёт на маму, а платит со своей карточки за игру папа.

Один из менеджеров проекта нам рассказал, что после того, как он легко выявил первые пятнадцать внешних ролей, он понял, на что незаметно уходило всё его время: **15 телефонных разговоров с исполнителями этих ролей в день по 10 минут каждый и 10 минут подготовки к разговору и обработки результатов разговора сразу дают 5 часов просто на поддержку адекватного понимания текущей ситуации согласования их интересов! А если нужно решать какие-то проблемы проекта с этими ролями, то 10 минут разговора явно не хватает.** Поскольку внешние роли никто в проекте явно не отслеживал, это время уходило «невидимо», оно тратилось неосознанно, в планах оно не отражалось, ресурсы менеджера на эту работу не выделялись и не учитывались. Осознанность по отношению к ролям и их интересам, планирование работы с агентами-в-ролях по поводу согласования их интересов — они важны!

Так для танцевания::метод/культура/практика/процесс/деятельность (мы табуируем слово «танец», поскольку это то ли «вид танца»/«стиль танцевания», то ли «танцевальный перформанс/выход/исполнение танца», то ли «танцевальная

культура») можно отдельно выделить роли[232]:

- Танцора (который, собственно, танцует — выполняет «танцевание»),
- подроль «партнёра» (только в тех видах/стилях танцев, где они есть! В других видах/стилях танца их может не быть, или наоборот, может быть целый ансамбль/команда/crew со множеством танцоров, а для стиля lap dance это не столько «партнёр», сколько «клиент». В социальных/парных стилях отдельно есть виды ролей leader и follower, причём гендерно-независимые[233]),
- зрителя (но только в танцах, где предполагается зритель. Например, в социальной кизомбе зритель не предполагается, партнёры в паре танцуют только друг для друга),
- хореографа (отвечает за композицию и набор движений),
- тренера/учителя танцев (учит танцевать),
- музыкального редактора/диджея (подбор музыки),
- организатора танцевального мероприятия (вечеринки, баттла/соревнования, концерта, семинара/фестиваля и т.п.),
- часто в этот список включают фотографа (на вечеринках) или видеографа (для концертных выступлений и баттлов),
- для сценических стилей танца (балет, эстрадные танцы) будет ещё художник по костюмам,
- нередко и гримёр/визажист.

И это тоже не полный список! Например, в спортивных танцах есть ещё

- судьи в жюри,
- член счётной комиссии (обсчитывает результаты работы судей),
- судья-информатор (объявляет результаты работы судей).

Конечно, это всё только роли, а исполнителей ролей может быть меньше. Если я танцую-импровизирую сам перед зеркалом дома, то множество из этих ролей выполняю я сам, одновременно, или не в один момент времени. Например, сначала я буду музыкальным редактором/диджеем, выбирая музыку, а потом танцором и зрителем, по ходу дела выполняя также работу хореографа — сочиняя сам танцевальный выход/перформанс/постановку. Или агентов-исполнителей ролей может быть больше: зрителей много, и даже можно **углублять разделение труда**, то есть расщеплять текущий набор ролей выполнения какой-то роли на подроли и раздавать эти подроли разным исполнителям: музыкальный редактор/диджей разделяется на художественного руководителя мероприятия (задаёт формат музыки, её стилевую направленность и проверяет следование им у диджея, слово «руководитель» тут не должно смущать, оно тут не относится к названию должности) и самого диджея, непосредственно реализующего заданный музыкальный формат и стилевую направленность.

Когда системный мыслитель думает о какой-то системе, о связанных с этой системой проектах по её созданию и использованию, он помнит, что успешность системы зависит именно от того, как учтены интересы/предпочтения для всех участвующих ролей. И ещё он помнит, что сам он::актёр не нейтральный «мыслитель над схваткой», а тоже выполняет в проекте какую-то роль, у него::роли есть

---

[232] https://vk.com/@buffdance-rasskaz-o-sistemnyh-urovnyah-v-tance

[233] https://jenabl.wordpress.com/2017/05/17/inclusivity-and-dance-terminology/

предпочтения/интересы, которые он хочет реализовать. Не забывайте учесть себя как роль, а также не забывайте показать другим агентам-участникам проекта, какую «позицию» вы заняли!

Часто из многочисленных синонимов проектной/трудовой/практической/организационной/инженерной/культурной/стратегической роли (особенно внешней проектной роли) выбирают слово «стейкхолдер» (stakeholder), привычное для англоязычной литературы. При этом в англоязычной литературе ещё могут отдельно рассматривать агента-в-роли как stakeholder и отдельно признавать наличие самой роли этого человека как stakeholder role, но это различение принято отнюдь не везде! Например, ISO 42010:2022 сообщает: «stakeholder: role, position, individual, organization, or classes thereof, having an interest, right, share, or claim, in an entity of interest. EXAMPLE: End users, operators, acquirers, owners, suppliers, architects, developers, builders, maintainers, regulators, taxpayers, certifying agencies, and markets». Как вы считаете, сколько человек тут прочтут, что это примеры ролей, а сколько-— что это агенты-в-ролях, учитывая данное в определении перечисление «роли», «должности»/position и агента/individual с организацией (или даже их классами). Сколько смутятся «рынком» в качестве примера-— он тут роль, должность, индивид, организация или их класс? С точки зрения нашего руководства по системному мышлению это перечисление аналогично «кресты металлические, кресты католические». Конечно, мы используем в нашем руководстве материалы международных стандартов, но призываем относиться к ним с осторожностью и обязательно включать при их чтении «машинку типов», отлавливать онтологический дребезг.

Мы признаём, что понятие «стейкхолдер» (и много реже «стейкхолдерская роль») распространены уже в русском языке, но избегаем использовать в наших руководствах. Слово «стейкхолдер» легко путает роль и актёра/исполнителя роли, и даже должность. «Стейкхолдер Принц Гамлет» и «стейкхолдер Вася Пупкин» для русскоязычного уха звучат одинаково, и с этого момента мышление про роль незаметно выключается, и остаётся привычное бытовое мышление «про людей» и даже не как актёров, а как агентов, которые не только играют трудовые роли как личности-актёры, но ещё как организмы едят, спят, болеют и ходят в туалет.

«Стейкхолдеры» начинают восприниматься как «просто люди, как они есть и почему-то иногда организации». При таком словоупотреблении легко посчитать, что принимают решение эти «просто люди», например, из-за вредности, плохого характера, по настроению, а не из-за каких-то ролевых предпочтений. Нет, роли не могут вредничать, они же роли, а не люди! Их интересы-— это ролевые интересы, их решения следуют из метода роли. Если инженер-прочнист считает, что деталь в системе нельзя делать для выбранного материала тоньше 4 см, а все остальные этому сопротивляются, потому как такое решение наступает на их интересы, то это не его мнение как агента-с-вредным-характером, а мнение роли!

Если путаются агенты, роли, должности, то-— методологического (про методы работы/деятельность, чем занимаются роли) мышления нет, а вследствие этого нет и системного мышления! Поэтому мы рекомендуем использовать терминологию с ролями, ибо «трудовая/проектная роль Принц Гамлет/инженер/продавец» хорошо звучит, а вот «внешняя проектная роль Вася Пупкин» — язык не поворачивается это выговорить. И это хорошо, что язык не поворачивается! Если не путать агентов, их роли и должности, то будет полноценное методологическое мышление, вы

сможете обсуждать культурно-обусловленные методы работы, ролевые предметы интереса, ролевые интересы, задействовать системное мышление-— и поначалу договариваться со всеми (квалификация «практик»), а затем и договаривать всех вокруг вас (квалификация «мастер»).

Первые версии нашего руководства использовали традиционную терминологию «стейкхолдера», и материал про роли плохо понимался. После изменения терминологии на «проектная роль» в её отличии от человека или организации дело наладилось, в том числе стало проще общаться с людьми в проектах, которые не были знакомы с нашей мета-мета-моделью, не осваивали наших руководств. Заодно исчезла шутка про то, какой степени прожарки стейк у стейкхолдера. **Не говорите про стейкхолдеров, говорите про роли (в крайнем случае — «стейкхолдерские роли», если у вас в организации без слова «стейкхолдер» не могут прожить)!** В текущей версии руководства ещё и «человек» был заменён на «агента», ибо роли в 21 веке не всегда выполняются людьми, а в коллективе людей не все решения принимают люди, появилась и умная вменяемая «нежить»/AI (и нельзя говорить, что это только «интерфейсный модуль к человеку», так и про секретаря-человека можно сказать, что это «интерфейсный модуль к человеку-начальнику».

Если говорить не про «роль в методе работы»/«проектную роль», а просто про практика/трудящегося/деятеля/инженера — то это вполне возможно, но вполне может быть путаница примерно та же, что и со стейкхолдером:

- то ли это человека::агента называем «практиком» или «деятелем» безотносительно его роли,
- то ли указываем в разговоре на роль агента, хотя и не специфицируем, какую именно.

В таких случаях лучше переспрашивать. У нас в тексте руководства несколько раз был приведён подобный ряд синонимов, обычно мы указываем при этом на актёра-в-какой-то роли. Но огромная вероятность, что ролевой аспект будет не считан, и это просто будет «эпитетом» для конкретного агента: «практик Вася» будет означать вовсе не то, что Вася исполняет работы каким-то конкретным предметно-специфичным методом (например, расклеивает листовки на заборах и он сейчас не Вася, а «расклейщик листовок», то есть «Вася::актёр практикует метод расклейки листовок», «Вася::актёр/практик расклейки листовок»). Для незнакомых с понятием метода работы и практикующей этот метод роли это будет просто указанием на самого Васю, типа «господин» или «мистер». Хорошо же звучит: «практик Вася», «деятель Вася».

К сожалению, лёгких решений и всем понятных терминов не будет, все слова давно заняты. Поэтому лучше говорить о ролях, это надёжней, но мы будем говорить и «деятель», и «практик», и «актёр», и «инженер», и даже «трудящийся», имея в виду актёра-в-роли, исполняющего работу по какому-то методу, работающему в какой-то культуре, в каком-то стиле, выполняющего практику/деятельность/«вид труда»/«вид инженерии»/«способ работы» или даже стратегию (как выбранный в ходе стратегирования::метод целевой метод работы). Всё это синонимы с небольшими нюансами.

Ошибки в определении ролей

«Если на клетке слона прочтёшь надпись «буйвол», не верь глазам своим»[234]. Этот афоризм Козьмы Пруткова полностью применим к проектным ролям: мы должны выявлять их по словам и действиям, а не ориентироваться на официальные титулы должностей. Иногда титулы, конечно, совпадают с проектными ролями. Но часто — не совпадают. Агент с должностью «программист», например, может оказаться техническим писателем пользовательских инструкций, вовсе не уметь программировать. То же самое-— с самоназваниями, от них тоже можно ждать подвоха. Только дела, только обсуждение предметной области какого-то метода работы (деятельности, культуры, практики — синонимов тут огромное количество) в разговоре.

Если Принц Гамлет вдруг начинает спрашивать «Молилась ли ты на ночь, Дездемона?», это уже не Принц Гамлет! Это Василий Пупкин, который переключился на другую роль (какую? В этот момент важен кругозор! Вы можете пропустить момент, когда вам ясно говорят, что сейчас кого-то задушат!).

В момент неожиданного для вас переключения ролей у исполнителя роли/актёра полезно задать вопрос, почему это он::актёр поменял обсуждаемый «предмет интереса»/тему и стал другой ролью: вы можете узнать много интересных подробностей. Скорее всего это означает, что всплыл какой-то новый интерес: актёр вашей команды Вася Пупкин, исполняющий предыдущую роль, припомнил что-то важное, поэтому и переключил роли, тем самым проявил интерес другой роли. Не забывайте задавать вопрос о причине смены темы/«предмета интереса«/«важной обсуждаемой характеристики», когда «люди или AI»::агенты будут вдруг неожиданно менять роли в ходе разговора. Должности этих агентов неважны, разве что они будут свидетельствовать об административных полномочиях участников разговора. Должность важна только при отдаче распоряжений и определении ставки, на мышление и деятельность должность не влияет!

Тем не менее, мы помним, что «встречают по одёжке», и роль чаще всего пытаются вывести из должности. Поэтому мы рекомендуем называться как можно ближе к реальной роли. Назовитесь «аналитиком»-— и с вами будут беседовать как с человеком, который сам ничего не решает, только «делает предложения начальству». Назовитесь «инженером»-— с вами будут беседовать как с человеком, который принимает решения сам, а не только делает предложения по поводу принятия решений кем-то другим.

Напомним основные ошибки, которые делают люди, определяя внешние для проекта и внутренние (командные) роли. Если вузовский студент будет делать эти ошибки (они ведь тут явно прописаны), он смело может рассчитывать на два балла на экзамене. Если эти ошибки будет делать не студент, а инженер-менеджер, то эти два балла поставит ему жизнь:

- Вместо роли указывают исполнителя — конкретного человека или конкретное подразделение (ФИО или название подразделения/оргзвена), или даже AI.
- Вместо роли указывают «ответственного» (должность, оргзвено, позиция в штатном расписании), чаще всего «начальника» (ошибка типа «Директор

[234] https://ru.wikipedia.org/wiki/Козьма_Прутков

театра» в списке действующих лиц «Сна в карнавальную ночь». Легко узнаётся и по квалификаторам «ведущий», «главный», «менеджер»: эти слова в названии роли должны сильно настораживать).

- Указывают звание (учёную степень, воинское звание) вместо роли, попытки указать степень мастерства, но не сам вид мастерства.
- Указывают тип организации, в которой много различных ролей (клиника, завод). Указать для медицинского проекта «поликлиника» — какими методами работы занимается поликлиника, какие у этой «сверхроли» важные характеристики и предпочтения в их значениях? Там внутри много людей, они исполнители самых разных методов работ в самых разных ролях, и там среди них наверняка будут разные предпочтения по одним и тем же характеристикам одной и той же системы, их нельзя замешивать в одно целое! Указывайте роли в больших организациях подробно, чем ближе к узким профессиональным ролям, тем лучше!
- Считают, что один агент-исполнитель роли — это одна роль. Нет, один исполнитель/актёр может исполнить десяток ролей, идёт ли речь о человеке, AI или целой организации. Впрочем, и одну роль может исполнить большое подразделение, или даже многие тысячи человек (скажем, роль «игрок» у игры с двумя миллионами игроков) или сложная сеть из множества AI-агентов.
- Считают, что пять внешних ролей в проекте — этого более чем достаточно. Нет, обычно счёт идёт от пятнадцати. Если выявлено всего пять внешних ролей, то это вы недостаточно подумали.
- Забывают учитывать свои собственные роли и их интересы. Но ведь вы-агент тоже участник проекта! Что вы там делаете, чего добиваетесь (интересы), чему для этого учились, у вас высокая степень мастерства в выполнении работ каким методом?
- Забывают про множество предметов интереса у проектной роли и множество проектных ролей, имеющих один предмет интереса. А вот предпочтения в одной важной характеристике/предмете интереса у этих ролей могут быть разными. И очень разные варианты реализации предпочтений: три актёра/исполнителя роли врача поинтересуются артериальным давлением и даже согласятся с тем, что его нужно срочно понизить, а потом предложат три абсолютно разных способа этого понижения! Поговорка про «два юриста, три мнения»-— как раз про это.
- Не обращают внимания на проявляемый в текущей ситуации интерес, указывают предполагаемый интерес из каких-то прошлых или ожидаемых ситуаций. Нет двух одинаковых проектов, всегда уточняйте, что происходит в вашем проекте, и вот прямо сейчас (ибо ситуация в проекте обычно быстро меняется: люди деятельны!).
- Не различают ролевых предметов интереса, самих интересов/предпочтений, не ожидают неминуемых действий агентов по реализации их намерений в достижении интересов их ролей.

Для того, чтобы разобраться с ролями в проекте, можно рекомендовать заполнять табличку примерно такого содержания. Тут приведён пример заполнения такой таблички для простейшего совещания на троих из отдела разработки:

Номер	Роль	Метод роли	Исполнитель роли	Должность испол	Организация	Мастерство испол	Предмет интереса	Предпочтение/интерес	Стратегия агента

п/п				нителя роли	исполнителя роли	нителя в данной роли			(«что будет делать» ::метод)
1	Разработчик концепции использования	Jobs to be done	Динара	программист	Отдел разработки	новичок	функциональность	Максимизация числа фич	Опрашивание всех, а что не удалось выяснить в опросе — то придумывание
2	Визионер	MVP	Я	программист	Отдел разработки	мастер	функциональность	Минимизация числа фич	Продавливание MVP, остальное игнорируется
3	Операционный менеджер	Оценка времени выполнения работы по функциональным точкам	Тимур	Начальник отдела	Отдел разработки	новичок	Время разработки	Минимизация времени разработки	Обрезание функции даже в MVP

Табличка эта специально составлена так, чтобы у вас появлялись вопросы, которые без этой таблички могли бы быть упущены: чтобы вы не перепутали роль и должность, чтобы подумали про метод работы, чтобы обратили внимание на мастерство актёров в исполнении конкретных ролей. Управление вниманием к ролям, предметам интереса, интересам, методам реализации интересов по ходу совещания — это пример методологического (о методах работы) мышления, маленькой его части. Без этого никакого системного мышления не будет, ибо «система в глазах смотрящего», а смотрящим тут будет роль.

Кого нужно было ещё пригласить на совещание, чтобы полноценно обсудить эти предметы интереса? Для ответа на этот вопрос прежде всего нужно подумать, какие роли могут иметь другие предпочтения в этих важных характеристиках. Затем предположить, кто::агент может быть актёром/исполнителем этих ролей. Например, можно было бы пригласить актёра в роли разработчика, который бы дал оценку времени разработки каждой предлагаемой фичи/feature[235] системы: это помогло бы актёру в роли операционного менеджера разобраться со своим интересом. Можно улучшить результаты работы роли разработчика концепции использования, пригласив кого-то более опытного вместо текущего исполнителя (там ведь у текущего исполнителя мастерство «новичок»). Можно пригласить кого-то из службы продвижения продукта (маркетинга), чтобы лучше оценить MVP[236].

[235] https://en.wikipedia.org/wiki/Software_feature
[236] https://en.wikipedia.org/wiki/Minimum_viable_product

Главное, что эта простая табличка заполняется за три минуты, но она помогает спланироваться и лучше понять ситуацию. В сложных случаях такие таблички экономят месяцы бесплодных разговоров непричастных людей на неважные темы.

Особо в такой табличке нужно уделять внимание своим собственным ролям. Заявляли ли вы на этом совещании свои роли, ролевые интересы? Раскрывали ли свои намерения, планы по реализации ролевых интересов? Знали ли участники совещания, какие роли вы играете в ходе совещания, если вы явно не заявляли эти роли?

Знание должности и из какой вы организации тут совершенно неважно, равно как и какие роли вне совещания вы ещё можете играть. Главное: вы на совещании вели себя как кто (по роли)? Каким методом/способом работы пользовались в своей роли? «Никаким» и «здравым смыслом» — это очень, очень плохой ответ. То есть вы не играли никакой роли, творили произвол, были не в культуре, не демонстрировали стиля, вели себя как «дикарь-на-производстве». Как вы помогли остальным участникам совещания лучше с вами коммуницировать? Скажем, в данном примере, рассказали ли вы явно, что будете пользоваться методом, задействующим понятие minimal viable product? Декларировали ли своё предпочтение в фичах с опорой на метод разработки с MVP и привели ли аргументы «из книжки по MVP» (то есть привели ролевое культурно-обусловленное, а не личное-без-роли мнение)?

Отвечали ли вы в своих выступлениях на интересы/предпочтения остальных присутствующих ролей, демонстрировали ли вы понимание их предметов интереса и учёт их планов по реализации их интересов в ваших планах? Или вы отвечали «просто людям» или должностям, а не актёрам-в-роли?

Вовсе необязательно строить такую табличку в письменном виде каждый раз в полном виде на каждом совещании (помним, что «поговорить в чате»-— это тоже совещание!). Но для начала вам будет более чем полезно «мыслить письмом», в данном случае выполнять моделирование ситуации с ролями с документированием в виде таблички. Сначала это будет трудно и требовать значительного времени на продумывание. Например, у каждого агента может оказаться по три роли, а совещание может быть целого десятка человек. Или возникнут трудности в выявлении роли AI. Как учитывать роль AI? Как роль, которую играет провайдер сервиса этого искусственного интеллекта (роль играет AI, или даже не создатель, а оператор этого AI, а сам AI его инструмент-— типа как «забивало» может играться плотником Васей или молотком плотника Васи), ибо ежели что пойдёт не так, договариваться придётся с провайдером сервиса AI, а не самим AI (с Васей, а не его молотком) — и даже не с ролью «провайдер», а какой-то подролью, ибо «провайдер AI» может легко разбиться на множество самых разных ролей, и договариваться нужно будет только с какой-то одной из них, а все они будут играться разными людьми или даже целыми службами фирмы-провайдера AI.

Думать про роли и их методы будет всё легче и легче, затем вы будете это делать «на лету» и в уме. И, конечно, без какого-то методологического/трудового/деятельностного/практического/инженерного кругозора эту табличку заполнить невозможно, для людей без кругозора происходящее на совещании будет магическим действием по взаимным уговорам агентов-участников. С табличкой же становится совершенно понятно, в чём интрига, и что можно предпринять для улучшения качества принятия решений на

подобных совещаниях.

Моделирование: кто был на последнем совещании

Заполните табличку для последнего совещания (не меньше трёх человек), в котором вы участвовали. Не забудьте проверить, попали ли вы сами в эту табличку!

Номер п/п	Метод роли	Роль	Исполнитель роли	Должность исполнителя роли	Организация исполнителя роли	Мастерство исполнителя в данной роли	Предмет интереса	Предпочтение/интерес	Стратегия агента («что будет делать»: :метод)

Задания: исполнение ролей в жизни

Выполните следующие задания:

1. Попробуйте предварять ваши реплики на совещаниях (и синхронных очно или онлайн, и асинхронных, то есть в чатах и письмах) фразами, обозначающими вашу роль для этой реплики («как финансист, я скажу», «как инженер по испытаниям, я ожидаю»). Напишите пост о результатах такого поведения: что вы ожидали, и что получилось.
2. По итогам всех моделирований текущего раздела напишите пост в порядке мышления письмом.
3. У вас есть какая-то должность и вы играете какую-то роль. Связаны ли они друг с другом? Попробуйте при знакомстве с другими людьми представляться не по должности, а по роли. Напишите пост про то, как это влияет на ваши ощущения от жизни, на восприятие вас собеседниками.

7. Системные уровни

Не всё системы, что ими называют

Все самые разные определения системы сходятся на том, что **система как целое состоит из взаимодействующих частей, которые в своём взаимодействии дают эмерджентность/системный эффект**, т.е. при взаимодействии части системы как целое проявляют новые (появившиеся, emergent) свойства, которых нет у частей системы. Грубо говоря, шестерёнки и пружинки в механических настенных часах имеют одни свойства (например, шестерёнки — точность изготовления, и износостойкость, пружинки — упругость), а вот сами эти часы как целое из частей-шестерёнок-и-пружинок — совершенно другие свойства (например, точность хода, габариты), а вот интерьер дома, как целое из часов, стен, мебели, домашней утвари и украшений — третьи (например, удобство прохода, освещение). Эмерджентность — это как раз то, что «точность хода» emerge при переходе внимания от частей часов к целым часам, а удобство прохода emerge при переходе внимания от часов к интерьеру в целом. Это важно для разделения труда:

точностью изготовления шестерёнок обычно занимаются одни люди (пишём «люди» для простоты, правильно было бы писать «агенты») с высокой степенью мастерства в методе точного изготовления металлических деталей, точностью хода — совершенно другие люди, инженеры-часовщики, а вот удобством прохода в каком-то интерьере — третьи, художники интерьеров.

Нюансы могут различаться, но вот сборка из частей (мы тут говорим не о делении на части, а о сборке, отношение composition), взаимодействие собираемых частей и вытекающая из этого эмерджентность/системный эффект присутствует во всех школах системного мышления. В разных школах системного мышления есть две традиции трактовки отношений части и целого: инженерная, где части могут быть только физическими объектами, и общефилософская, где части могут быть любой природы — и физическими, и ментальными объектами.

Трактовка отношений части и целого в нашем руководстве — инженерная, материальная/физическая, где речь идёт о частях системы как физических объектах в пространстве-времени. Мы тут более-менее придерживаемся положений инженерного онтологического стандарта ISO 15926-2:2003. Часть может быть и ролевым объектом, но только в тот момент, когда роль этого объекта-части играет какой-то конструктивный объект или даже группа конструктивов. Частью может быть место в пространстве — но в физическом пространстве, а не «ментальном»/математическом.

Мы не рекомендуем считать частями системы процессы::поведение, но у самих частей системы и у системы в целом обязательно есть поведение: изменение состояния частей и системы в целом в ходе взаимодействия частей. Поведение шестерёнок в часах — они вращаются и передают движение, поведение пружинок — они сжимаются и распрямляются, запасая и отдавая энергию, поведение часов — они «идут» и показывают время, поведение интерьера — он проявляет удобство для нахождения в нём антропоморфных агентов (людей и роботов) и хорошо освещён как в светлое, так и в тёмное время суток. Эти изменения состояния (сборка системы из частей в ходе создания, какие-то изменения частей в ходе использования) и есть «поведение», так что мы всегда привязываем рассмотрение процессов к физическому миру: процессы представлены участвующими в этих изменениях состояний частями. Если хочется показать процесс «падение яблока», то нужно представить его как яблоко, участок Земли под яблоком и изменение состояния яблока (его относительную скорость относительно участка Земли и расстояние между яблоком и Землёй при вертикальном падении). Всё материально.

В материальном/физическом варианте для абстрактных/идеальных/математических объектов (классов, типов, множеств и т.д.) частей нет, ибо эти части не физичны: для них не существует объёма в физическом мире, который они занимают, поэтому непонятно, что там из чего состоит. Например, нельзя проконтролировать, что все «математические молекулы» как части входят в число «математической клетки» или «математического организма», ибо нельзя трактовать часть как часть объема пространства-времени, занимаемого целым.

Мы отрицаем системное мышление с использованием понятия «часть» для нефизических/математических/ментальных объектов (например, описаний: разговор о частях картин, частях текстов), ибо будет полностью непонятным, что

такое дающее системный эффект «взаимодействие» для частей как нефизических объектов. Чаще всего тут можно услышать о каком-то алгоритме, который производит вычисления, задействующие одновременно несколько нефизических объектов — но сами-то эти объекты не взаимодействуют, разговор перемещается на вычислитель с его алгоритмом и методы его работы! Это конструктивизм в математике, переход на операции создания объектов вместо обсуждения отношений объектов. Мы в нашей версии системного мышления следуем пока не этому ходу на конструктивную математику, но инженерной традиции исключать математические объекты из рассмотрения в качестве полноценных систем.

Тут надо оговориться, что всё-таки идут исследования ровно по этой линии конструктивизма в математике: если включить в рассмотрение не сами ментальные объекты, а компьютеры или их обобщение — создателей/constructors, работающие с описаниями по каким-то алгоритмам, а сами описания выражать как операции по построению этих описаний (конструктивизм в математике), то как-то ещё можно подходить к системному мышлению для ментальных/математических объектах. Но это самый-самый фронтир, в инженерных и менеджерских методах работы для такого подхода пока применений нет, а мы чуть подробней об этом расскажем дальше в руководстве и дадим литературу — для самых любознательных. Пока же оставим: **системы всегда физичны, всегда материальны**.

Эта «физическая» (но и ролевая, и конструктивная, и пространственная, и т.д. — все они базируются в конечном итоге на физичности системы и её частей) трактовка деления системы на части и есть наш вариант системного подхода. Выбор именно этой трактовки делается в силу критерия адекватности мышления: **наше мышление о системах всегда согласовано с физическим миром.**

Наши проекты по созданию и развитию систем всегда как-то изменяют физический мир, они не фантазийны. Рассуждения о системах тем самым заземлены/grounded, то есть все абстракции в конечном итоге имеют основания в ситуациях в физическом мире, это не чистые «игры разума», физически невозможные ситуации отслеживаются и убираются из рассуждений. Если в физическом мире ничего не происходит в результате наших проектов, то мы такими проектами не занимаемся: наша мысль всегда опирается в конечном итоге на физический мир.

Во многих других трактовках системного подхода (например, в классической и порядком устаревшей «общей теории систем»[237]) слово «часть» используется неформально, нестрого, и «целое» собирается из самых разных объектов, в том числе абстрактных и плохо определяемых в части их присутствия в физическом мире: физических предметов (тоже, как у нас!), но также и слов, правил, настроений, намерений — всего чего угодно. **В нашем варианте системного подхода мы не будем считать нефизические/абстрактные объекты системами и частями систем. Описания живут в мире понятий/математическом мире, они не системы.**

Тем самым мы не признаём системами-из-системного-подхода разные «**системы» знаний/правил** — корпуса знаний, наборы правил и даже система уравнений как «набор закономерностей». Система Станиславского, система Монтессори, система Платона, политическая система, система «минус 60» (так называют один из наборов правил для похудения), законодательная система — это всё некоторые абстрактные целые, состоящие из каких-то абстрактных частей-элементов (знаний, правил,

[237] https://ru.wikipedia.org/wiki/Общая_теория_систем

иногда даже подразумеваемых/implicit/неявных/tacit — не выраженных в знаках на каком-то носителе, т.е. не документированных) — все эти «системы» (в кавычках для нашего варианта системного подхода!) не занимают места в пространстве-времени. Это не настоящие системы из системного подхода, они только называются словом «система», которое используется в другом словарном значении. Очень часто люди используют слово «система» просто для того, чтобы указать, что они как-то думали, когда собирали какие-то части этих знаний, как-то согласовывали эти знания, правила, уравнения друг с другом, наводили какую-то структуру. Например, разбивали все правила на отдельные группы правил, что-то включали, что-то не включали — даже и (непространственная!) «граница» есть, отделяющая «включённое» от «не включённого». Но слово-термин «часть» для такого употребления слова «система» в других (не из системного подхода) словарных значениях тут не обозначает физического предмета, она не занимает объём/место в пространстве-времени, сами эти «части» обычно не составляют иерархии по отношению «часть-целое» между физическими предметами.

Не системы — это и алгоритмы, в том числе алгоритмы/знания/теории/дисциплины методов работы»::«обобщённые алгоритмы для constructor», а также «обобщённые алгоритмы для выполняемых создателем работ» (алгоритмы/знания методов, в которые подставлены материальные/физические типы преобразуемого материала и задействование оборудования для работы с этими материалами). Все такие «алгоритмы» — для универсальных (полных по Тьюрингу) вычислителей/компьютеров/computers (физичных, хранящих состояние на материальном носителе!) и универсальных преобразователей-создателей/constructor (которые физичны!), действующих по немного расширенному понятию алгоритма (то есть описанию метода работы), как это описано в constructor theory. Настоящие/true системы тут — это

- универсальные (по Тьюрингу) компьютеры/вычислители/computers (классические, квантовые, нейроморфные, оптические и т.д.), «исполняющие алгоритмы»/«следующие теориям»/«делающие вывод по имеющимся знаниям». Если алгоритмов несколько и разговор идёт о «системах алгоритмов», то это «ненастоящие системы из системного подхода», а просто слово, обозначающее продуманный набор каких-то описаний (примерно так же, как речь идёт о «системах уравнений»).
- универсальные создатели/constructors (станки, роботы, люди, организации из них всех), которые выполняют работу по каким-то алгоритмам/знаниям/объяснениям, описанным в учебниках/регламентах/инструкциях/уложениям по методам этой работы (часто для неодушевлённых конструкторов их называют не методами/практиками/культурами работы, а просто функциями — токарный станок выполняет функцию токарной обработки детали, но вот агент-токарь — следует методам токарной обработки деталей). Если тут говорят о «системе методов работы»::«набор поведений» или «системе описаний»::информация, то это «ненастоящие системы из системного подхода», тут слово «система» используется тоже как отсылка к продуманному набору описаний.

Ещё один класс систем-не-из-системного-подхода в силу их абстрактности (неприсутствия в мире, отсутствия занимаемого места в физическом пространстве-времени) — это **систематики/таксономии**. В систематиках/таксономиях речь

идёт о классификаторах: классах классов (множества множеств), которые в конечном итоге классифицируют в чём-то похожие системы (физические и абстрактные). Это иерархии, которые строятся с использованием двух видов отношений: **классификации** (classification, «подведение под класс», включение элементов множества в множество) и **специализации** (specialization, род-вид, подмножества во множестве). Классификатор Ламарка (система Ламарка) состоит из классов в чём-то похожих животных, универсальный десятичный классификатор (УДК, система десятичной классификации) классифицирует книги, объединяя в своих классах чем-то похожие по содержанию книги, Общероссийский классификатор изделий и конструкторских документов ОК 012-93 (классификатор ЕСКД, единой системы конструкторской документации, которая сама система знаний/правил) — они все не настоящие системы-индивиды, они лишь классификаторы для классов систем (физических, как мы их понимаем в нашем варианте системного подхода) и классов абстрактных объектов. При этом классификаторы удобны тем, что дают какие-то имена для систем, то есть при встрече классификатора полезным бывает при поиске систем смотреть не на сам классификатор, а на классифицируемые им физические объекты. И легко потом разобраться, что классификатор всего лишь описывает объект, у которого может быть много описаний. «Роза пахнет розой, хоть розой назови её, хоть нет»: один и тот же объект может быть легко отклассифицирован разными классификаторами, то есть описан по-разному, иметь разные имена, принадлежать разным классам/множествам/таксонам/типам в зависимости от целей классификации. В любом случае, решить эту проблему можно, только обсуждая систему и отличая её обсуждение от обсуждения классификатора.

«Спор о терминах» бесперспективен, избегайте его. Вопросы «является ли X объектом типа Y» интересны только когда это какой-то классификатор, из результатов классификации по которому будут материальные следствия. Обычно это государством утверждённый классификатор, например классификатор болезней, из которого будет следовать ваша инвалидность по классификатору инвалидности, а из отнесения вас к какой-то группе инвалидности будет следовать получение или неполучение пенсии. От отнесения или неотнесения какого-то подвида тигра к виду тигров, по большому счёту, ничего не зависит. Но если это в законодательстве прописанный вид из Красной Книги, то последствия понятны: штрафы для охотников.

Классификатор видов деятельности (по сути, это классификатор по видам методов работы) может быть важен, если какие-то виды деятельности надо прописать в уставе и несоблюдение влечёт за собой штраф, или вид деятельности лицензируется, и на выходе опять штраф — я думаю, вы поняли идею. Классификаторов, отнесение к которым ведёт к каким-то административным последствиям, довольно много[238]. В любых других случаях систематики интересны для примерного обозначения темы разговора, но могут быть сменены в любой момент, системы можно описывать самыми различными способами, они от смены способа описания сами не меняются: сто разных фотоаппаратов и сто разных художников дадут на выходе 200 изображений Эйфелевой башни, а она останется прежней. Систематики, которые используются для проектирования — это уже интересней, но всё равно в любой момент можно иметь несколько таких систематик.

[238] https://classifikators.ru/

Так что систематики, «системы классификации», таксономии интересны как наводки на то, что они классифицируют какие-то реальные/материальные/физические системы. Как всегда: если при поиске систем вам встретилось «описание», ищите то, что описано — системы могут быть там. Но могут и не быть, проверяйте, описывать можно и описания. Классификаторы как систематики легко могут классифицировать и не-системы, а какие-нибудь описания (другие классификаторы, например).

Так что **не всё системы::«тип из мета-мета-модели в части системного подхода»/«тип из онтики системного подхода», что называют системой::термин/слово/имя. И не всё не-системы::тип, что системой::термин/слово/имя не называют!** Скажем, бытовой холодильник мало кто назовёт «системой охлаждения пищи» (а не бытовой холодильник — запросто), но в системном подходе это явно будет «бытовой холодильник»::система. **Тип «система» какому-то физическому объекту даёт создатель в какой-то роли, которому зачем-то потребовалась эта система и он выделил её своим вниманием из пестроты самых разных физических объектов окружающего мира.**

Вопросы. Что считать системой

Что можно считать «системами» с позиций преподанных вам методов системного мышления? (1/2)

+++ отопительная система ### Да, это отопительную систему можно найти в физическом мире, она состоит из частей — батарей, трубопроводов, вентилей и т.д. Входит как физическая часть в физическое здание или сооружение.

+++ вольтметр ### Да, вольтметр занимает какое-то пространство-время в нашем физическом мире, он состоит из частей — стрелок или цифрового индикатора, корпуса, катушек или микросхем и т.д. (в зависимости от выбранного принципа работы и конструкции). Входит как материальная часть в какое-то материальное оборудование.

+++ система обеспечения безопасности на водах ### Да, это вполне находящаяся в нашем физическом мире система, состоящая из материальных частей — спасательных вышек на пляжах, самих спасателей, их лодок, спасательных кругов и т.д. Входит как часть в пляж::система (пляж тут — физический объект, занимает место в пространстве-времени, система безопасности на водах — часть входящих в него физических объектов).

--- система Монтессори (образование в детском садике и школе) ### Нет, это просто правила организации образования — нет материальной части, только описания того, как должна быть устроена школа (ещё эти описания могут быть документированными, но документы с описаниями как физические объекты — это не «система Монтессори», а «физические носители для документированных описаний»!).

--- система Брайля (шрифт для слепых) ### Нет, это правила письма для слепых — принципы того, как отображать информацию на физическом носителе (но системы Брайля нет в физическом мире, по ней нельзя «постучать», это описание того, как должны быть устроены физические носители).

--- система сдержек и противовесов (политических) ### Нет, это принципы (правила) устройства государственных институтов. Нельзя показать пальцем на те места в пространстве-времени, где находятся части этой системы.

+++ Общество с ограниченной ответственностью «Система» (организация) ### Да, на части этой системы можно указать как находящиеся в пространстве и существующие во времени: оргзвенья/подразделения из вполне материальных людей, компьютеров, зданий, оборудования, запасов сырья и т.д.

+++ кафедра системного менеджмента ### Да, это система, кафедра::организация находится в физическом мире и занимает там место, состоит из взаимодействующих и вполне физичных авторов курсов, преподавателей, их помощников, администраторов, помещений с мебелью и оборудованием (при этом помещение и оборудование тут опциональны, что не отменяет физичности кафедры::организации).

+++ система защиты Земли от астероидной атаки ### Да, это вполне чётко указуемое место в пространстве-времени (нас не волнует, что это место будет существовать только в будущем по отношению к тому актуальному времени, когда мы его рассматриваем), состоящее из ракет, датчиков (телескопов), компьютеров, людей, AI-агентов, организаций и прочих физических объектов.

--- система защиты в шахматах ### Нет, это просто правила выбора ходов, её нет в физическом мире, она абстрактна. Это описание какого-то метода/способа игры в шахматы (а описание метода описывает поведение создателя, то есть это даже не про шахматы, а про шахматиста!).

Что можно считать «системами» с позиций преподанных вам методов системного мышления? (2/2)

+++ система пищеварения ### Да, это система, она состоит из взаимодействующих и вполне физичных (занимающих место в пространстве и времени) желудка, кишок, разнообразных желёз, а сама система пищеварения является частью организма::система::«физический объект».

+++ наночастица ### Да, она явно занимает место в пространстве-времени, имеет физические части (поверхностный слой, внутренние слои) и сама — физическая часть чего-то (например, раствора с наночастицами, или покрытия из наночастиц).

+++ система теплоизоляции ### Да, система теплоизоляции вполне физична, в ней есть физические части: теплоизолятор и его крепление, они занимают пространство-время. Система теплоизоляции сама — часть какого-то оборудования::система или здания::система.

--- единая система кодирования (обозначения) частей машин и оборудования ### Нет, это систематика/классификатор::описание (набор названий для таксонов) и правила присвоения кодов::имя документам и изделиям, самой этой системы в физическом мире нет.

--- система тренировок ай-ки-до ### Нет, это набор правил, по которым проводят тренировки ай-ки-до (описание метода тренировок).

--- система рангов в Counter Strike:Global Offensive ### Нет, это систематика: набор классов (рангов), описание.

+++ Солнечная система ### Да, она физична, то есть занимает место в пространстве-времени, в ней есть физические части (Солнце и планеты, астероиды) и она сама — часть физической Галактики.

+++ программная система моделирования потоков жидкости ### Да, это система, существующая в физическом мире (и меняющая в нём состояния в момент исполнения программы) как часть компьютера. Эта система имеет в качестве своих частей компьютерную память и экран в каких-то своих изменяющихся состояниях, она сама — часть полного программно-аппаратного комплекса моделирования, а этот комплекс, в свою очередь, часть какой-то инженерной фирмы.

--- периодическая система Менделеева ### Нет, это упорядоченный набор типов объектов — систематика::описание, она не занимает место в пространстве-времени, она не физична по своей природе.

--- система моральных ценностей ### Нет, это правила, по которым определяется моральность тех или иных действий. Её нет в нашем пространстве-времени, она не физична, это описание.

Системное разбиение

Системы одновременно являются целым для каких-то частей внутри них (подсистем) и частями для какой-то объемлющей их целой системы (надсистемы). Каждая подсистема тоже является целой для своих уже под-подсистем (относительно начальной системы), а каждая надсистема — часть для над-надсистемы. Тем самым можно говорить об иерархиях **системного разбиения/breakdown/decomposition** на части сверху вниз, или они же иерархии составления (composition) системного целого снизу вверх. Если какую-то систему мы пока не планируем бить дальше на части, то её называют «элемент системы», подчёркивая, что где-то есть целая система, для которой эта подсистема является элементом::«часть, далее не разбивающаяся на части».

Уровни в иерархиях системного разбиения (иерархиях/«графах-деревьях», выстроенных по отношению композиции/«часть-целое»)

называются системными уровнями. Системные уровни выделяются в физическом мире вниманием, для их выделения не надо «разбивать» систему на части! Если вы разобьёте физически на функциональные части-органы, например, живой/функционирующий/работающий организм — он будет после этого мёртвый! Поэтому системное разбиение выполняется только вниманием! Ничего разбирать-собирать не надо! Но вот переходить вниманием от более крупных частей к более мелким, и обратно — надо. Отслеживать, что в иерархию по отношению «часть-целое» не попало какое-нибудь иное отношение (скажем, отношение классификации или специализации) — надо. Отслеживать, что вы случайно не включили в физическое разбиение на части какой-нибудь ментальный объект — это тоже надо.

Классический пример системного разбиения — это пришедшее из биологии разделение на системные уровни (от мелких к крупным) атомов-молекул-клеток-органов-организмов-популяции-биоценоза-биосферы. При этом в биологии кроме названия «системные уровни» используют названия «уровни организации», «эволюционные уровни», «уровни сложности жизни».

Вы видите, что системы обозначены на схеме системной иерархии кружочками, а **стрелки-ромбики традиционно обозначают отношение состава, где целое — со стороны ромбика на стрелочке**. На рисунке видно, что клетки состоят из молекул, но клетки сами части органов. Органы состоят из клеток, но органы сами части организмов.

Вот это слово «состоят» (composed by) не означает, что вы можете разбить органы на клетки, а потом вновь составить из этих клеток органы. Нет, это «состоят» относится к тому, что вы вашим вниманием выделяете в работающем организме разные органы, в органах вашим вниманием выделяете клетки, но можете пройтись вниманием и в обратном порядке: в работающем организме вниманием объединить клетки в органы, а органы в организм. Ничего не нужно физически разбирать, системные уровни — они только для управления вниманием в ситуациях со множеством объектов, которые состоят из других объектов и сами части ещё каких-то объектов. Системные уровни определяются работой внимания (о которой агенты договариваются в своих проектах, чтобы скоординировать свою работу), они не «объективны», то есть самые разные агенты вдруг будут обнаруживать их одни и те же в самых разных ситуациях, самых разных проектах! С другой стороны, системы::«объекты каких-то системных уровней» выбираются не случайно: в физическом мире системы обычно стабильны, они не так легко поддаются действию энтропии, их не так-то легко разрушить. И кроме всего прочего, системные уровни часто культурно-обусловлены и эта культурная обусловленность как-то ещё и поддерживается языком. То, что «смартфон состоит из экрана, корпуса, камер,

аккумулятора и какой-то электронной начинки» — это уже общекультурное знание, о нём не надо специально договариваться сегодня (а вот во времена создания первых смартфонов — надо было). Но вот что именно будет относиться к корпусу, а что уже к экрану (скажем, защитное стекло экрана — это часть экрана, или это часть корпуса?), что войдёт в состав камер (чип, поддерживающий исключительно работу камер — это часть камер, или «какая-то электронная начинка? а кольца крепления объективов на корпусе — это часть камер, или часть корпуса?) — это уже будет зависеть от решений конкретных команд проектов создания и развития смартфонов даже сегодня.

На схеме не показан системный уровень биоценоза/эко-системы, а потом ещё биосферы (они выше уровня популяций). Есть исследования, которые показывают, что рост уровней сложности неизбежен, то есть неизбежно объединение частей во всё более и более сложно устроенные целые на многих и многих системных уровнях[239].

На схеме также не показаны и уровни атомов и элементарных частиц (они ниже уровня молекул), но это не означает, что их нет: **посылка открытого мира[240], «что не сказано в нашем тексте, это просто не сказано, но это не значит, что этого нет».** Вот так и нужно читать системные диаграммы: любой верхний уровень на них не самый верхний (всегда можно найти какую-то надсистему, в конечном итоге все системы являются частью вселенной или в некоторых физических теориях — частью мультиверса как бесконечного множества вселенных). Молекулы показаны на этой диаграмме элементами (далее неделимыми подсистемами) клеток, но это тоже подсистемы, их неделимость условна, она просто не рассматривается в тех деятельностях, для которых была составлена эта схема. Так что и снизу всегда можно найти какие-то части вроде как «элементов». Это для текущего проекта внизу системной иерархии могут быть далее неделимые «элементы», но в пределе деление идёт до уровня «ниже кварков», то есть тех мельчайших физических объектов, из которых состоят кварки (эти объекты самые разные в разных физических теориях).

Набор системных уровней, на каких рассматривается система, всегда зависит от проекта. Возможно, в другом проекте из этого же разбиения будет взят какой-то другой набор уровней. В одном проекте вы увидите подробное разбиение дома на отдельные стены, крышу, фундамент, коммуникации, а в другом проекте — разбиение нескольких городских кварталов на дома, но вот в домах уже вы не увидите стен, крыш, фундамента, коммуникаций, ибо «элементы» в этом проекте будут — дома. В третьем проекте (скажем, проект девелоперской компании) будет и подробное описание «городских кварталов»::«системный уровень», включая дома, скамейки, отдельные деревья, и «дома»::«системный уровень» будут разбиты на какие-то части. Для девелоперской компании важно не только расположить дома::подсистемы в «городских кварталах»::системы, но и построить их, а для этого спроектировать их устройство из подподсистем (и дальше подсистем этих подподсистем — до систем самого мелкого уровня, на котором ведётся проектирование и изготовление домов, то есть до уровня дощечек, гвоздей, кирпичей, электрических розеток и т.д.. Но вот уже детали внутри электрической розетки в ходе проектирования дома вряд ли будут учтены — но будут интересовать

---

[239] https://www.pnas.org/doi/10.1073/pnas.1807890115

[240] https://en.wikipedia.org/wiki/Open-world_assumption

конструкторское бюро и завод, который выпустит электрические розетки для дома).

В соответствии с нашим вариантом системного подхода мы будем требовать, чтобы системное разбиение было полностью физично/материально: каждый системный уровень выделялся по отношению часть-целое между материальными предметами, то есть занимающими какие-то места/объемы в пространстве-времени. Инженерам-менеджерам нужно запомнить: если в их системном разбиении появляются нематериальные части, то в системном мышлении это как 2*2=5, за такую ошибку студенты сразу получают два балла и отправляются на пересдачу, а в рабочем проекте появляются сомнения о том, что этот инженер-менеджер профпригоден. Мы не можем советовать, чтобы взрослых за такое увольняли с работы, не особо сомневаясь, но очень близки к такому совету.

Внимания должно хватать, чтобы отслеживать физичность отношений часть-целое на несколько уровней вниз и вверх. Если внимания на такое отслеживание не хватает (с учётом того, что это внимание поддержано записями — не нужно всё держать в голове!), то с этим нужно что-то делать, например, вернуться к освоению рациональной работы по соответствующему руководству (честно, выполняя задания). Работать с отсутствующим вниманием нельзя, улетать неосознанно мыслью в нефизический мир или путать что там часть, а что целое — это опасно для проекта.

Многоуровневость системного разбиения принципиальна: на самом верхнем уровне любого такого разбиения потенциально будет вселенная, на самом нижнем уровне — суперструны, но вот эти крайности мало кого интересуют, поэтому системное разбиение делается для какого-то диапазона размеров физических объектов. Для деятельности людей крайне важен очень маленький диапазон, который как-то соразмерен размеру самих людей. Эти не большие и не маленькие в космических масштабах и масштабах микромира системы служат предметом деятельности — чаще всего это от нескольких километров (например, строительные проекты: небоскрёбы, дамбы и мосты) до нескольких нанометров (транзисторы на компьютерных чипах). Вот с предметами/системами в этом небольшом диапазоне размеров чаще всего взаимодействуют люди (с их инструментами! Необязательно руками!). В космофизике и физике микромира есть очень много чего интересного за пределами этих размеров, но людям пока удобней менять мир в этих пределах, и не выходить за них.

Явное указание системного уровня важно для управления вниманием. Формально, дотронувшись до цветка у растения (уровень «органы» в системном разбиении растительных организмов) я дотрагиваюсь и до вселенной (цветок ведь часть вселенной!), и дотрагиваюсь до элементарных частиц (в цветке они точно есть!). Ну, или дотрагиваюсь и до всего растения, и до клеток растения. Непонятно, до чего я дотронулся, что является предметом моего внимания! Поэтому требуется явно выделить ту систему::«объект внимания», который как-то позволит мне определить масштаб «дотрагивания». В нашем примере это будет — «цветок».

Когда я машу рукой, то я машу всеми молекулами руки — но неправильно ведь говорить о молекулах, когда говорим о размахивании рукой? И неправильно «махать телом» (слишком большой объект). Поэтому «машу рукой» — машу системой, находящей на много системных уровней выше, чем молекулы, но всё-таки ниже уровня целого тела.

Системные уровни принципиальны, они отражают саму суть системного подхода — на каждом системном уровне в силу эмерджентности/системного эффекта появляются новые свойства, поэтому чаще всего **обсуждения ведутся отдельно для отдельных системных уровней агентами, которые разбираются со свойствами, проявляемыми системами на этом системном уровне, а также свойствами надсистем, которые получаются из свойств систем текущего уровня. Грубо говоря, клетки обсуждает микробиолог, организм человека в целом — медик, коллектив людей — менеджер. Отдельное подробное обсуждение на каждом системном уровне существенно упрощает обсуждение сложных систем.** Если вы обсуждаете предприятие, то обсуждайте людей, оборудование, запасы материалов, здания и сооружения, но обычно при этом не нужно обсуждать планетарную систему вокруг Солнца, частью которой является предприятие, как часть Земли, или биохимию клеток у людей, как биохимию части предприятия.

Помним, что если вы обсуждаете проблему на слишком низком системном уровне — это ошибка редукционизма, если вы обсуждаете проблему на слишком высоком системном уровне — это ошибка холизма. Правильно удерживать обсуждение на трёх уровнях:

- надсистема в её окружении и её свойства. Например, компьютер в целом, его характеристики (габариты, вес, энергопотребление, производительность по каким-то признанным тестам)
- целевая система как целое и другие системы как целое на том же системном уровне в составе надсистемы. Как совместная работа систем этого уровня влияет на состояние и свойства надсистемы. Например, центральный процессор как целевая система, память, материнская плата, блок питания, вентилятор/охлаждение, корпус.
- подсистемы целевой системы: как их совместная работа (взаимодействие) влияет на состояния и свойства целевой системы. Например, для центрального процессора — ядра с арифметико-логическими устройствами, многоуровневая кэш-память, блоки ввода-вывода. И тут же вам тест на внимание: если вы представите «настоящий» (а не из диаграммы в книжке) центральный процессор, то увидите залитый пластмассой чип (или несколько сразу «чиплетов») с выводами, а уже на чипе какие-то зоны с теми самыми ядрами с арифметико-логическими устройствами, многоуровневая кэш-память, блоки ввода-вывода.

Это легко сдвинуть на уровень вниз: рассматривать

- транзисторы (уровень подсистем) в составе какой-нибудь
- кэш-памяти в центральном процессоре (уровень целевой системы) и как характеристики этой памяти и других IP будут влиять на характеристики
- центрального процесса (надсистемы).

В любом случае, обычно обсуждают «что делаем» (целевая система), «из чего оно состоит функционально, из чего делаем конструктивно» (подсистемы), «в состав чего это войдёт» (надсистема). В любом случае, вам **для обсуждения системы обычно надо не меньше трёх системных уровней, вы будете обсуждать самые разные характеристики объектов на этих уровнях, эти обсуждения будут делать самые разные роли, а на каком системном уровне будет ваше целевая система — это зависит от проекта. Что целевая система для**

одного проекта — надсистема для другого, подсистема для третьего.

Инженеры описание системного разбиения дают обычно через описание типов (например, «самолёт»), но в реальности этим типам объектов соответствуют подводимые под этот тип физические предметы (самолёт с бортовым номером 128, самолёт с бортовым номером 2467 и т.д.). В то же время философы (но не мы в нашем руководстве) часто обсуждают системные уровни с произвольными частями, в том числе абстрактными[241] — и там отношения «часть-целое» определяются очень по-разному.

Мы предпочитаем подход инженеров, а не философов. Если менеджер (инженер предприятия) или политик (социальный инженер) будут думать как философы, перемешивая ментальные и физические объекты в одних и тех же видах иерархий, они могут утерять связь с реальностью, добиваться реализации какой-то чрезвычайно привлекательной утопии. Утопии тем и отличаются, что очень привлекательны, но их беда в том, что они нереализуемы: мир будет меняться, но не к лучшему! Скажем, будет такая «борьба за мир», что камня на камне не останется — итогом желания мира будет ровно война, прямая противоположность!

Утопии (описания нереализуемого физического мира) именно так и устроены: хочется изобилия (коммунизма), а получаются продовольственные карточки, голодная жизнь. Нам же нужно, чтобы наши системы были и привлекательны, и реализуемы в физическом мире с задуманными, а не «уж как получится» характеристиками, мы не просто мечтатели, мы деятели/практики/инженеры. Современное системное мышление устроено так, чтобы при помощи понятий системного подхода удерживать внимание на физическом мире на самых разных уровнях его масштаба/системных уровнях. А ментальные объекты? С ними будем работать именно как с описаниями, в крайнем случае — описаниями описаний, но в конечном итоге — описаниями физического мира.

Это удержание внимания на объектах физического мира стабилизируется (собранность!) при помощи документированных информационных моделей, которые заодно помогают и предсказать характеристики будущих систем. Физические модели (типа как модели самолёта в аэродинамической трубе) сейчас используются всё реже и реже, но вот информационные модели помогают удерживать внимание на объектах физического мира, и готовить детальные и согласованные между собой описания этих объектов. Подробней о том, как увязаны между собой математические и физические объекты (изучаемые математикой и физикой соответственно), а также как мы работаем с выражаемыми знаками моделями этих объектов (семантика) будет рассказано в руководстве по интеллект-стеку.

Моделируемые/удерживаемые во внимании на разных «уровнях размера»/«системных уровнях»/«уровнях состава/разбиения» взаимодействующие физические системы, которые создаются и развиваются другими физическими системами-создателями — это и есть предмет системного мышления.

В силу разделения труда разные роли, участвующие в проекте, для одной и той же целевой системы свои предметы интересов могут иметь у систем на разных уровнях — подсистем или надсистем. «Предметы интереса»/«важные

[241] http://en.wikipedia.org/wiki/Holon_%28philosophy%29

характеристики» эмерджентны, поэтому можно ожидать, что они существуют на каких-то уровнях и отсутствуют на других — но в этом-то и предмет многоуровневого рассмотрения, что все они зависят друг от друга, даже если находятся на разных системных уровнях. Точности хода часов нет на уровне шестерёнок, там есть точность изготовления. Но вот от точности изготовления шестерёнок зависит точность хода механических часов. И поэтому у часовщика появляется интерес в получении как можно более точно изготовленных шестерёнок. Изготовитель шестерёнок наоборот, заинтересован в минимальной точности за получаемую цену (нет проблем поднять точность, но и цена тогда вырастет — до неприемлемой для часовщика). Часовщик и изготовитель шестерёнок ведут переговоры, чтобы достичь согласия.

Никакого «истинного» или «объективного» как независимого от ролевых интересов разбиения системы на части нет, хотя обычно в качестве частей выделяются какие-то более-менее стабильные объекты, но даже эту стабильность разные роли могут оценивать по-разному. **Для одной и той же системы в проекте по созданию системы обычно одновременно рассматривается несколько вариантов разбиений на части. Минимально обычно разбивают на**

- **Функциональные/ролевые части, область интересов «как работает»**
- **конструктивные/материальные части, область интересов «из чего сделано»**
- **пространственные части (места), область интересов «где находится, как скомпоновано»**
- **затратные/стоимостные/cost части (но это уже будет затрагивать разбиение и целевой системы, и системы создания — совокупная стоимость владения), область интересов «во сколько это обойдётся».**

Таких разбиений может быть много больше, и каждое из них может проводиться по-разному, в зависимости от области интересов. Эти разные варианты разбиений на части агенты (люди, AI, организации), играющие разные внутренние и внешние роли в проекте, согласовывают между собой, добиваясь успешности системы.

Даже один агент может использовать несколько разных разбиений, по-разному для удобства мышления и действия разных ролей выделяя части системы — чтобы ему было удобней думать в каждой из своих ролей, чтобы удобней было действовать. Некоторые роли подразумевают одновременное задействование нескольких разбиений. Например, роль разработчика работает с изобретением — когда подбираются аффордансы (конструктивные части) для ролевых частей системы. Одно рассмотрение конструктивное — времени создания, другое — функциональное, времени использования, и никакого изобретения не будет, если одновременно не удерживать оба этих рассмотрения. Системное творчество подразумевает одновременную работу с разными способами разбиения системы на части, и в каждом из этих разбиений будут абсолютно другие части! Например, ножницы конструктивно будут состоять из двух железных половинок и скрепляющего винтика, а функционально — из ножевого блока и рукоятки (а винтика так вообще в этом рассмотрении не предусмотрено, как и «половинок, скрепляемых потом винтиком»!).

Разные способы разбиения системы на части (среди них есть обязательные, прежде всего уже упомянутые четыре — функциональное, конструктивное, пространственное, стоимостное) будут подробнее рассмотрены в нашем руководстве позже, и это рассмотрение продолжится в руководствах по системной инженерии, инженерии личности, системного менеджмента.

Договориться о том, что все эти разные варианты разбиения системы относятся к одной и той же системе, относительно легко: нужно указывать на место в пространстве-времени, участвующее в разбиении. Если две разные физические сущности (например, функциональные объекты и материальные объекты) занимают одно и то же место в пространстве-времени (это тоже системное рассмотрение!), значит это одна и та же сущность. Если разговор идёт в типах (на уровне мета-модели предметной области «из учебника» прикладной дисциплины или регламента/стандарта, или уровне мета-мета-модели предметной области «из учебника» фундаментальной дисциплины, например, нашего руководства), то нужно обязательно давать примеры заземления/grounding. Крайне полезным бывает строить более конкретные описания, то есть демоделировать/рендерить/заземлять примерно так же, как схематические изображения из систем автоматизации проектирования или ручных эскизов «рендерят» в фотореалистичные изображения[242], добавляя детали, которых даже нет в исходных моделях, то есть уменьшая абстракцию, порождая/generate более подробное описание возможного состояния физического мира, отвечающее типам исходной более абстрактной модели.

Эмерджентность и мета-системный переход

Для того, чтобы какой-то набор физических частей был системой, нужно кроме признания факта, что эта система часть надсистемы и у неё самой есть части-подсистемы, удовлетворить ещё одному условию: эта система как набор взаимодействующих частей-подсистем должна проявлять какое-то свойство, которого нет у её частей-подсистем, ведущее к возможности новой функции целого-системы, в которые входят части-подсистемы. Это явление называют **эмерджентностью** (emergence, системный эффект). А ещё это свойство и новая функция пропадает у надсистемы, куда входит рассматриваемая система (но зато у надсистемы появятся новые/эмерджентные свойства, которых нет у системы как её части, этот системный эффект проявляется на каждом переходе от одного системного уровня к другому).

Показа времени нет ни в стрелках::часть механических часов, ни в их шестерёнках::часть, ни в корпусе::часть, ни в пружине::часть. А в часах::целое во время их работы показ времени возможен — в силу взаимодействия их частей. Каждая часть часов выполняет свою функцию в часах в целом, и возникает (emerge) системный эффект (проявляется эмерджентность, новое свойство, новая функция): часы начинают показывать время для надсистемы в системном окружении. Но вот комната (и даже ещё более мелкая подсистема комнаты, куда входят часы — интерьер), уже время не показывает, хотя показ времени часами в ней доступен и делает её удобней для проживания.

Обратите внимание: **пересечение вниманием границы системного уровня (называют также мета-системным переходом) меняет всё, он буквально**

---

[242] https://en.wikipedia.org/wiki/Rendering_(computer_graphics)

слышен в разговоре о системах, виден в тексте описаний:

- одни роли в связанных с часами проектах специализируются на методах, меняющими состояние уже собранных и работающих часов (используют эмерджентное свойство показа времени, создают интерьер комнаты, у которого другие эмерджентные свойства, нежели у часов), при работе с часами регулярно используется слово «время» как то, что надо будет увидеть (например, нужно ли добавить дополнительный светильник для того, чтобы видеть циферблат механических часов ночью, «посмотреть время») и даже не вспоминается о материалах для деталей часов (это на два системных уровня вниз от интерьера: сами часы и уже дальше детали — их материалы)
- другие роли — занимаются методами сборки часов (создают систему с эмерджентными свойствами часов), показываемое время их волнует в плане точности хода, а не использования.
- третьи роли ничего не говорят о самих часах, но просто изготавливают их детали, интересуясь прочностью и способами обработки металла — шестерёнки, пружинки, корпус. И у этих деталей тоже есть эмерджентные свойства, они ведут себя и характеризуются совсем не так, как ведут себя и характеризуются отдельные атомы, входящие в эти детали.
- … можно идти и дальше вниз по системным уровням: ещё какие-то роли занимаются материалами деталей часов: их прочностью, расширением при нагреве и т.д. — и это отражается в речи агентов в этих ролях как использование совсем другой терминологии.

Конечно, все эти системные уровни согласованы между собой, и регулярно в разговоре сборщиков часов упоминается и материал (его свойства влияют на свойства шестерёнок и стрелок), и время (часы ведь нужно наладить, чтобы шестерёнки и стрелки показывали правильное время!), и даже интерьер дома (обсуждение крепления часов на стене, размеров циферблата, чтобы его было видно через комнату). Но каждый системный уровень отличается своими системами, проявляющими какие-то свойства, меняющими свои состояния и в ходе работы, и в ходе создания из них надсистем методами создания самих этих систем и надсистем из этих систем (какие-то системы надо отливать в формах из расплавленного металла, какие-то — собирать из деталей, как панельные дома, какие-то — обучать, как мастерство). Эти методы меняют состояния частей системы, чтобы получить целую систему и запустить её в работу.

Разные методы создания и развития систем выполняются разными ролями в системе создания, эти роли говорят характерным языком, на котором обсуждаются системы одного системного уровня. Как мы увидим дальше — даже целым набором языков, ибо разные роли, специализирующиеся на обсуждении проблем какого-то системного уровня, используют разные языки для преследования своих интересов.

Разные роли ведут к специализации агентов на мастерстве выполнения методов работы этих ролей. Агенты, специализирующиеся на системах разных системных уровней, тоже будут разными. Люди (или даже AI, или подразделения), которые занимаются проектированием и изготовлением часов — это не те люди, которые занимаются интерьерами, и не те, которые занимаются материалами для деталей часов. Это и есть разделение труда: разные методы работы используются в проектах не просто сами по себе, а в силу появления эмерджентных свойств и функций у систем при смене системных уровней во внимании задействующих их ролей. Создатель интерьеров — архитектурное бюро, специализирующееся на

методах создания интерьеров, создатель часов — конструкторское бюро (дизайнеры, занимающиеся эстетикой часов — там) и работающий в партнёрстве с ним часовой завод, создатели материалов для деталей часов — заводы, которые будут поставщиками часового завода.

Переход внимания от одного системного уровня к другому — меняется набор ведущих ролей создателей, меняются их методы работы и объекты этих методов, меняется терминология этих рабочих культур/«языки разговора». Это происходит из-за смены набора понятий, выражающих предметы внимания системного уровня: когда внимание переходит с систем одного системного уровня на системы другого уровня, меняются понятия для частей системы и их важных свойств, поэтому меняется и терминология для выражения этих понятий.

Организм::целое животного прыгает и бегает в момент использования, а его органы::части — нет. Прыжки::поведение и бег::поведение — эмерджентные функции организма. Органы производят какие-то действия внутри организма, они имеют функции в организме (например, мышцы сокращаются, печень чистит кровь, лёгкие насыщают её кислородом и освобождают от углекислоты), это проявление их системных/эмерджентных свойств как целых органов, отдельные клетки внутри органов (части органов) этого делать сами по себе не могут, хотя именно клетки внутри органов вроде как всё и делают в их взаимодействии!

Системы не просто сами состоят из частей, они проявляют своё (субъективно определяемое!) назначение через выполнение какой-то (субъективно выделяемой!) их::часть функциональной роли в составе надсистемы::целое. В предыдущей фразе было упомянуто три системных уровня: 1. подсистемы, 2. целая система из этих подсистем, она проявляет эмерджентные свойства, является частью/подсистемой и имеет функцию в 3. надсистеме.

Основная особенность систем — это то, что «всё со всем связано», **части системы в системе ведут себя не так, как они же вне системы, ибо они взаимодействуют между собой.** Атомы вне молекулы ведут себя не так, как внутри молекулы. Клетки вне органа ведут себя не так, как внутри органа. Органы в организме ведут себя абсолютно не так, как органы отдельно от организма.

Чтобы разобраться в очень сложных системах, состоящих из огромного количества элементов, их представляют как разбиение/декомпозицию, на каждом уровне которых ожидают системного эффекта/эмерджентности. Например, вот индивидуальные детали автомобиля:

Разбираясь с этими индивидуальными деталями, невозможно понять, как автомобиль работает, для чего все эти винтики и тросики. Чтобы объяснить, откуда появляется движение автомобиля, мы должны рассмотреть двигатель как отдельную подсистему автомобиля, то есть двигатель::подсистема как целую часть машины::система (как бы ни противоречиво звучала эта «целая часть»), составленную из подсистем двигателя, ещё более мелких частей. Чтобы объяснить, почему в автомобиле удобно могут находиться несколько пассажиров, мы должны рассмотреть (выделить вниманием) салон::подсистему автомобиля::система как единое целое из многочисленных деталей, из которых составлен салон.

Мы должны рассмотреть отдельно собранные все детали::«подсистемы» тормозной системы::«система» автомобиля::«надсистема» (для разработчиков тормозной системы автомобиль — надсистема), чтобы показать, каким образом автомобиль может тормозить. Всё это эмерджентные свойства, бесполезно обсуждать для понимания работы тормозной системы не её детали, а опускаться сразу на системный уровень ниже и обсуждать материалы, из которых сделаны детали тормозной системы. То есть эмерджентность/системный эффект торможения появляется, когда материалы уже стали деталями, и торможение происходит из-за взаимодействия уже изготовленных и правильно расположенных в пространстве деталей. Хотя формально «для математиков» торможение происходит из-за взаимодействия материалов, и даже молекул материалов! **Системное мышление — это не математическое мышление, оно ориентировано на управление вниманием для обсуждения каких-то действий агентов, а не формальную верность.**

Корова Маргарита имеет своей частью хвост, корова Маргарита является частью коровьего стада.

Нехорошо позволять говорить, что коровье стадо имеет хвост, хотя это вроде как корректно: все молекулы хвоста (того самого: коровы Маргариты) входят в молекулы стада. Причем стадо тут не абстрактный объект «множество коров», а вот прямо-таки все молекулы коров в загородке загона для скота. Говорить «хвост стада» математически, логически, физически корректно, но совсем не системно, и это вроде как интуитивно понятно всем: трудно предположить, что вы можете делать с «хвостом стада».

Выделение системных уровней субъективно и существенно зависит от метода, по которым идут работы с системами этих уровней. Так, интуитивно не понятно, можно ли говорить, что карбюратор — часть автомобиля с двигателем внутреннего сгорания. Он отдельная часть автомобиля, или он часть топливной подсистемы, или часть двигателя?! Какие действия надо выполнять с карбюратором, на каком системном уровне надо его рассматривать? Это будет решать инженерная команда в каждом проекте, она должна договориться внутри себя. Но интуитивно понятно: неправильно говорить, что поршень или цилиндр двигателя — это часть автомобиля. Формально это верно, но неправильно, как и в случае хвоста как части стада, вы не сможете обсуждать функционирование поршня и цилиндра непосредственно в автомобиле.

Хороший критерий тут — эмерджентность: обсуждение автомобиля в целом обычно требует обсуждения свойств и функций двигателя в целом, но есть ли там внутри двигателя поршень и цилиндр — это при обсуждении автомобиля в целом неважно.

На каждой границе системного уровня меняются системы как важнейшие объекты внимания, а также связанные с ними «соображения»/«озабоченности»/«предметы интереса»/«важные характеристики»/concerns. При мета-системном переходе (переходе от одного системного уровня к другому — от частей к целым или от целых к частям, изменение уровня гранулярности[243]) вместе с методами работы меняются ведущие дисциплины/теории/модели/алгоритмы этих методов, описывающие поведение задействованных методами работы системы на данном системном уровне, меняется профессиональное сообщество ролей создателей, поддерживающее язык разговора на этом уровне и использующее эти дисциплины/знания методов вместе с необходимым инструментарием.

На одном уровне обсуждаем хвосты и рога в корове (со всей проблематичностью выделения их как частей с определёнными границами!), на другом — целых коров и быков в составе стада как надсистемы. На одном уровне обсуждаем двигатель и

[243] https://en.wikipedia.org/wiki/Granularity

салон, на другом — поршень и цилиндр в двигателе. Разные системы задают разные предметы интереса, разные языки разговора об этих предметах интереса, разные роли с разными предпочтениями в одних и тех же характеристиках как предметах интереса. Часто эти роли играют даже разные люди-актёры, ибо разные методы работы требуют разного мастерства, а все виды мастерства в приемлемой степени одному человеку получить обычно не удаётся.

Нужно чётко понимать, что сами по себе границы всех упомянутых систем «необъективны», это какие-то деятельностные роли в проекте разработки автомобиля для удобства своих обсуждений часть деталей в совокупности (иногда в системном мышлении это даже называют «целокупостью», подчёркивая, что это не просто «сборка», а создание нового целого из частей) называют «двигатель», другую целокупную/собранную часть деталей «салон», третью — «тормозная система».

Собирать вниманием отдельные части в целое для того, чтобы обсудить проявляющийся системный эффект — это сердцевина системного подхода, самое в нём главное. В инженерии отдельные части ещё и изготавливают отдельно, а потом и физически собирают — но это уже инженерия, а не мышление. В мышлении эти части выделяют вниманием, а вот будет там сборка, или выращивание, или обучение, или ещё какой-то способ изготовления частей на самых разных уровнях — это обсуждается отдельно, при обсуждении инженерных методов.

Эмерджентность в системном мышлении — это главный критерий для объединения каких-то объектов в систему. Если вы собрали в системе какие-то объекты по критериям принадлежности к предметной области/domain (заменили отношение «часть-целое» отношением классификации по принадлежности) и не проверили, что их взаимодействие (а они не взаимодействуют в момент эксплуатации! Они ж просто «принадлежат», являются «частью» в бытовом смысле, а не физической частью! Бытовое использование «части» не различает физичность и ментальность!) даёт эмерджентность — вы неправильно определили систему, у вас это просто объект, выделенный вашим вниманием, а не система! **Нет системного эффекта — не система!** Три детали на складе на одной полке — не система, нет взаимодействия, нет эмерджентности, нет нового свойства, не нужно выделять систему! Это просто три детали, лежащие на одной полке.

А вот клиентура, собранная из отдельных клиентов — это система. Так, вы можете обсуждать рост клиентуры (число клиентов), обсчитывать значения оттока клиентов, выбирать какой-то «канал продвижения» для клиентуры как сообщества. Получается, что клиенты вроде как не общаются между собой, не взаимодействуют, но взаимодействие таки есть: через компанию, у которой они — клиенты и которая собирает из них целое — клиентуру во взаимодействии с собой. Это может быть проще понять, рассмотрев аккумуляторную батарею: отдельные аккумуляторные элементы работают каждый «внутри себя» и не взаимодействуют друг с другом, но вот вместе они дают «батарею» — и свойства этой батареи определяются числом этих элементов. Вот «батарея» из клиентов — это и есть клиентура, свойства её другие, нежели свойства отдельных клиентов, а взаимодействием клиентов можно обычно пренебречь, но нельзя пренебречь происходящим «внутри клиента».

В силу эмерджентности на каждом системном уровне появляются новые свойства и функции, которые нужно отдельно, но подробно обсуждать агентам, обладающим

нужным мастерством в методах работы с этими свойствами::характеристики и функциями::поведение. Так, специалисты по автомобильной мебели могут обсуждать удобство мебели отдельно от специалистов по двигателям, которые будут обсуждать мощность двигателя.

Все проектные/деятельностные роли будут реализовывать предпочтения в важных для них характеристиках/предметах интереса, их методы работы глубоко связаны друг с другом. Понятие системного уровня даёт способ структурировать реализацию этих предпочтений. Предметы интереса и меняющие их методы работы центрируются вокруг систем на разных уровнях системного разбиения, буквально «на разных уровнях крупности деления вещества». В центре теории/знаний/дисциплин/алгоритмов этих методов/практик/способов/культур работы — предметы интереса к эмерджентным свойствам систем.

Проводимые в проекте работы по самым разным методам учёта интересов дают максимальное проявление **полезной эмерджентности** или предотвращают появление **вредной эмерджентности**. Конечно, и «полезность», и «вредность» тут субъективны, они определяются ролями исходя из знаний/дисциплин выполняемых ими методами работ: весь проект сводится тем самым к сначала достижению договорённостей между ролями по поводу значений важных для них характеристик/concerns (проект должен дать успешную систему! Все должны договориться!), а затем к воплощению этих договорённостей в физическом мире через выполнение работ по изменению самых разных систем самыми разными методами.

Системные уровни нужны для того, чтобы структурировать достижение этих договорённостей, лучше понимать происходящее: обсуждаются всегда системы из каких-то определённых системных уровней, которые и проявляют свои характеристики, воспринимаемые разными ролями как важные/предметы их интереса. Нужно убеждаться, что обсуждение ведётся специалистами по объектам этого уровня. Точность хода часов обсуждается часовщиками, интерьер обсуждается спецами по интерьеру, а материалы деталей часов обсуждаются специалистами по материалам. При этом обычно специалисты как-то разбираются в характеристиках своего уровня, а также в характеристиках подсистем («из чего сделано») и надсистем («для чего сделано»).

Вопросы. Системное разбиение и эмерджентность

Что является частью объекта «стол из гостиной»?
--- Столетний дуб, из которого был сделан стол. ### Нет, столетний дуб исчез из мира в тот момент, когда был распилен для того, чтобы потом стать столом (потерял форму — перестал занимать место в пространстве-времени).
+++ Шуруп, крепящий столешницу к ножке. ### Да, шуруп (молекулы шурупа) занимает часть места в пространстве-времени, которое занимает стол в целом (все молекулы стола в целом).
--- Обломки стола на полигоне бытовых отходов 1 января 2019 года. ### Нет, это уже не стол, а его обломки.
+++ Трехмерная столешница стоящего в гостиной стола на 13:30 1 января 2016 года. ### Да, эта столешница — часть стола: она делит с ним

занимаемое пространство и занимаемое время (хотя это и отрезок времени нулевой длины).
+++ Четырехмерная ножка стоящего в гостиной стола в период с 13:30 1 января 2016 года по 14:30 3 января 2016 года. ### Да, ножка — часть стола, она делит с ним часть занимаемого столом пространства-времени.
+++ Царапина на столешнице стоящего в гостиной стола ### Да, царапина занимает какое-то конкретное место в пространстве-времени, и это место является частью места, занимаемого столом в целом.

Выберите из приведённого ниже списка систем те, которые входят в разбиение из максимальной по длине цепочки отношений «часть-целое» (максимальное число системных уровней) в какой-то системе:
+++ Электродвигатель ### Да, это система нижнего из 4 уровней системного разбиения
+++ Троллейбус ### Да, это система третьего уровня из четырёх, часть городского общественного транспорта.
+++ Городской общественный транспорт ### Да, это второй сверху уровень системного разбиения, часть города.
--- Городская дорожная сеть ### Нет, всего два уровня системного разбиения: город — городская дорожная сеть
+++ Город ### Да, это верхний уровень самой длинной цепочки из 4 уровней системного разбиения.
--- Городская энергосистема ### Нет, это средний уровень в одной из двух цепочек из трёх уровней системного разбиения (город — городская энергосистема — тяговая подстанция или городская электростанция)
--- Тяговая подстанция ### Нет, это нижний уровень в цепочке из трёх уровней системного разбиения (город — городская энергосистема — тяговая подстанция)
--- Городская электростанция ### Нет, это нижний уровень в цепочке из трёх уровней системного разбиения (город — городская энергосистема – городская электростанция)

Какие из перечисленных последовательностей являются системными разбиениями:
+++ Надсистема-система-подсистема ### Да, речь идёт о двух отношениях часть-целое физических объектов
--- Класс-подкласс-член класса ### Нет, это отношение специализации (подмножества) класс-подкласс и отношение классификации (класс-член класса)
--- Множество-подмножество-элемент множества ### Нет, это отношения специализации (множество-подмножество) и классификации (множество — элемент множества).
--- Тип-объект-метод ### Нет, это просто три разных слова из терминологии объект-ориентированного программирования.

Что из перечисленного наиболее корректно метафорически описывает суть понятия «эмерджентность»?
--- (A + B) >= (A) + (B) ### Нет, при эмерджентности целое не может быть равно сумме частей, всегда появляется что-то новое.
+++ (A + B) > (A) + (B) ### Да, целое больше суммы частей.
--- (A + B) = (A) + (B) ### Нет, целое не равно сумме его частей — при эмерджентности должно появиться что-то ещё, новое свойство.
--- (A + B) != (A) + (B) ### Нет, целое не просто не равно сумме частей — целое больше суммы частей, обязательно появляется что-то добавочное, новое свойство, которого нет в частях.
--- (A + B) ? (A) + (B) ### Нет, целое именно больше, чем сумма частей (а не непонятное отношение).

Неустроенность и мета-системный переход

Эволюция::«процесс/изменения» — «физическая эволюция вселенной» это всеобщее явление, но мы всё-таки разделим её на отдельные варианты:

- **Биологическая эволюция.** Усложнение[244] идёт от молекул к клеткам, от клеток к организмам, от организмов к популяциям и т.д.
- **Меметическая эволюция.** Это эволюция понятий-мемов (meme[245], образовано так же как ген — gene, и не путайте с интернет-мемами), которые живут как в символьных локальных представлениях (одно место — один символ), так и в коннекционистских[246] распределённых представлениях/distributed representations (а в последнее время добавились distributional representations[247]), например, в нейронных сетях (в головах людей, в системах AI на базе больших языковых моделей — это всё нейронные сети разного вида). Например, понятие-мем (иногда при обсуждении меметики говорят «идея», иногда даже «мысль») о том, что «если кошка перебежала тебе дорогу, то это нехорошо». Эта очень странная идея/мысль путешествует по мозгам, как вирус (повторяет себя, реплицируется), и почему-то не умирает окончательно: обязательно находится какой-то ещё один мозг, в котором живёт эта идея. Или идея Ньютона, что массивные тела притягиваются друг ко другу силой гравитации, эта идея тоже не умирает, хотя Эйнштейн выдал идею получше: никакой силы гравитации и в помине нет, а есть искривление пространства-времени массами. Эта идея тоже живёт, на её основе мы имеем GPS навигацию, но она не вытесняет полностью идею Ньютона про силу гравитации. Тут те же закономерности, что и в любой эволюции (ибо любая эволюция сводится к накоплению знаний). Современные рыбы (их знания накапливаются в геноме) тоже не вытеснили эволюционно древнюю рыбу латимерию, хотя остальные древние рыбы давно вымерли. Природа явлений техно-эволюции как эволюции идей та же, что в биологической/дарвиновской эволюции. Репликация какой-то информации между нейросетями (людей и сейчас AI-агентов) в техно-эволюции в меру точная, но иногда проходит с мутациями. В одном случае эволюционируют гены (получаются разные организмы, их

[244] https://www.pnas.org/doi/10.1073/pnas.1807890115
[245] https://en.wikipedia.org/wiki/Meme
[246] https://en.wikipedia.org/wiki/Connectionism
[247] https://www.frontiersin.org/articles/10.3389/frobt.2019.00153/full

сложность и разнообразие растут, но простые виды не исчезают окончательно), а в другом случае — эволюционируют мемы (получаются разные теории/объяснения, их сложность и разнообразие растёт, но простые теории/объяснения не исчезают окончательно). Эволюция (дарвиновская и техно-эволюция) вполне физична, но она связана с тем, что накапливается информация на каких-то носителях, эта информация реплицируется — и надо глядеть первоисточники, на которые указано в нашем руководстве, чтобы разобраться. Этот материал в руководстве будет не столько разъясняться, сколько использоваться. В какой-то момент инженеры-менеджеры, работающие с руководством, не выдерживают, и смотрят в первоисточники — и формируют таки у себя физический-биологический, а не «общебытовой» взгляд на эволюцию.

- **Техно-эволюция.** Мутации технических систем производят главным образом люди путём изобретений (догадываются, что для получения какой-то функции::поведения в функциональных объектах можно использовать какие-то конструктивные объекты как аффордансы/«подходяшки»), и мы имеем развитие электротранспорта, робототехники, водопроводов, небоскрёбов и смартфонов, хотя вилка-ложка-нож, лопата и молоток остаются довольно давно неизменными, разве что существенно улучшены материалы, из которых они изготавливаются, да и само изготовление стало существенно дешевле. Важно, что техноэволюция использует сейчас существенно цифровое/дискретное (то есть с точной репликацией) символьное представление описаний систем.

В нашем руководстве мы не будем касаться механизмов эволюции, при этом мы уже упоминали в разделе руководства «2. Наш вариант системного мышления: третье поколение» в подразделе «Варианты системного подхода» достаточно материалов, чтобы пытливые инженеры-менеджеры могли разобраться в физической природе происходящего (и лучше бы не оттягивать это знакомство).

Основная идея в современном взгляде на эволюцию — это что явление **неустроенности/неустаканенности/frustrations** ведёт к огромному разнообразию квазиустойчивых **конфигураций** (наборов вариантов подсистем, вариантов конструкции) эволюционирующих систем, причём каждая из этих конфигураций квазиоптимальна, а переход между квазиоптимумами невероятно лёгок и поэтому общий набор конфигураций самых разных систем оказывается неустойчивым, неравновесным.

Нам понятие неустроенности потребуется для того, чтобы обсуждать разные варианты эволюции (и не путайте с психологическими «фрустрациями», в английском языке это слово имеет множество значений, да и оригинальный термин был — geometrical frustration[248], никакого отношения к мыслительным затруднениям, речь идёт о физическом явлении).

В упомянутой статье «Physical foundations of biological complexity»[249] понятие неустроенности поясняется следующей иллюстрацией:

---

[248] https://en.wikipedia.org/wiki/Geometrical_frustration — и посмотрите на видео, которое очень хорошо объясняет суть этой «неустроенности/неустаканенности».

[249] https://www.pnas.org/doi/10.1073/pnas.1807890115

Fig. 2.

Complexity of fitness landscapes and nonergodicity of evolution.

Она показывает, что у эволюционирующих систем в каком-то определённом смысле есть память (физики говорят про системы с памятью «не эргодические[250]», они не хаотично блуждают по всем своим состояниям), они запоминают конфигурацию «оптимума выживания». Но в реальной ситуации память не запоминает найденное конфигурационное решение намертво, ибо существует очень много разных конфигураций системы, у которых примерно одно и то же квазиоптимальное «соответствие нише»/fitness, между которыми очень маленькая разница в «оптимальности». Конечно, есть самые разные конфигурации систем, которые не смогут выжить, и их большинство, но и конфигураций, в которых системы выживут, предостаточно, и они очень мало отличаются друг от друга.

В приведённых картинках это не одна конфигурация с безусловной принадлежностью к квазиоптимуму, а остальное — не выживет, а огромное количество конфигураций, которые эволюция может проходить в ходе бесконечных мутаций. И ни одна из этих конфигураций не будет окончательной, «целью и итогом эволюции»! Эволюция тем самым оказывается бесконечной, хотя и является хорошо описываемым процессом оптимизации конфигурации какой-то системы.

Скажем, вы приходите в магазин смартфонов и видите одновременно сотню очень похожих друг на друга моделей, все из которых находят какой-то рыночный спрос. Это и есть проявление эволюции: огромное разнообразие более-менее выживающих в разных экологических нишах[251] (в нашем случае — техно-экологических нишах, то есть техно-ценозах, а не биоценозах) конфигураций (то есть наборов вариантов подсистем, вариантов конструкции) каких-то эволюционирующих систем. Это разнообразие и неустроенность как множество

[250] https://ru.wikipedia.org/wiki/Эргодичность
[251] https://ru.wikipedia.org/wiki/Экологическая_ниша

вариантов конфигурации, каждый из которых квазиоптимален, поэтому переходы от одного варианта к другому почти незаметны в части оптимальности включения в нишу, верны для всех видов эволюции — био, мемо, техно. Ну, и время от времени случаются скачки в уровне оптимизации. Скажем, мобильные телефоны были заменены на смартфоны — «нарисованная тастатура» оказалась лучше, маленькие размеры телефонов заменились на «лопаты» больших размеров — большой экран оказался лучше. Динозавров уже нет, хотя когда-то оптимальными были именно динозавры. А сегодня нет старых мобильных телефонов, хотя когда-то оптимальными до 2007 года (выпуск iPhone) были именно они.

При этом нужно аккуратно обращаться с типами связанных с эволюцией понятий. Скажем, реплицирующиеся в нейросетях мемы могут пониматься и как гены из генома, то есть части наследственного материала (гены физичны! Это буквально физические части молекул ДНК), и как реплицируемая в ходе обучения нейросети информация, то есть не быть материальными. Но тут помогают ходы, в которых мем может быть рассмотрен как «элементарная программа», базирующаяся на физической аппаратуре нейросети (например, на мозге) — это рассмотрение близкое к рассмотрению мастерства, как системы из запрограммированной на выполнение метода части физического мозга. В других школах мысли мем — это информационный объект, описание для программирования мозга.

При этом даже с биологическим геном есть разночтения: классическое понимание гена в том, что он физичен (часть ДНК), но используют понятие «ген» часто не как собственно «ген»::система, а как «информация, закодированная геном»::описание (например, так можно использовать и слово «мастерство» — не то, что умеет выполнить агент, а «описание метода, которое содержит в себе агент»).

В любом случае, разбирайтесь по контексту, с каким типом вы имеете дело: геном и мемом как информацией или геном и мемом как физическими объектами. С «мастерством», «программой» и прочими объектами такого сорта (алгоритм и универсальный вычислитель/computer для его выполнения, «расширенный/обобщённый на изменения/трансформацию физического мира алгоритм» из метода работы и универсальный создатель/constructor с инструментарием для его выполнения) — то же самое, будьте осторожны в присваивании типа, разбирайтесь с контекстом, в котором употреблён термин.

Скажем, вы легко поменяете описание метода (например, поменяете учебник игры на рояле с учебника для классической музыки на учебник для джазовой музыки), это просто заменить одно локальное/символьное представление на другое. Но вот заменить мастерство игры на рояле так просто не удастся, переучить одно мастерство на другое (одно распределённое представление на другое) не так-то просто. Заменить ген в «генах-описании» (заменить символ) — легко. А вот заменить физический ген в организме не так просто. Экземпляр физического гена будет представлен в каждой клетке организма, надо будет дотянуться до каждой клетки, или прицелиться точно в замену гена в зиготе (помните ещё из школьного курса биологии, что такое зигота?).

Ещё одна проблема — это понимание того, что именно эволюционирует, какая система. В биологии это проблема биологического индивида: эволюционирует геном (набор всех генов), организм с феномом (организм с набором всех проявленых свойств, определяемыми геномом), популяция (ибо для эволюции требуется самец и самка, а это уже больше одного организма — популяция), вид

(все особи, относимые к виду)? Точного ответа на этот вопрос нет, и будет сразу встречный вопрос: а какой у вас проект, как вам в этом проекте удобней для того, чтобы менять мир к лучшему? Современные результаты также показывают, что бесполезно обсуждать эволюцию только одного вида, ибо эволюция всегда идёт только в контексте эволюции всех остальных видов.

Особи/индивиды/экземпляры каждого вида эволюционирующих био, социо, техно систем состоят из многочисленных подсистем и входят в надсистему, как и любые другие системы. И сложность надсистем растёт за счёт того, что в какой-то момент эти надсистемы начинают объединяться друг с другом, выполняя разные (функциональные) роли в целой уже над-надсистеме, то есть совершая очередной мета-системный переход. Это есть и в техно-эволюции (эволюции инструментария): молотки, как и микробы, живут сами по себе. Но вот телевизор, навигатор, телефон, часы, записная книжка, пульсометр и музыкальный центр как функциональные объекты (в которых важна функция, но неважна конструкция) вдруг оказались одним более сложным конструктивно устройством — смартфоном, в котором живут как подсистемы множество устройств поменьше, «приложений»::подсистем, выступающих чем-то типа органов в организме. Это вот оно и есть: какие-то из удачных для их объединения систем какого-то системного/эволюционного уровня/масштаба объединяются в надсистему — и сложность биологического, меметического, технического мира растёт.

Читаем по геометрическим неустроенностям объяснения в статье[252], росту сложности биологических (и социальных! И технических!) систем из-за этих неустроенностей как происходящих от конфликтных взаимодействий на разных системных/эволюционных уровнях статью[253], термодинамические объяснения и математику для обоснования эволюционных процессов как процессов познания/learning и в том числе как причины появления жизни в статье[254], и сами подходы к теории эволюции как многоуровневому обучению, в том числе понимание, что эволюция сама по себе многоуровневая и движется вот этими неустроенностями[255]. Вот как в этих работах определены основные принципы (понимаемые как основные онтологические догадки, то есть объекты, в терминах которых можно обсуждать происходящее, «аксиоматический подход» в физике) дарвиновской эволюции, то есть эволюции жизни, где геном внутри организма. Техно-эволюция, где мемом в конструкторском бюро, а целевая система делается на заводе и не содержит в себе мемома, или меметическая эволюция, где мемом лежит где-то в форме фрагментов видео на серверах или «идей» в бумажных книгах — это тоже эволюция, там могут быть небольшие отличия в трактовках, но в целом там будут выполняться те же принципы:

1. **Функция потерь/loss function.** В любой эволюционной системе существует функция потерь от переменных/физических величин, зависимых от времени. В ходе эволюции потери минимизируются, эволюция — это оптимизационный (поиск минимума функции потерь) процесс.
2. **Иерархия масштабов времени.** Эволюционирующие системы включают множество меняющихся во времени/динамических характеристик, которые

[252] https://en.wikipedia.org/wiki/Geometrical_frustration — и посмотрите на видео, которое очень хорошо объясняет суть этой «неустроенности/неустаканенности».
[253] https://www.pnas.org/doi/10.1073/pnas.1807890115
[254] https://www.pnas.org/doi/full/10.1073/pnas.2120042119
[255] https://www.pnas.org/doi/10.1073/pnas.2120037119

меняются на различных временны́х масштабах (с различными характерными частотами: секунды, годы, тысячелетия).

3. **Разрывы частот.** Характеристики, проявляющиеся на разных организационных/системных/эволюционных уровнях (уровнях системы/организации), разделены достаточно широкими (три порядка величины) разрывами в частоте.
4. **Ренормализуемость.** По всем уровням организации статистические описания быстроменяющихся (высокочастотных) характеристик получаются через описания медленнее меняющихся (низкочастотных характеристик).
5. **Расширение свойств.** Эволюционирующие системы имеют способность добавлять дополнительные характеристики, которые могут быть использованы для выживания системы и возможность исключать характеристики, которые могут дестабилизировать систему (эмерджентность для позитивных свойств и устранение эмерджентности для проявляющихся негативных свойств).
6. **Размножение.** В эволюционирующих системах размножение/репликация и исключение из размножения информационно-обрабатывающих единиц (IPU, information-processing units, мы их называли «агентами») может быть на каждом системном/организационном/эволюционном уровне.
7. **Поток информации.** Системы медленно «эволюционирующих»/меняющихся системных уровней передают информацию системам быстро меняющихся в ходе познания/обучения/learning уровней, чтобы системы быстро меняющихся уровней могли лучше предсказывать состояния как окружающей среды, так и окружения системы в целом. В центральной догме молекулярной биологии добавляют, что обратный ход информации устроен по-другому: от «быстро обучающегося организма» к медленно меняющемуся «геному» информация идёт не путём передачи, а другим механизмом: мутации и смерть организмов с неподходящими мутациями. В техно-эволюции и мутации не любые из-за неминуемых сбоев в точности репликации, а smart (с расчётным наиболее вероятным успехом, не случайные).

А вот феномены жизни, которые из них следуют:

1. **Наличие единиц, обрабатывающих информацию** (IPU, information processing units) как отдельных систем на всех уровнях организации, то есть какая-то автономия и устойчивость существования молекул, клеток, организмов, популяций и биосферы в целом — они как-то поддерживают свою идентичность, не растворяются в окружении.
2. **Неустроенность.** Цели поддержания устойчивости (NESS, речь идёт об устойчивости в неравновесной ситуации, грубо говоря, «не дать разрушиться, поломаться, рассеяться в окружении») систем разных уровней конфликтуют, и это порождает неустроенность. Неустроенности/неустаканенности бывают или пространственные (geometrical frustrations), или во времени. Так, нейрон в мозге имеет своей функцией реакцию на входящий сигнал, это означает, что он будет реагировать на этот сигнал не как хочет, а как нужно для обработки сигнала целой группой/кластером нейронов как надсистемой, в которую входит система-нейрон. Это пространственный конфликт целей: оптимизировать обработку сигнала «как попроще» в рамках самого нейрона не получится, ибо это будет конфликтовать с обработкой сигнала «как надо» со стороны как кластера нейронов, и далее по системным уровням — со

стороны каких-то долей мозга, отделов мозга, далее всего мозга. С другой стороны, нейрон работает с какой-то своей частотой, в своём временном масштабе, но это может не соответствовать временно́му масштабу, в котором ему нужно работать в более вышестоящей единице организационного устройства (кластере нейронов, долей и отделов мозга, мозге). И пространственные, и временны́е неустроенности не могут быть устранены полностью, но в рамках какого-то оптимального баланса может быть достигнуто локальное (не глобальное!) устойчивое неравновесное состояние — поиск этого устойчивого неравновесного состояния (NESS) и есть познание/обучение/learning эволюционирующей системы.

3. **Иерархия системных уровней.** Если система эволюционирует, то у неё с необходимостью будет множество системных уровней (рост сложности в ходе эволюции неизбежен).
4. **Субоптимальность** (near optimality). Для сложных многоуровневых оптимизаций можно использовать только стохастические методы, которые не гарантируют как устойчивость решения (то есть при попытке повторить будет получаться слегка другой вариант), так и нахождение глобального минимума (будут получаться не абсолютно оптимальные варианты, а просто более оптимальные, чем многие другие — но сравнимые с другими возможными к нахождению вариантами). В биологических системах через 4 миллиарда лет эволюции это проявляется в том, что **всё уже более-менее оптимизировано, поэтому изменения в системах чаще всего ведут к ухудшениям, очень редко к улучшениям и в общем случае вообще ни к чему не приводят, ибо не так далеко уж уводят от уже найденного локального минимума потерь. Изменения, приводящие к переходу на локальный (никогда не глобальный, он недостижим принципиально) очередной уровень минимума потерь, крайне редки.**
5. **Разнообразие субоптимальных решений.** Есть огромное конструкционное/конфигурационное разнообразие систем, для которых функция потерь субоптимальна и имеет очень близкое значение. Это само по себе свойство неустроенных систем, при этом неустроенные системы неэргодичны[256], то есть они имеют какую-то память и поэтому эволюционные их пути скорее расходятся, чем сходятся. Конструкционное разнообразие неустроенных систем растёт, и все они имеют очень близкие значения функции потерь.
6. **Разделение фенотипа и генотипа.** Критически важным в феномене жизни оказываются два нарушения симметрии: а) «цифровая» с дискретными значениями, стабильная, медленно обновляемая память и по большей части аналоговые системы обработки информации и б) информационный поток между IPU/системами разных уровней как прямые «инструкции» генотипа фенотипу, а фенотип не может изменить генотип прямо, и изменения идут через мутации. Пример тут — «центральная догма молекулярной биологии», что информация идёт от ДНК к белку, но не наоборот. В компьютерной архитектуре то же самое: в архитектуре фон Неймана память отделена от центрального процессора, и поток инструкций идёт из памяти в процессор. И если речь идёт о любой более-менее сложной «информационной системе»::система, то память::подсистема как-то отделяется от

[256] https://ru.wikipedia.org/wiki/Эргодичность

процессора/обработчика::подсистема.

7. **Размножение.** Существование долговременной цифровой памяти (например, ДНК или РНК для генома) даёт возможность стабилизации IPU/систем — если бы памяти не было, то каждый раз приходилось бы информацию о стабильной системе получать эволюцией заново (и помним, что эволюционные процессы сразу обсуждают шкалу времени минимум на три порядка более медленную, чем возможная скорость нарушения обсуждаемой стабильности, там ведь «поколения»), но вместо этого она просто копируется. Цифровость/дискретность вместо аналоговости в памяти должны быть достаточно выражены, чтобы не накапливалась ошибка при многократном размножении/репликации, если точность репликации недостаточна, то жизнь не получается. **На данный момент только биосфера представляет собой механизм передачи информации об устойчиво существующих системах, который не сломался за примерно четыре миллиарда лет.**
8. **Естественный отбор.** Дарвиновская эволюция по принципу естественного отбора возникает из комбинации всех принципов и феноменов жизни, которые были описаны. IPU отличаются от среды и друг от друга, они разнообразны, они стабильны и содержат в себе кроме обработчиков/процессоров информации ещё и цифровую память (геном), а ещё они могут размножаться, сохраняя эту память (передавая геном потомству). Стабильность наиболее удачных/стабильных IPU при этом будет потихоньку расти, они будут чаще встречаться. Менее удачные/стабильные (с высокой функцией потерь в результате накопившихся мутаций) будут исчезать. Ключевая особенность дарвиновской эволюции: генотипы размножаются, но это размножение ограничивается откликом от окружающей среды, передаваемой через фенотипы (которые либо далее успевают размножиться, либо не успевают, уж какими свойствами их снабдил для этого генотип). Похоже, что этот феномен присущ главным образом биологической жизни, а не любой эволюции.
9. **Паразитизм.** Паразиты и совместная эволюция паразитов и хозяев повсеместны на множестве организационных уровней и являются неотъемлемой чертой эволюции жизни. Паразиты высоковероятны, а высокоэффективная иммунная система дорого стоит в части затрат энергии. В краткосрочной перспективе паразиты вредны, а в долгосрочной перспективе полезны, ибо являются источником новых функций в хозяине, от паразитизма возможен переход к симбиотическому существованию и даже полной интеграции. Паразиты и хозяева могут быть одноуровневыми (паразиты-организмы живут на хозяевах-организмах), но могут быть и транс-системны (генетические элементы паразита паразитируют на геноме организма или клетки-паразиты паразитируют на многоклеточных организмах хозяина). Этот феномен тоже вроде как присущ главным образом биологической жизни, но не техно-эволюции (впрочем, хакеры могут не догадываться о таком замечании, а их работа имеет и теоретическое обоснование[257]).
10. **Программируемая смерть** является неотъемлемым свойством жизни. И это относится к биологической эволюции, не факт, что это нужно соблюдать,

[257] https://ailev.livejournal.com/1622346.html

когда мы рассматриваем другие виды эволюции (скажем, техно-эволюцию).

Тем самым запоминайте и постарайтесь учесть в мышлении:

- **Мета-системные переходы по линии увеличения числа системных уровней, которые можно легко наблюдать (рост сложности), появляются в ходе эволюции. Рост числа системных уровней происходит из-за неизбежных неустроенностей между системными уровнями, выживают только какие-то конфигурации систем, наиболее точно решающие задачу многоуровневой (а не одноуровневой!) оптимизации, и появление более высоких системных уровней увеличивает устойчивость полученных решений.** Самый успешный на своём системном уровне (например, раковая клетка печени очень успешна! Она размножается без ограничений!) может вымереть из-за того, что не учтена многоуровневость оптимизации конфигурации. Империи, которые растут как раковые опухоли, неминуемо распадаются, ибо будут встречать сопротивление с более высокого уровня и более низкого уровня (оптимизация многоуровневая!), а устойчивого состояния страновых границ не будет никогда: эволюцию не остановишь, её суть как раз в нахождении этих концептуальных новаций, увеличении разнообразия видов в ходе естественных мутаций в биологической дарвиновской эволюции, или в ходе инженерных изобретений при меметической (например, социальной, культурной) или техно-эволюции (технических систем). Страны тут такой же объект, как и любой другой, многообразие видов стран будет только расти.
- **Для каждого мета-системного перехода между системными уровнями внимание рассматривает новые свойства систем каждого уровня за счёт эмерджентности, поэтому меняются методы работы с этими системами, язык разговора об этих системах, дисциплины/теории/объяснения функций/методов работы этих систем.** Нельзя функции/методы работы этих систем объяснить в терминах функций/методов работы надсистемы (грех холизма, работу человеческого организма нельзя объяснить политическими теориями или даже теориями семейной жизни — это теории более высоких системных уровней), нельзя функции/методы работы этих систем объяснить через методы работы подсистем (грех редукционизма, нельзя работу человеческого организма в целом объяснить через понятия кардиологии или проктологии). Это означает, что нельзя слишком центрироваться на переносе свойств людей на системы других уровней: нельзя обсуждать работу человеческих органов так, как будто это маленькие люди, нельзя объяснять происходящее в международной политике так, как будто страны — это такие огромные люди. Отслеживайте, что вы используете адекватный язык обсуждения, меняйте объяснительные теории при переходе с уровня на уровень. **Например, у города не может быть инфаркта, но у отдельных людей в нём — может быть, страна не может «обидеться», но обижаются в стране отдельные люди — особенно часто застревание происходит на уподоблении самых разных систем человеку. И «организация» — это будет «лицо» навроде человека. Но «оживлять» в уме и приписывать намерения, желания можно и смартфону, и городу, и стране. Отслеживайте такое и у себя в мышлении, и в других, деантропоморфизируйте по возможности и учитывайте наличие**

многих системных/эволюционных уровней/масштабов.

Вопрос, насколько можно говорить об обидевшемся смартфоне, городе, стране, нации и даже каком-то пользовательском сообществе — он просто решается для неживого смартфона, пока в нём нет сильного AI. Но вот для сообществ агентов это уже не такой простой вопрос. Мы тут придерживаемся принципа методологического индивидуализма, но есть и другие точки зрения на этот вопрос[258].

Нужно также отметить, что математические описания физических процессов, познания/learning и эволюции удивительно похожи. Мы уже говорили о том, что математика задаёт набор удобных понятий для типизации, а физика задаёт набор функциональных объектов, удобных для описания взаимодействий в физическом мире. В рамках исследований с использованием физики и математики происходит довольно жёсткая проверка (сначала логикой и рациональными рассуждениями, а потом и экспериментом) того, что поведение математических объектов в мысленных экспериментах соответствует поведению физических объектов, взятых как функциональные объекты по отношению к конструктивным/материальным объектам. При этом будет использован принцип 4D экстенсионализма: физический рассказ про мир в терминах функциональных объектов из учебника физики типа «физическое тело» и рассказ про мир в терминах учебника предметной области, где вместо физического тела будет какая-нибудь «ракета» позволяют отождествить мир учебника физики и мир жизни.

Посмотрите пример таблицы, которую сделали на базе сравнения похожести математических описаний в термодинамике, теории нейронных сетей (глубокое обучение/deep learning) в машинном обучении и теории эволюции[259]:

[258] https://ailev.livejournal.com/1735626.html

[259] https://www.pnas.org/doi/full/10.1073/pnas.2120042119

Corresponding quantities in thermodynamics, machine learning, and evolutionary biology

	Thermodynamics	Machine learning	Evolutionary biology
$\mathbf{x}$	Microscopic physical degrees of freedom	Variables describing training dataset (nontrainable variables)	Variables describing environment
$\mathbf{q}$	Generalized coordinates (e.g., volume)	Weight matrix and bias vector (trainable variables)	Trainable variables (genotype, phenotype)
$H(\mathbf{x},\mathbf{q})$	Energy	Loss function	Additive fitness, $H(x,q) = -T\log f(\mathrm{q})$
$S(\mathbf{q})$	Entropy of physical system	Entropy of nontrainable variables	Entropy of biological system
$U(\mathbf{q})$	Internal energy	Average loss function	Average additive fitness
$Z(T,\mathbf{q})$	Partition function	Partition function	Macroscopic fitness
$F(T,\mathbf{q})$	Helmholtz free energy	Free energy	Adaptive potential (macroscopic additive fitness)
$\Omega(T,\mu)$	Grand potential, $\Omega_p(\mathscr{T},\mathscr{M})$	Grand potential	Grand potential, $\Omega_b(T,\mu)$
T or $\mathscr{T}$	Physical temperature, $\mathscr{T}$	Temperature	Evolutionary temperature, T
μ or $\mathscr{M}$	Chemical potential, $\mathscr{M}$	Absent in conventional machine learning	Evolutionary potential, μ
N_e or N	Number of molecules, N	Number of neurons, N	Effective population size, N_e
K	Absent in conventional physics	Number of trainable variables	Number of adaptable variables

Близость формул, которые описывают явления термодинамики, познания как обучения нейронных сетей и эволюции показывает, что термодинамика, познание/learning, эволюция — это всё разные описания разными словами примерно одного и того же набора закономерностей, одного и того же процесса, происходящего в природе. Тем самым системное мышление, которое обсуждает эволюционно появляющуюся многоуровневость и разнообразие в чём-то стабильных систем (удерживающих NESS, их называют «системами» или «агентами», или «IPU»), пригодно для мышления об очень широком классе жизненных явлений.

Системный подход против и редукционизма, и холизма.

Системный подход (подход/approach — это перенос разработанного в физике понятия «система» на другие области, прежде всего биологию) появился как раз для того, чтобы бороться с **редукционизмом** — попытками описания сложных объектов без выделения системных уровней и связанных с ними системных эффектов (эмерджентностей). Поскольку редукционисты не выделяют отдельных системных уровней, они ведущий метод/функцию и его дисциплину/теорию/объяснение какого-то более низкого системного уровня (частей системы, иногда это много уровней вниз от уровня рассматриваемой целой системы) выпячивают как средство объяснения поведения всей системы в целом. Так, поведение человека редукционисты могут объяснять химическими и электрическими процессами, которые проходят в мозгу между основными клетками мозга, нейронами. Верно ли это? Да, это верно, математически абсолютно верно! Но совершенно бесполезно! Помним про «хвост стада»: трудно понять, какие операции можно делать с подсистемами на несколько уровней ниже, если обсуждать свойства текущего уровня. Хотя, конечно, можно как-то оттрассировать/отследить, как какие-то свойства системы следуют из свойств её подсистем, причём эти свойства при переходе на каждый уровень подсистем с увеличением числа уровней отслеживать будет труднее и труднее. Скажем, можно ещё отследить, что «кирпичные дома обычно теплее, чем панельные», если заметить, что железобетонные армированные панели прочные и поэтому тонкие, а кирпичей приходится класть много и стенки там толще. Но если тянуть эту линию дальше на много системных уровней к материалам — обожжённой глине, бетону и арматуре, то объяснение становится запутанным. Без «панели» и «кирпича», без «стен» (системы промежуточных системных уровней) объяснения не получатся.

Поведение текущего уровня в системном мышлении объясняется взаимодействием систем многих системных уровней вверх и вниз от текущего, что не снимает тезиса «упрощенчества» при каждом переходе от более верхних уровней к более нижним в целях объяснений. Но системный подход не случайно выбирает системные уровни для подобных объяснений, а берёт устойчивые (хотя и неравновесные, NESS) физические объекты в качестве используемых для объяснений систем, и ещё строит объяснения принципиально многоуровнево/многомасштабно — и при этом не пропускает системные уровни, чтобы не впадать в редукционизм.

В классическом редукционизме никаких многих системных уровней в объяснениях нет, а отсылки идут только к нижележащим уровням. В редукционизме нет никаких интеллектуальных средств для предотвращения объяснения поведения человека даже не поведением нейронов его мозга, но вообще квантово-химическими процессами с участием электронов и элементарных частиц атомных ядер, которые лежат в основе химических процессов, или наоборот — клеточными процессами, для которых основой служат химические процессы с клеточными молекулами. Ещё очень модны объяснения, как работа гормонов на уровне биохимии влияет на поведение человека (человек в целом — это же много-много системных уровней выше уровня биохимии происходящего в мозгу). Конечно, биохимия работы мозга влияет на поведение человека, но связь эта настолько неочевидна, что требуется много системных уровней для её прояснения: предмет разговора меняется на каждом системном уровне, этими уровнями занимаются разные деятельностные роли, знания эти описаны в разной литературе. Так что огромное количество поп-психологической литературы, объясняющей низкоуровневыми

нейрофизиологическими механизмами поведение человека только вводит в заблуждение, ничего не объясняет, не даёт оснований для надёжной работы по изменению поведения человека по образцу инженерной работы. Поп-психология на то и поп-психология, что не следует строгим стандартам мыслительной работы, а только даёт ощущение «объяснений». Почему ракета летит? Потому что в ней окисляется топливо! Почему мозг понимает текущие строчки? Потому что в нём срабатывают нейроны! Всё это правда, но бесполезная для содержательной и точной инженерной работы правда. Чтобы эта правда была полезна и точна для инженерной работы, нужны многоуровневые (в плане системных уровней) рассмотрения, и в них не должны пропускаться рассмотрения высоких уровней, не должен проскакивать редукционизм.

Не будем всё происходящее с человеком сводить только к мыслительной работе, у человека есть ещё и выход в физический мир через телесную работу. Танцевание, как одно из возможных поведений человека (другие поведения — жизнь, отдых, труд), можно «объяснять» в кавычках, редукционистски, как набор химических процессов между молекулами в клетках мышечной ткани человека, или даже как набор движений сотен мышц. Можно даже при этом упоминать, что мышцы управляются мозгом, а в ходе танцевания (жизни, отдыха, труда) задействуется и спинной мозг. Но эмерджентности именно танцевания (свойств, присущих танцору как роли агента, когда он танцует — выразительность, эстетичность, набор фигур, характерные позы) в этом не будет, не будет обсуждения танцора и его поведения (танцевания). Все эти редукционистские описания танцевания как мышечной работы будут формально-логически верны, но это будут редукционистские описания, сводящие эмерджентные свойства к свойствам частей системы, в том числе свойствам частей, лежащих на много системных уровней ниже, чем уровень, удобный для описания обсуждаемого эмерджентного свойства.

У коровьего стада есть хвост (коровы Маргариты), но обсуждая этот хвост, стада не обсудишь. Обсуждая мышцы и спинной мозг, танцора не обсудишь! Для обсуждения танцора в ходе танцевания (а Вася Пупкин если не танцует, то и не танцор — когда Вася Пупкин обедает, то он «едок», когда лечится — «пациент», а танцор — только когда танцует) нужно и движение мышц описывать, и работу всего тела в разных «неравновесных балансах», и использование в движении всего тела инерции частей тела (голова — это 5.4кг, если махнуть как следует головой — то тело улетит за ней примерно так же, как улетит за арбузом с подобным весом), и принципы танцевального стиля (и тут надо будет задействовать эстетику, а ещё стиль культурно-обусловлен), и как эти принципы воплощаются в отдельных движениях (лексика танца), и как эти движения собираются в целый танцевальный перформанс/исполнение танца (хореография — тут), и почему танцевания подобного рода повторяются в разных сообществах мастеров танцевания (скажем, почему танго танцуют в самых разных местах на планете, и вы можете узнать, что это танго, и люди-агенты, которые его танцуют, как-то принимают решение научиться танцевать и становятся мастерами танго — это мы уже обсуждаем уровень сообщества, много выше уровня мышц!). Каждый шаг этого обсуждения делается с отдельными объектами, объяснить «как устроен танцор» через «мышцы и нервы» нельзя, поведение танцора невыразимо через поведение «мышц и нервов». Если вы описываете устройство городского квартала, то вам нельзя описывать его через расположение и форму кирпичей, и даже стен в зданиях. Надо описывать квартал через расположение и форму зданий и дворов, газонов и скверов, пустырей и проезжих частей дорог. Кварталы, танцоры, стада — надо быть

внимательным, на каком «системном уровне»/«уровне частей объекта» мы пытаемся описать поведение той или иной системы как целого.

Редукционистские методы (описание стада в терминах хвостов и рогов, а не коров) всё-таки работали, но работали плохо. Если редукционисты случайно угадывали удобный системный уровень (танцор как задействующий методы удержания постуры/«танцевальной осанки», танцевальных шагов в ритм, занятия каких-то положений в паре, если это парные танцы, квартал как состоящий из зданий и дворов, стадо из коров и быков) — редукционистские методы работали, а если не угадывали (танцор из мышц и костей, квартал из кирпичей и травинок на газонах, стадо из ушей, ног и хвостов), то ничего хорошего не происходило. Со стороны это выглядело как «редукционизм — благо!» (и перечисление удачных случаев) и «редукционизм — вред!» (и перечисление случаев неудачных).

Классический пример первых систем, на которых оттачивался системный подход в момент его становления в виде общей теории систем Ludwig von Bertalanffy — это цветущий луг весной. Сотня видов растений и животных, почва, оставшиеся от разлива лужи и болотца. Как обсуждать происходящие на лугу сезонные изменения? Биохимией взаимодействия всех живых клеток всех животных и растительных организмов на цветущем лугу? Понятно, что это не срабатывает, хотя формально верно — но как это интуитивно понятное «не срабатывает, хотя формально верно» перевести на уровень явного принципа? Вот тут и пригождаются понятия системных уровней и эмерджентности, и тем самым появился «системный подход», как перенос разработанного в физике понятия «система» на другие предметные области. Обсуждать такую сложную систему, как цветущий луг, стало проще. Этот сложный объект начал «помещаться в голове», мышление о нём «разлеглось по полочкам».

Эмерджентность нужно отличать от **синергии** — эффекта взаимоусиления свойств. Если при объединении двух компаний с небольшой прибылью мы наблюдаем, что их взаимополезность и прибыль резко растёт, то мы наблюдаем просто более сильную итоговую компанию, никакого системного эффекта/эмерджентности (появления у целого нового свойства, новой характеристики, нового предмета интереса от взаимодействия частей этого целого) нет, есть просто синергия этих исходных компаний в части старых свойств. Не появилось никаких новых свойств, просто усилились прошлые свойства. Никакого системного эффекта, никакой эмерджентности, никакого нового предмета и новых ролей, связанных с новыми свойствами объединённого в целую систему синергичных частей этого целого. Но налицо синергия: свойства усилились! Системный подход тут ни при чём.

А вот если соединить кирпичи и цемент с водой в правильной форме, то из их взаимодействия появится дом — и можно обсуждать комнаты, жильцов дома, что для обсуждения просто раствора и кирпичей просто невозможно. Кирпич в цементе и воде ничего не усиливает, ничему не способствует, цемент у кирпича ничто не усиливает, но если их взять в достаточном и правильном количестве и соединить сначала цемент с водой (получив раствор: свойства раствора несводимы к свойствам воды и свойствам цемента), а потом кирпичи с раствором, то будет дом — свойства дома будут несравнимы со свойствами кирпича, цемента и воды. И никакой синергии при этом нет, то есть нет взаимоусиления свойств цемента, воды и кирпичей при их объединении в два уровня: «кирпичи плюс «цемент плюс вода»».

В домах живут, в кирпичах не живут, даже сложенных в кучку. Хотя формально живут в кирпичах с раствором (редукционизм!), но в этих терминах трудно

обсуждать жизнь. Например, у дома есть архитектурный стиль (в смысле строительной архитектуры, «визуальный образ») — модерн, барокко — а у кирпичей с раствором этого архитектурного стиля нет, его в терминах кирпичей и раствора не обсуждают, он только уже у домов в их готовой форме. Но элементы этого архитектурного стиля — это просто та или иная выкладка кирпичей, чистая правда. Это и есть редукционизм, «правда, но бесполезная правда», обсуждать дом так нельзя! Синергия (сложение частей без появления новых качеств, но изменением старых качеств частей) тем самым может обсуждаться в рамках редукционизма, а эмерджентность/системность редукционизм исключает.

Вторая ошибка, которую исправляло системное мышление — это обратный редукционизму **холизм.** При этом само системное мышление плохо владеющие им люди (и наверняка это унаследовано AI-агентами) относят к холистическим учениям, то есть учениям, объясняющим поведение системы поведением её надсистемы. Танцоры танцуют так, как они сейчас танцуют, ровно потому, что человечество обеспечивает существование субкультуры танцевания (и для зрителей, и для себя, и даже для друзей-подруг: танец живота в гаремах развивался именно как такое развлечение). Эта субкультура танцевания как набор мемов в головах людей (а сейчас и на других носителях — кино, видео в социальных сетях) заставляет людей танцевать ровно так, как они танцуют, а мышцы и нервы людей отрабатывают этот «приказ культуры», помогают воспроизводить танцевание как распространяющее эти мемы, никуда они не денутся, ибо «поведение нижних системных уровней полностью зависит от того, что происходит вовне их», основная мысль холизма. Для холизма танцевальная культура возникает не потому, что люди танцуют и совокупность их паттернов поведения (методов работы), перформансов и рассуждений о танцах называют культурой, но наоборот, для холизма — танцуют (феном танцевальной культуры) из-за того, что этого требует танцевальная культура (мемом танцевальной культуры) как набор всех идей/мемов/мыслей по поводу танцев и связанного с ними человеческого поведения. И для холизма идеи про постуру, танцевальные шаги в ритм, идеи про «подёргивания мышц», опорно-двигательный аппарат и его состояние в танцах объявляются несущественными. Для холизма всё это «как-нибудь само подстроится под общественный заказ» (и не спрашивайте, что такое «общественный заказ»: идеи высоких уровней, связанных с сообществами и обществами довольно мутны, хороших онтологий нет, поэтому язык больше художественный, в нём легко фантазировать, в том числе антропоморфизировать общество — и приписывать ему, например, «заказ»).

Даже не будем приводить подробных объяснений, почему чистый холизм такая же ошибка, что и редукционизм, ибо нелепость утверждений холизма вскрывается очень быстро и поэтому он реже встречается. Но тоже встречается, холизм очень распространён в гуманитарной среде. Инженеру в тепловозостроении трудно внушить, что у паровоза КПД в ходе его работы такой низкий, потому как это железнодорожные пути и вокзалы сделали его таким, а не физические процессы определяют КПД паровоза по сравнению с КПД электровоза. Но гуманитарий легко может утверждать, что танцевальный перформанс у танцора такой, какой он случился — это прямое следствие «общественной потребности». Что это такое «общественная потребность», «запросы общества» — это даже не будем обсуждать. Но слова-то привычные и не вызывают у гуманитариев вопросов. Хотя вопросы есть: что такое «потребность» у «общества» из многих отдельных людей с совершенно разными идеями по поводу танцев и не только танцев, и как она проявляется в конкретном перформансе в части работы мышц и нервов?!

Системное мышление ассоциируется с холизмом потому, что в многоуровневых рассуждениях учитывается взаимодействие уровней (включая неизбежные конфликты, которые приводят к необходимости многоуровневой оптимизации конфигураций систем всех уровней, а результатом тут будет неустроенность этих конфигураций и рост сложности систем в ходе эволюции. Системное мышление признаёт и существование вышестоящих уровней, и влияние их на нижние (влияние целого на части), как в холизме. Но в системном мышлении учитывается также и существование нижних уровней, влияние их на верхние (влияние частей на целое), как в редукционизме. Причём одновременно, и ещё на многих системных уровнях сверху вниз и снизу вверх!

В современном (третьего поколения) системном мышлении ещё и говорится о многоуровневой оптимизации всех этих влияний в ходе эволюции: ибо все эти части и целые появились такими, какие они есть, в ходе эволюции (био/дарвиновской, меметической, техно). При этом нужно рассматривать масштабы времени в тысячи раз более длинные, чем масштабы времени в ходе непосредственного взаимодействия систем в момент их использования — в танцах какая-нибудь фигура длится полсекунды, но эволюция её появления заняла годы, а иногда и десятки лет, а появление мышц и нервов агентов-танцоров на более медленно меняющихся низких уровнях заняло миллионы лет, и развитие по этой линии продолжается. Так, уже есть исполнение танца агента с шестью руками: две свои и четыре механические из рюкзака с манипуляторами[260], эволюция танцующего агента продолжается, но это не очень быстрая эволюция.

Системное мышление и редукционистское, и холистическое одновременно. Ну, или ни редукционистское, ни холистическое. Оно снимает противоречия между двумя этими подходами к объяснению поведения систем.

Целевая система и коллективное системное мышление

Для того, чтобы как-то проводить системные (т.е. с использованием понятий системного подхода) рассуждения, нужно как-то научиться управлять фокусом своего внимания на разные системы в многоуровневом системном разбиении — и уровнях выше целевой системы, и уровнях ниже целевой системы. Системный мыслитель всё в мире рассматривает как системы, сами входящие в системы более высокого системного уровня как подсистемы, и имеющие свои подсистемы более низкого системного уровня, поэтому прежде всего нужно выделить какую-то из них, к характеристикам которой мы проявляем интерес, и пытаемся что-то в мире изменить для реализации этого интереса/предпочтения. Эта выделенная вниманием агента (субъективно! Удобным для ролевого метода работы агента способом!) система и будет **целевая система** (system-of-interest, буквально «система нашего интереса»). Это та в будущем успешная система, с которой мы::«команда проекта» что-то хотим делать: придумать и создать её, починить, эксплуатировать, уничтожить.

Дальше можно обсуждать любой системный уровень выше и ниже уровня, на котором расположена целевая система, но на этой системе остаётся фокус нашего::«команды проекта» внимания, системный эффект именно этой системы нас будет интересовать прежде всего. Одна камера внимания каждого агента,

[260] https://vk.com/wall-179019873_1747

участвующего в проекте, всегда удерживает целевую систему, другие камеры внимания для разных отыгрываемых этими агентами ролей могут при этом бродить по разным системам других уровней и по системам графа создателей. Но на целевой системе внимание у каждого агента закреплено хотя бы одной камерой внимания в ходе проекта всегда, про неё не забываем.

Метафора тут — полярные координаты, всё (коллективное! Не только одной роли!) мышление крутится вокруг целевой системы. Это точка отсчёта, движение внимания в системном мышлении отсчитывается от неё. Если потерялись с длинными рассуждениями среди огромного множества систем в проекте, то вернитесь к целевой системе, и дальше двигайтесь по цепочкам разных отношений (главным образом отношений композиции и создания) ко всем остальным системам. Целевая система даёт стабильность внимания самых разных ролей в коллективных проектах.

Для разных ролей целевыми системами представляются самые разные физические объекты, но в ситуации коллективной деятельности предполагается, что все роли, выполняющие методы непрерывного создания и развития целевой системы (всё, что с ней происходит с момента замысла через множество модернизаций до прекращения использования) **договорились** о том, какая же это система, где эти границы. **Системное мышление прежде всего поддерживает не мышление одного человека, а поддерживает коллективное мышление в организации как множестве агентов, которые могут известным методом (этот метод — тот или иной вид менеджмента) распоряжаться их совокупным трудом и ресурсами**. Организации — это команды проектов, предприятия/компании/фирмы, некоммерческие организации, кружки, учебные группы и прочие, где понятно «кто начальник». Если это сообщество, то скорее всего начальника нет, системного мышления вокруг одной целевой системы может и не случиться, договорённостей о том, что делаем (о целевой системе) не будет. Но если это сообщество кем-то организуется и формальное вступление в сообщество означает то, что просьбы этого организатора будут выполняться в каком-то объёме — то тут с системным мышлением всё может быть в порядке (ну, или не может быть, если люди специально ему не обучались).

Системное мышление помогает агентам (в том числе людям) договориться друг с другом о своих действиях, и прежде всего договориться о том, какая система их интересует, какая система является целевой, каковы границы (что входит, а что не входит в состав) целевой системы. Так что **системное мышление — коллективное мышление, а не мышление одного агента, оно предполагает договорённости о предмете деятельности, то есть о целевой системе. Целевая система не ваша::агент, не какой-то одной роли, она общая для команды проекта, и о ней команда договаривается (в том числе в договорённостях участвуют и внешние проектные роли). Конечно, можно вообразить и задействование системного мышления одним агентом, но этому агенту придётся выполнять множество самых разных ролей, поэтому он не сможет задействовать разделение труда — высокую квалификацию в выполнении отдельных ролей. Создание сложных систем потому и возможно, что системное мышление позволяет договориться разным агентам, каждый из которых обладает высоким мастерством выполнения разных ролей — визионеров, проектировщиков, архитекторов, инженеров производственной платформы и т.д.**

Целевая система обычно мыслится **на момент времени её использования** (operations, эксплуатации, функционирования, работы) уже в готовом виде, когда она взаимодействует со своим окружением/средой/environment и играет в этом окружении свою роль/выполняет функцию. То есть «самолёт» мыслится, когда он летит, «компьютер» — когда он считает, «мастерство» — когда выполняется работа по методу этого мастерства. Целевая система во время её создания всё-таки ещё не в том состоянии, когда она что-то делает, выполняет функцию. При создании системы мы больше думаем о функциях создателей по отношению к изменениям состояния частей будущей целевой системы, эти части меняют свои состояния и в ходе производства собираются в готовую систему, это всё «время создания». А когда думаем про целевую систему и системные уровни — это время использования/эксплуатации/функционирования/работы целевой системы.

На рисунке представлена системная (по отношениям композиции) иерархия с тремя системными уровнями, системы показаны кружками, стрелки с ромбиками показывают отношения композиции/разбиения/состава/часть-целое, целевая система показана как система 2:

Система, в состав которой входит целевая система, называется **надсистема.** На рисунке это система 1. Часы будут надсистемой для шестерёнки, молекула — надсистемой для атома. Целевая система имеет свою функцию (поведение с ожидаемыми результатами) в надсистеме, её функция позволяет надсистеме проявить в конечном итоге эмерджентное свойство, т.е. выполнить функцию надсистемы. Функция показа времени у часов как целевой системы в интерьере как надсистеме помогает интерьеру быть удобным для проживания.

Помним, что все системы определяются прежде всего по их функции в надсистеме, они в надсистеме играют какую-то роль. И надсистема — тоже система, ей нужно выполнять свою роль в над-надсистеме (просто на картинке эта над-надсистема не показана, но она всегда есть!).

Если целевая система шестерёнка, то шестерёнка используется в часах (входит в состав работающих часов как надсистемы), её роль «передатчик движения», назначение/функция::поведение — передача (для функций используем отглагольные существительные) в надсистеме-часах движения на стрелки так, чтобы часы могли показывать время, т.е. надсистема (часы) могла «выполнять своё назначение»/функционировать/«выполнять свою функцию»/«выполнять свою роль»/работать.

Все системы в момент эксплуатации целевой системы, которые не входят в целевую систему, называются **системами в окружении** (environment, среда, operation environment, рабочая/эксплуатационная среда, рабочее/операционное окружение — термин «окружение» предпочтительней, поскольку подчёркивает центральную роль целевой системы, а термин «среда» не подразумевает какого-то явного центра). **Окружение — это всегда в момент работы готовой системы.**

Когда целевую систему (впрочем, и подсистему, и надсистему) создают, там будет другое рассмотрение: в центре будут какие-то из систем создания, а будущие системы находятся в неработоспособном ещё состоянии «сырья», «полуфабриката». На картинке системы создания (условно изображены как оргзвено из людей) изображены отдельно от иерархии системного разбиения, это ведь вообще другое время (время создания, а не время использования/эксплуатации/функционирования/работы целевой системы, её подсистем и её надсистемы).

Частая ошибка — это считать, что системы создания (enabling systems, constructor systems) находятся в системном окружении. Нет, они не входят в системные уровни той иерархии по отношению «часть-целое», к которой принадлежит целевая система. Отношение их с целевой системой (и другими системами обсуждаемых системных уровней, то есть надсистемой, подсистемами) — отношение создания (на диаграмме оно не показано).

На рисунке одна из систем в окружении (этих систем множество!) — система 3. Например, для шестерёнки в часах таким окружением будут стрелки, тоже входящие в состав часов. А ещё в окружении могут быть какие-то системы, даже не входящие в состав надсистемы, но без которых трудно обсуждать работу/функционирование целевой системы — хотя они входят в какую-нибудь над-над-надсистему. Например, солнце, нагревающее часы и тем самым влияющее на шестерёнку (при нагреве она может поменять свои размеры, что может оказать влияние на основное поведение — точно передавать движение, что далее оказывает влияние на поведение надсистемы — точный показ времени). **Окружение — это системы из состава надсистемы, над-надсистемы и т.д. Главное, что окружение — это вовне границ целевой системы и речь идёт о моменте, когда система работает и выполняет свою функцию (а не когда она задумывается, изготавливается, испытывается).**

Заправочная станция для целевой системы такси, входящего в состав таксопарка как надсистемы — это система в окружении. Дорога для едущего по дороге такси — это система в окружении. Повторимся: совершенно необязательно, чтобы система в окружении была именно подсистемой надсистемы, «смежником» для целевой системы. Хотя иногда все другие (кроме целевой системы) подсистемы надсистемы для целевой системы выделяют специально, называя **ближним окружением**, а за пределами надсистемы системы называя **дальним окружением.** Для

автомобильного мотора в составе автомобиля как надсистемы салон автомобиля и его колёса — ближнее окружение, это подсистемы автомобиля. А вот дорога и палящее солнце — это системы из дальнего окружения).

Подсистема — какая-то часть системы. В системном мышлении подсистемы в момент их использования рассматриваются главным образом как роли, функциональные объекты, причём только после рассмотрения окружения и определения функции целевой системы в окружении — ибо пока мы не понимаем, что должна делать целевая система, какую функцию/поведение она несёт в окружение, мы не можем ничего сказать про её состав и уж тем более не можем обсуждать конструкцию (какие там конструктивы, задействованные во время создания, играют роли выделенных подсистем). На рисунке пример такой подсистемы целевой системы показан под номером 4.

Проблема в том, что целевой системой для разных проектов может стать любая, которая будет проявлять интересную для них эмерджентность, нужный для них системный эффект. И тогда все остальные виды систем будут определяться по-другому. Скажем, если целевой системой в каком-то проекте объявить систему 4, то система 2 будет в этом проекте её надсистемой. И если все в том проекте договорились, что целевой системой будет система 4, то так тому и быть — именование разных систем в том проекте будет другое, хотя состав систем в системной иерархии будет одним и тем же. Если какая-то фирма делает интерьеры, то они в ней будут «целевая система», а если интерьерные часы — то «целевая система» — они, а если массово изготавливает шестерёнки (в том числе и к часам) — то «целевой системой» будет шестерёнка. И все разговоры будут крутиться вокруг целевых систем. А как же взаимодействие между проектами? Придётся договариваться: участники этих трёх проектов друг для друга будут «внешними ролями». При этом общения между проектом интерьера и проектом изготовления шестерёнок не будет, а вот часовщикам придётся общаться и с заказчиками — интерьерной фирмой, и с поставщиками — фирмой по изготовлению шестерёнок.

Форматы описания системных уровней

Системные уровни состоят из иерархии по отношению часть-целое/composition. Эти иерархии вроде как удобно изобразить картинкой графа-«дерева», вроде той, которую мы только что обсуждали в связи с именованием систем по отношению к целевой системе. Для иллюстративных целей это подходит, для работы — нет. Графические модели крайне неудобны в редактировании, в изменении. При разработке системных разбиений, их согласовании в команде проекта вам нужно будет быстро (это ключевое) менять варианты системного разбиения. Для этого мы предлагаем использовать «аутлайн», текстовое представление дерева иерархии, где каждый уровень представляется отступом.

Вот предыдущая картинка как аутлайн, проставлены те же цифры, что и на картинке:

- Надсистема (1)
    - Система в окружении
    - Система в окружении (3)
    - Целевая система (2)
        - Подсистема

        - Подсистема (4)
        - Подсистема

Конечно, эти именования (как и любая терминология) более-менее условны. Так, в ТРИЗ надсистема так и называется — надсистема, а англоязычные системные инженеры обычно слово «надсистема» не говорят (очень редко не-инженеры говорят suprasystem, но не supersystem), хотя и говорят «подсистема» (subsystem).

В основополагающем стандарте системной инженерии ISO 15288:2023 вообще не говорят обо всех этих видах систем, подчёркивая их одинаковость: различают только целевую систему (system-of-interest) как вершину системного разбиения. «Выше по системным уровням» идут сразу «системы в окружении», которые рассматривают вообще отдельно, включая саму надсистему. В составе целевой системы в этом стандарте только системы (если у этих систем будут части) и системные элементы (elements, если было принято решение не рассматривать их части, а только ограничиться существованием этих системных элементов как целых, являющихся частями их надсистем).

ISO/IEC/IEEE 15288:2023(E)

Figure 2 — System-of-interest structure

Дадим эту картинку как многоуровневый список/аутлайн, заодно переведя на русский:

- Целевая система
    - Система
        - Элемент системы
        - Элемент системы
        - Система
            - Система
                - Элемент системы
                - Элемент системы
                - Элемент системы

        - Элемент системы
        - Элемент системы
    - Система
        - Элемент системы
        - Элемент системы
        - Система
            - Система
                - Элемент системы
                - Элемент системы
            - Элемент системы
            - Система
                - Элемент системы
                - Элемент системы
    - Элемент системы

Картинка и текст тут примерно одинаковой наглядности, ибо представлено очень немного элементов в системном разбиении. Теперь представьте, что нужно изобразить 500 элементов (масштаб проблемы: в авиалайнере или ледовой буровой платформе 5-6 миллионов элементов). Диаграмма-картинка сразу будет помещаться только на многих сложно сопоставляемых друг с другом страницах, она не влезет ни на один компьютерный монитор, а если её бить на части, будет абсолютно нечитаема. Правки в этой диаграмме будут очень долгими и многие правки будут вести к ошибкам.

Текстовый вариант с аутлайном легко править, он легко вытягивается в просто «длинный текст» (люди привыкли работать с многостраничными текстами, используя скроллирование), длинные имена свободно помещаются в строку (даже ещё можно и комментарии давать!). Так что используйте для работы с системными разбиениями аутлайны/outlines/«nested lists»/«многоуровневые списки».

Альтернативный способ — это использовать явную нумерацию уровня как отдельный элемент списка. Вот та же структура, что в предыдущем списке, только представлена в табличной форме (например, Excel таблице). Целевая система как целая при этом в нашем варианте представления не имеет кода, а число позиций в коде — это системный уровень (1.3.1.1. — это будет четвёртый уровень, если целевая система на нулевом уровне. Как ни печально, но в стандарте ISO 15288 на картинке опечатка в нумерации уровней, четвёртый уровень показан как пятый!)

Код системы	Имя системы
	Целевая система
1.	Система
1.1.	Элемент системы
1.2.	Элемент системы
1.3.	Система
1.3.1.	Система
1.3.1.1.	Элемент системы
1.3.1.2.	Элемент системы

1.3.1.3	Элемент системы
1.3.2.	Элемент системы
1.3.3.	Элемент системы
2.	Система
2.1.	Элемент системы
2.2.	Элемент системы
2.3.	Система
2.3.1	Система
2.3.1.1	Элемент системы
2.3.1.2	Элемент системы
2.3.2.	Элемент системы
2.3.3.	Система
2.3.3.1.	Элемент системы
2.3.3.2	Элемент системы
3.	Элемент системы

Конечно, в этом примере таблицы «система» и «элемент системы» — это типы (из мета-мета-модели, типы важных объектов системного подхода) тех объектов, которые будут заданы в таких таблицах или типами мета-модели «из учебника предметной области», или даже типами мета-модели предприятия, или даже для конкретного экземпляра системы — именами конкретных физических объектов. То есть в таблице будет стоять не «система», «система», «система», а для автомобиля «шасси», «левое колесо», «двигатель», для парусника «корпус лодки», «мачта», «грот» и т.д.

Каждый раз используйте перевод графических представлений в иерархии, а иерархии представляйте затем в таблицах — иначе вы не справитесь с сопровождением, не сможете вносить многочисленные изменения. Форматы должны быть удобны не для «представления», а изменения: описания систем (в том числе описание системных разбиений) — это предмет непрерывных переговоров в команде проекта, а поэтому — предмет непрерывных изменений. Форматы подгоняются под удобство изменений, всяческая «наглядность» простых примеров «из учебника» и «из стандарта» не должна отвлекать: учебники и стандарты вам не надо менять, изображения из них вам не надо разбивать на много страниц. Используйте табличные представления (в том числе и для работы с системными разбиениями), это будет удобно.

Системные уровни в системной инженерии. Пример вычислительной техники.

В современной вычислительной технике легко насчитать десяток системных уровней. Вот пример выделения этих уровней от нижних к верхним для классических компьютеров (их больше, тут приведён только пример мышления об этих уровнях, разные производители могут выделять их по-разному для разных проектов):

1. Чистые материалы (кремний и тщательно дозированные его «загрязнения», дающие эффект полупроводимости).
2. Части транзистора (исток-сток-затвор).
3. Полевой транзистор и межтранзисторные соединения.
4. Логические элементы (состоят из нескольких связанных транзисторов).
5. Арифметически-логические устройство/вычислительное ядро, память и прочие части процессоров и разных видов памяти.
6. Процессоры/чипы: общего назначения (CPU), графических/матричных вычислений (GPU), сетевой коммутации между процессорами, оперативной памяти.
7. Компьютерные платы с процессорами и памятью, внешними устройствами (например, твердотельными накопителями).
8. Компьютерный рэк с несколькими компьютерными платами, связным оборудованием ввода-вывода, системой питания и охлаждения. Связные кабели.
9. Серверная стойка для нескольких компьютерных рэков с питанием и охлаждением.
10. Поды (pods) из нескольких связанных друг с другом серверных стоек.
11. Датацентр из нескольких «подов», в его состав может входить даже робот телеприсутствия с видеокамерой для наблюдения ночью. Суперкомпьютер уровня датацентра (по состоянию на июнь 2020 седьмой в мире по мощности) мог быть уже несколько лет назад собран за 3.5 недели всего 6 инженерами[261].

Чистые материалы в масштабах частей транзистора имеют размеры, сравнимые с размерами атомов (ориентир в 2023 году был 1 нанометр для выпуска к 2030 году, хотя речь идёт не совсем о физических размерах транзисторов[262]). Датацентры с суперкомпьютерами сегодня — это десятки тонн оборудования, сотни кубометров по объёму. Инженеры организуют своё мышление вокруг системных уровней: системы на каждом из этих уровней состоят из частей нижележащего уровня и сами являются частью систем более высокого уровня. Системы каждого из системных уровней разрабатываются инженерами, умеющими решать проблемы этих уровней. Вместе все эти уровни дают мощность вычислений, которую нельзя получить, используя редукционистский подход, «суперкомпьютер прямо из чистых материалов» и даже «суперкомпьютер из транзисторов».

Совершенно необязательно называть системы на всех этих уровнях «система», «система транзистор», «система серверная стойка». Система — это тип объекта. Мы можем «приговаривать» к каждому слову его тип, но это будет звучать странно. Мы не говорим «процесс летит система самолёт», мы говорим «летит самолёт». Мы не говорим «система центральный процессор в составе системы компьютера», мы говорим «центральный процессор в компьютере». Это не убирает факта, что мышление инженеров вычислительной техники системно, их деятельность организована вокруг системных уровней вычислительной техники, этих системных уровней много, только в нашем сверхупрощённом примере этих уровней

[261] https://blogs.nvidia.com/blog/2020/08/14/making-selene-pandemic-ai/
[262] https://www.verdict.co.uk/rapidus-1-nm-chip-technology/

одиннадцать.

Приём опускания типа «система» (включая разнообразие видов систем — целевая, надсистема, подсистема, система в окружении, далее будет ещё и система создания) важен, если вы разговариваете с людьми, не знакомыми с системным мышлением. Для коллеги, который знаком со стандартами системной инженерии и менеджмента, или знаком с нашими руководствами для инженеров-менеджеров, будет уместно сказать «X — это целевая система», что эквивалентно X::«целевая система». Для других людей лучше сказать «X» или «целевой X» или даже «важно учитывать X», но слово «система» как принадлежность к типу лучше произнести в уме, если собеседник незнаком с этими типами. А вот для собственного мышления принадлежность к типу нужно отслеживать обязательно!

Успехи сегодняшней вычислительной техники достигаются именно многоуровневой организацией компьютеров — как от уровня компьютерного чипа вплоть до частей транзистора, так и от уровня «компьютера в сборе» (рэка для датацентра) до уровня датацентра в целом. Это ровно тот же рост сложности, какой происходит в биологической эволюции, и он регулируется теми же закономерностями[263]. Успехи электроэнергетики, ракетостроения, транспортного машиностроения, вообще инженерии в целом — они достигаются точно так же, многоуровневой организацией их целевых систем, а также многоуровневой организацией систем-создателей (скажем, завода по производству полупроводниковых чипов, завода по производству ракет и т.д.).

Разделение труда людей проходит по линии систем разной природы, но ещё и разных системных уровней систем одной природы, ибо системы разных уровней создаются/construct/enabling разными «методами работы»/«видами практик»/«содержанием труда» самых разных создателей/constructors/«enabling systems»/ «систем создания» в самых разных ролях.

Инженер, разрабатывающий и производящий чистые материалы для полупроводниковой промышленности, существенно отличается по своим знаниям от знаний инженера по производству полупроводниковых чипов методом литографии. Ещё и мастерство разработки и производства будут различаться настолько, что ими будут заниматься разные предприятия (одни только разрабатывают, другие только производят — и специализация людей-инженеров этих предприятий разная). Это и есть разделение труда. Мастерство этих инженеров, выпускающих чистые материалы, инженеров, которые выпускают из них чипы (интегральные микросхемы) существенно отличается от мастерства инженеров, проектирующих компьютерные рэки и производящих эти рэки. Но все эти разные виды инженеров-технарей и менеджеров как инженеров организаций договариваются друг с другом о том, как из систем более низкого системного уровня изготавливаются системы более высокого, более сложного уровня — включая организацию менеджерами предприятий, которые всем этим займутся.

Системное мышление позволяет не потерять внимание в этих сложных

---

[263] Напомним источники, где всё это можно прочесть в подробностях: прежде всего надо посмотреть фильм Frustrated magnetism in solids в https://en.wikipedia.org/wiki/Geometrical_frustration, затем https://www.pnas.org/doi/abs/10.1073/pnas.1807890115 и https://www.pnas.org/doi/10.1073/pnas.2120037119. С поправкой на то, что генокод не содержится внутри самой целевой системы, сама эта система не размножается, а её делают системы создания/создатели, которые имеют память о целевой системе. Но это мало что меняет в ходе рассуждений.

коллективных переговорах по поводу создания и модификации многоуровневых инженерных систем и сложной сети предприятий, которая эти системы производит. Не будет забыт ни один сток транзистора, ни один вентилятор в рэке, ни один кабель в датацентре, ни одно подразделение, этими системами занимающееся, ни один вид оборудования в этих подразделениях, ни одна категория клиентов, ни одна категория поставщиков, ни один закон тех обществ, где всё это должно заработать и наносить непоправимую пользу самым разным агентам-собственникам самых разных систем.

Моделирование: системные уровни

Отмоделируйте системные уровни какой-то вашей рабочей системы таблично (используйте индексы, например 1.1.3.6 для четырёх уровней). Начните обязательно с надсистемы вашей системы, сама ваша система должна быть не на верхнем уровне, а на втором или даже третьем сверху. Проставьте отметки «да» прохождения простейшего чеклиста на правильность моделирования — это две колонки «Материальность» и Отношение «часть-целое» с системой предыдущего уровня» (если отметка в этих колонках «нет», надо будет исправить модель, так, чтобы отметка стала «да»).

Индекс	Название системы	Материальность	Отношение «часть-целое» с системой предыдущего уровня	Пример эмерджентного свойства, отсутствующего у частей	Создатель (возможно, их будет несколько)

4D системность: паттерны в пространстве-времени, ритмы

Системность — она всегда 4D системность, при этом в 3D у нас системные уровни соответствуют разным масштабам частей традиционно понимаемой физической системы по их размерам (длины — парсеки, метры, нанометры), а повторяемость/patterns описывается формами/shapes. Переходя к 4D, во времени мы тоже говорим о системных уровнях: масштаб времени задаётся «размером» во времени (длительности — миллиарды лет, минуты, фемтосекунды), а повторяемость/patterns описывается ритмами/rhythms/cadence.

Системная ритмика[264] — смотрим не только на классически понимаемый музыкальный и танцевальный «ритм», но и выходим на другие уровни рассмотрения в музыке и танцах: cadence на уровне такта (несколько секунд) и на уровне музыкальной/танцевальной фразы (десяток секунд), крупные формы на уровне произведения (например, запевы-припевы в песнях, кульминации-развязки танцевального перформанса — до минуты), на уровне альбома/концерта или вечеринки — уровень «сета» (двадцать-сорок минут), «большой концертной

[264] См. пример литературы по музыке и танцам (в обратном хронологическом порядке): https://vk.com/wall-179019873_615, https://vk.com/wall-179019873_586, https://vk.com/wall-179019873_505, https://vk.com/wall-179019873_483, https://vk.com/wall-179019873_258, https://vk.com/wall-179019873_237, https://vk.com/wall-179019873_232: Рекомендуем также смотреть работы, посвящённые распознаванию и порождению ритма программами искусственного интеллекта.

формы» (это забытый в музыке после старта стриминговых сервисов уровень, где тоже складывается какая-то ритмика движения какой-то идеи размером где-то в полчаса-час, осталось сегодня это только в виде выкладывания сетов диджеев на mixcloud[265]). Дальше могут идти в музыке и танцах более крупные ритмы — регулярные танцевальные вечеринки и филармонические абонементы (еженедельно и ежемесячно), регулярные фестивали (ежегодно).

В самом ритме выделяем

- функциональные темпоральные части (роли в ритме) и
- конструктивные темпоральные части (из чего делаем, какие появления-снятия/onsets-offsets на каком движении у нас появляются — удары, или ноты, или жесты, или заседания научно-технического совета, или ежеутренняя чистка зубов).

Проще всего разбираться с этим там, где ритмы абстрактны и первичны: музыка и танцы, начиная с «тактового уровня» (у него и названия-то специального нет! Ритмика-по-бытовому игнорирует системность, в литературе по ритмике пока никакой системности, системность туда нужно привносить, это фронтир системной мысли. В какой-то мере это перекликается с текущими разбирательствами с паттернами разворачивания физики во времени, например, временны́ми кристаллами[266]).

Мультидансер::«вид/специализация роли танцора» танцует в значительном стилевом разнообразии. Мультидансер работает с системной ритмикой но ещё и с разными уровнями телесности (буквально: разными частями тела, телом в целом, телом в окружении, парой партнёров в окружении и т.д.), разворачивая это всё на разных уровнях ритма.

Одна из проблем обсуждения в том, что надо учитывать разницу референтных индексов (описание от первого лица — танцора, или от третьего лица — внешнего тренера или зрителя). Тело — это то, что воспринимается снаружи третьим лицом, сома — это тело, воспринимаемое изнутри. Поэтому для мультидансера ритмика не телесная, а соматическая.

Системная соматомеханика (биомеханика — это механика тела/body, как она воспринимается биологом «снаружи», а соматомеханика — это механика сомы/soma/организма) как раз даёт возможность обсуждать движение сомы, то есть движение тела, воспринимаемого и порождаемого изнутри. Конечно, системная соматомеханика работает для самых разных методов/способов/культур/стилей движения — фигурного катания, сидения за компьютером, плавания, борьбы, мультиданса (как раз метод, которым движется танцор-мультидансер). В синонимическом ряду в случае движений чаще всего используются при этом не термин «метод» или «способ», а «культура» и «стиль». Мультиданс при этом включает множество подстилей/субкультур танцевания.

В ритмике нас интересуют паттерны/шаблоны/повторяемости/закономерности изменений тела и сомы танцора, причём изменения на разных масштабах времени, минимально:

- Медленный масштаб начального обучения как «создания мастерства/MVP»,

---

[265] https://www.mixcloud.com/

[266] https://www.quantamagazine.org/first-time-crystal-built-using-googles-quantum-computer-20210730/

то есть передача оснований мемома из культуры. Состояния личности танцора меняются как в ходе создания/изготовления/обучения/тренинга, время создания танцора, у агента изготавливается мастерство по самым абстрактным уровням мемома: принципы «танцевальной композиции»/«универсальных крупных паттернов движения», лексики/«менее универсальных более мелких паттернов движения», «танцевального стиля»/«мельчайших паттернов движения». В профессиональном балете на это даётся примерно десять лет, в социальных/парных танцах (любительский уровень) примерно пару лет. На выходе — MVP мастерства, репликация культурно-обусловленных оснований. Не принято говорить о «ритмике» на таких масштабах времени, но это не означает, что этой ритмики нет. Эксперименты показывают, что чем более длительны интервалы времени (годы), тем хуже люди умеют рассуждать про такие интервалы. Этому умению рассуждать про длинные промежутки времени надо специально учиться.

- Более быстрый масштаб изменения мемома в ходе личного творчества, выработка личного уникального подстиля/манеры, обновление мастерства танцора (smart мутации мемома, получаются путём изменения мастерства танцевания в ходе мутаций мемома, наблюдаемый зрителями и ощущаемый изнутри танцора метод при этом тоже меняется, ибо поменялся алгоритм). Это постоянная деятельность, «развитие мастерства», эволюция. При успехе — появление нового танцевального стиля, если результат этих изменений мемома признаётся как значительно отошедший от культурно-обусловленного эталона. Изменения тут в масштабе полугода-года. Вот такие промежутки времени уже доступны для обсуждения, но работа с ними требует собранности, проявляемой на этом масштабе времени — и это уже не так тривиально, тут не скажешь «соберись!» так, чтобы получить немедленный эффект. Поможет, конечно, письменная культура: письменное отслеживание достижений (в случае танцев, например, можно делать видеозаписи и хранить их).
- Быстрый масштаб изменения фенома (состояния тела и сомы во время задействования мастерства танцевания в ходе перформанса/выхода/исполнения), работа/функционирование мастерства танцора. Изменения тут в ритме музыки — можно за основу самой длинной меры времени взять уровень композиции (скажем, три с половиной минуты танцевального времени), а короткая мера тут — доли секунды между ударами пульса/pulse в музыке.

Эта необходимость рассмотрения разных масштабов времени сегодня проходит красной нитью во всех работах по эволюции, причём подчёркивается, что разница между типичным временем событий на разных шкалах будет отличаться на несколько порядков (и действительно: от долей секунды до нескольких лет — это много порядков!).

Мультиданс::метод/культура/стиль::поведение звучит как существительное (данс, танец), но это стиль поведения (вид поведения), которым занимается роль мультидансера в ходе «работы танцевания» (задействования метода). Мультиданс ещё и составной метод (он разбивается на различные входящие в него более мелкие культуры/стили/методы/способы танцевания, например, танго, сальса, кизомба — в мультидансе будут задействованы все три).

И ещё учитываем, что иерархия самых разных разбиений не должна бы носить разные имена — поэтому называть крупные паттерны «композицией», более мелкие «танцевальным контентом/содержанием/лексикой/фигурами» и ещё более мелкие «танцевальным стилем» и иногда выделять ещё и личную ещё более мелкую структуру как «манеру» — это плохой метод/стиль именования. Если дана какая-то многоуровневая/иерархическая структура, то тип объектов каждого уровня лучше бы оставить одним и тем же, разве что добавлять приставки «под» и «над» (и при этом следить, что речь идёт об отношениях объектов, которые приняты в иерархии, а не о каких-то других отношениях, например, следить, что «под» относится к отношению композиции, а не специализации/рода-вида, так что лучше говорить не «под-», а «вид», скажем «кубинская сальса» это «вид сальсы», наряду с «сальсой Нью-Йорк»).

Число уровней в разбиениях заведомо меняется, отнесение к конкретным уровням — предмет договорённостей в проекте. Что в одном проекте относится к уровню танцевальной композиции, в другом проекте будет уровня танцевального стиля — поэтому в танцах хорошо бы как-то унифицировать в рамках проекта разбиение паттернов движения/поведения/изменения в части имён этого разбиения. Как? Решите это в своём танцевальном проекте, пока нет устоявшихся имён для методов/практик/культур танцевального стека методов.

Всё то же самое можно рассматривать по отношению к любым другим агентам, включая даже агентов в широком понимании, то есть в том числе не очень «умных». Например, какой-то простой автомат стиральной машины, который «ощущает себя изнутри» через замеры с каких-то датчиков уровня и температуры воды, и тем самым у него есть аналог «организма», внешнее поведение «тела»/аппаратуры и его ориентация на работу с «сомой»/аппаратурой-«ощущаемой»-изнутри. Конечно, «измерение величины датчиком» это очень условно относим к «ощущениям», поэтому пишем в кавычках, это антропоцентричная метафора. Ну, а если речь идёт о какой-то организации, то уподобление организации танцору — это общепринятая метафора. Что мы тут привносим:

- **референтный индекс** (реферируем/ссылаемся ли мы в описаниях практики на внешнее тело или внутреннюю сому, иногда называется «позиция восприятия»)
- эволюционно понимаемую **ритмику** (частоты, различающиеся на три порядка в ходе эволюции — норма, помним о работах группы Ванчурина).

Собранность тоже работает с управлением вниманием в 4D, её нужно рассматривать на разных шкалах времени: прямо в споте, «вот сейчас» на несколько секунд, на время длинного рассуждения (от пары минут до пары часов), на время постановки привычки или выполнения проекта (месяц-год), на время удержания длинных периодов времени (несколько лет, планирование образования, планирование занятости в какой-то сфере, выполнение очень длинных проектов типа проектирования и строительства атомных станций). И кроме этих временны́х интервалов нужно отмечать, какие системы, их надсистемы и подсистемы выделены вниманием и на каком системном уровне мы работаем, какие важные характеристики рассматриваем, как и когда (ритм!) запоминаем прерванное рассуждение про одни объекты и переходим к другим. Собранность тоже вроде как существительное, но это метод/культура/практика, которым занимается «собранный» агент (в нашем случае мы отдаём предпочтение инструментальному, а не чисто «мозговому/ментальному» методу собранности, это чаще всего агент-

киборг, то есть человек с компьютером или хотя бы бумагой-ручкой — собранность обеспечивается не мастерством «тренированного на хорошую память мозга», а мастерством использования надёжной внешней памяти, внимание удерживается напоминаниями о важных объектах на внешнем по отношению к мозгу носителе).

В **системном менеджменте** на организацию смотрим как на систему, выделяем там системные уровни в организации (иерархия оргролей, иерархия оргзвеньев), привязываемся к системным уровням целевой системы (разные методы и тем самым разные оргроли работают с разными уровнями целевой системы), разбираемся с ходом создания и развития целевых систем в части ритмичности шагов развития (шагов создания инкрементов и перехода к эксплуатации созданных инкрементов) и сокращения времени таких шагов. В менеджменте тоже приняты всевозможные ритмы (циклы/cadence/ритмы), развёртки во времени на разных шкалах в ходе создания и развития: циклы PDCA, OODA, POOGI и многие другие (рассматриваются в руководстве по системному менеджменту). Ритмичность производства также предмет операционного менеджмента: проход партий/batches сырья через производство и сбыт с целью исключения задержек (сбоев с ритма). Все виды методов менеджмента имеют в своём рассмотрении привязку ко времени и какую-то цикличность/ритмичность/«шаблонность развёртки по времени», и это общее для любой инженерии, не только инженерии организации.

Лидерство (один из методов менеджмента) занимается постановкой агента как актёра в организационную/проектную/трудовую роль и налаживанием сотрудничества (аналог «изготовления, сборки и наладки технической системы из деталей»). В лидерстве внимание удерживаем к организационным системам на нескольких уровнях длительностей времени. Например,

- время внимания в споте, «секунды» (ведение лидерского разговора с агентом, который ставится в роль, со всеми неожиданными поворотами в коммуникации),
- время ведения лидерских бесед как минуты и часы,
- время постановки привычки к действию в роли как обычно недели-месяцы,
- время экзистенциальное как «смены одних ролевых привычек на другие», годы карьерного роста и смены профессий.

Лидерством как методом своей работы занимается роль лидера (агент в роли лидера), лидерство может быть распределённым/shared[267] и тогда надо отслеживать ритмику по отношению к агенту, который становится в роль — но роли лидеров будут исполнять разные агенты, задающие «атмосферу» невозможности уклониться целевому агенту от постановки в роль, а ещё можно 5) учитывать время становления этой «атмосферы» распределённого лидерства, оно может быть весьма существенным — ибо каким-то вариантам метода лидерства надо научить большинство сотрудников.

Но ещё тут и выход на более высокий системный уровень оргзвена=личности в оргзвено=команда и даже оргзвено=«команда команд»/коллектив. Методы тут будут разные для каждого уровня длительностей/ритма/cadence (удерживать внимание на рассуждениях о роли/в роли агента-личности как актёра нужно одним способом/методом/практикой/культурой, а вот работать с командой уже другими, равно как «поставить привычку работы», то есть организовать одного человека для работы в роли нужно одно время и один метод, а для команды нужно и время

[267] https://en.wikipedia.org/wiki/Shared_leadership

другое, и другой метод). Можно ещё увеличить масштаб времени и обнаружить, что там меняются варианты метода лидерства (например, идея распределённого лидерства появилась не сразу, предыдущее понимание было лидерства как единоличности в управлении людьми), это время масштаба эволюции методов, время изменений культуры.

Для каждого системного уровня в организации лидерская работа проводится с опорой на разные методы:

- уровень личной культуры/стиля/метода работы, когда рассматривается один человек (или один AI-агент) — это обучение (с нейросетями сейчас — это prompt engineering). Эта учебная работа с агентами-людьми с учётом их нейрофизиологии, а также похожесть такой работы на обучение AI-агентов подробно разбирается в руководстве по инженерии личности,
- уровень корпоративной культуры, коллектива, работаем с культурами/стилями/методами поведения в группах — в основе там уже менеджерские методы (а в их основе — социология::дисциплина, учение о поведении людей в их группах).

Рассмотрение методов работы и использующих их ролей создателей должно быть системно, то есть многоуровнево, и это уровни как размера, так и длительности:

- **Разные методы работы с под- и над- системами разных системных уровней в целевых системах и их окружениях в ходе их создания (одни роли делают кирпичи, другие кладут стены из кирпичей и раствора, третьи сооружают дома, четвёртые — развивают территории, эти роли разведены по разным агентам, имеющими мастерство в этих методах работы)**
- **Разные методы работают с одними и теми же объектами на разных временны́х масштабах, и тоже могут быть разведены по разным людям. Одни агенты медленно обучают других агентов мастерству (например, три года учат танцевать), а затем эти другие агенты используют это мастерство в совершенно других масштабах времени (например, танцуют три минуты какой-то трек на вечеринке, а ещё задействуют это полученное за три года обучения мастерство многократно, например, двадцать лет)**

Методы мышления, включая само системное мышление, могут иметь свои нюансы на разных масштабах размеров и длительности, но они потому и трансдисциплинарны, что применимы ко всем масштабам и помогают увязывать рассуждения в разных масштабах. Само вот это рассуждение про много масштабов, много уровней размеров и длительностей, много уровней сложности — это и есть рассуждение системного мышления как части интеллект-стека методов фундаментального мышления.

Системность оказывается всегда 4D, то есть паттернированной как в части пространственных форм, так и временны́х последовательностей. **Проектирование в ходе создания идёт в классах/паттернах вещей и ритмах/паттернах изменений вещей во времени. Экземпляры вещей в ходе изготовления-по-проекту и использования тоже существуют в пространстве-времени, это экземпляры вещей, взятые как примеры реализаций не только классов этих вещей, но и ритмических каких-то классов, «форм развёртки**

во времени», шаблонов/паттернов повторяемости во времени — мы обычно говорим об изменениях-во-времени экземпляров как проводимых работах, а изменениях как паттернах во времени — как о методах/функциях. Помним, что рассуждать о времени мы можем только через какие-то изменения каких-то пространственных объектов — обсуждая моменты времени как «события», изменения состояния.

Для чего все это нужно? Системность нужна для того, чтобы победить сложность: представить для каждой практической ситуации не одно суперсложное описание, а много более простых, соответствующих знакомым (классы!) пространственно-временны́м частям (физические объекты в 4D). Формы в пространстве, ритмы смены состояний форм во времени. Когда к этому привыкаешь, разбирательство со сложными ситуациями становится более простым: о самых разных ситуациях думаешь одинаково. И эта «одинаковость» существенно ускоряет мышление, поэтому появляется запас времени на глубокие размышления о всяких неожиданностях и даже на проведение каких-то дополнительных экспериментов.

Если нет одинаковости, неожиданностями будет «всё»! В такой ситуации постоянно непонятно, что делать. Системное мышление сразу задаёт вопрос: на каком системном уровне работаешь и в какой роли каким методом? И ответ должен быть не только на уровне «крупности вещества», но «крупности во времени». Это стандартное размышление: не «что делать вообще», а «что у нас за система, как она должна себя вести в окружении, как она должна быть устроена, из чего и какими методами её делать, какие создатели должны её делать, как создать этих создателей» — это суперкратко выражена всё та же системная мантра. Это типовое размышление, оно будет проходить быстро, и сразу будет понятно, что уже продумано, а что нет, и что делать дальше. Это в каком-то смысле тоже «ритм системного мышления», постоянно крутящийся цикл, паттерн, шаблон мышления.

Внимание по отношению «часть-целое» выделяет паттерны из фона не только в 3D (то есть вытягивает части пространства), причём иерархично на разных масштабах. Внимание выделяет иерархично по отношению часть-целое паттерны из фона в 4D (вытягивает части пространства-времени): кроме форм мы там имеем дело и с ритмами. И это тоже про системы и их поведение, это тоже системное мышление. **Если в вашей жизни ничего не повторяется во времени, значит вы делаете что-то совсем не так, живёте в хаосе. Жизнь и работа — они ритмичны!**

Моделирование ритмики

Возьмите пять видов мастерства, которые отличают вашу личность и оцените эффективность их получения:

Номер п/п	Мастерство в каком методе	Время, затраченное на получение мастерства	Время, потраченное на использование мастерства	Время на применение мастерства, ожидаемое в будущем	Ваш комментарий по эффективности затрат сил на обучение
Пример	Игра на пианино	360 часов	200 часов	20 часов	Ожидания по важности и оправданности

					обучения не оправдались
1.					
2.					
3.					
4.					
5.					

Задания по ритмике

Поставьте отметки о выполнении:

1. Написан пост с эссе о ритмике в вашей жизни. В посте приведены примеры ритма/цикличности в вашей собственной жизни — от секундных и минутных ритмов до недельных, месячных, годичных и многолетних. В примерах описано, какие объекты меняют своё состояние в этих ритмах. В заключении поста раскрыт на вашем примере смысл последних фраз предыдущего подраздела («Если в вашей жизни ничего не повторяется во времени, значит вы делаете что-то совсем не так, живёте в хаосе. Жизнь и работа — они ритмичны!»).

Системные уровни выделяются вниманием. Пример методов телесной работы

Люди плохо думают о «поведениях», плохо выделяют вниманием части и целые в динамических (меняющих свои состояния во времени) системах, требующих 4D представлений о частях в пространстве-времени. Опишем методы/культуры/стили и роли, используемые сообществом танцоров, практикующих **социальные/парные танцы**[268] как типичный пример.

Работа «танцевального выхода»/перформанса/«исполнения танца»::метод интересна тем, что существует всего несколько минут. Метод этой работы исполняется артистами::роль, в которой танцор::«подроль артиста». В нашем случае артист (с задействованной подролью танцор) — это исполняющая метод проведения танцевального выхода подроль «члена сообщества социальных танцев»::«культурный человек»::человек, пример других подролей ролей людей-членов сообщества социальных танцев — организатор мероприятий, диджей. Можно легко добавить ещё один системный уровень, например, взять не всё сообщество социальных танцев, но ещё и дополнительный уровень сообщества какого-то стиля социальных танцев, например, сообщество сальсы. Повторимся: в этих сообществах/тусовках не только танцующие артисты, но и организаторы мероприятий, фотографы, видеографы, диджеи, учителя — причём и «артистами» тамошних артистов не называют, если только они не профессионалы, зарабатывающие своим танцевальным творчеством. Коллективно выполняемый ими стиль/культура поведения сообщества этого стиля — субкультура, для сообщества всех социальных танцев — иногда говорят про эту культуру «субкультура», иногда и просто «культура», всё зависит от проекта, для которого ведётся моделирование/«присвоение типов для важных объектов предметной области». Себя артисты::роль предпочитают называть «танцоры», но это характеризует их

[268] https://ru.wikipedia.org/wiki/Социальные_танцы

общетанцевальную подготовку, мы увидим дальше, что «танцор» тут находится на довольно глубоком системном уровне, выше «танцора» ещё много самых разных ролей.

Системное мышление о танцевальном сообществе или даже отдельном артисте::роль сложней, чем системное мышление о каком-нибудь электронном гаджете. Гаджет можно физически собрать-разобрать из конструктивных частей, выделяемые вниманием функциональные части гаджета хотя бы примерно соответствуют конструктивным частям этого гаджета, то есть деталям и их сборкам. Артиста или более узко артиста с подролью танцора (это системы, названные по их роли в окружении, а не по конструкции агента «человек» или «робот») невозможно собрать-разобрать в уме в форме взрыв-диаграммы/«explosion diagram/schema». Проблема в том, что «взрыв-диаграммы» — это не выделение вниманием функциональных двигающихся и меняющих своё состояние в ходе работы частей, а прямо таки физическое разбиение (хотя оно может проводиться в уме) конструктивов, о котором думают обычно во время создания системы, а не во время работы системы. Вот пример взрыв-диаграммы велосипеда, на диаграмме не функциональные части, а именно конструктивные части[269]:

Если таким образом представить себе организм человека с его функциональным разбиением на органы::«роли в организме», то сразу видна ошибка: такой «физически разобранный организм из конструктивов» нельзя представить живым-работающим: он будет мёртвым в таком виде, «время разборки»! По взрыв-диаграмме нельзя объяснить «как работает», изменения::поведение в ходе работы. Для показа изменений надо как-то отображать потоки вещества или энергии,

[269] http://solexclubspirales.free.fr/eclates.html

разность потенциалов. Для этого используются «функциональные диаграммы»/«принципиальные схемы», на которых между функциональными частями что-то течёт (жидкость, электроэнергия, механическая энергия, тепло), производя изменения в функциональных частях. Об этом подробней будет чуть дальше.

Проблема в том, что взрыв-диаграмма часто неосознанно появляется в уме при функциональном рассмотрении, как неосознанное представление функциональной декомпозиции. Это грубая ошибка системного мышления. Обратите внимание, конструктив велосипеда на картинке не едет, невозможно описать его движение. Вот конструктивные части часов, изображение взрыв-диаграммы:

MANUFACTURE ROGER DUBUIS
SWISS MADE
JEWELS

Такие часы не тикают. Но нам-то надо как-то обсуждать системы прежде всего во время их функционирования/эксплуатации, и выделять в них совсем другие части — это можно сделать только вниманием. Верный способ: попытайтесь представлять системы в ходе их работы не вот такими статичными картинками, а в виде коротеньких видео, «гифок»/.gif — даже если во время работы/функционирования системы нет движущихся частей всё равно представляйте потоки тепла, электричества.

Про артиста::роль (певца::подроль, танцора::подроль, жонглёра::подроль, в том числе подроли определённого жанра/стиля соответствующей художественной/артистической культуры/деятельности/метода певца, танцора, жонглёра и т.д.) мы думаем в ходе его выступления/выхода/перформанса, и его функциональные (то есть во время работы/использования) части приходится выделять вниманием прямо «по ходу работы», а не разбирать. Разобрал на конструктивные части в реальности, а не вниманием — всё, системы нет, для неживых систем она ещё «в изготовлении» или «уже разобрана», а для живых систем — «уже мёртвая».

Для системы есть только один способ думать о её функционировании — выделять функциональные (а не конструктивные) части используемой/эксплуатирующейся/работающей/функционирующей системы вниманием в ходе её работы, а не мысленной, даже и внимательной сборкой (разборкой) пока ещё (или уже) «мёртвой» системы в момент её создания (или уничтожения).

Вот картинка социальных танцоров, в которой даже нужного нам отдельного танцора приходится выделять вниманием, и это может быть даже не первый попавшийся танцор (который только отвлекает, ибо смотрит прямо на нас, да ещё на переднем плане, да ещё и показан очень резко). Нет, нам нужен тот, кто нужен — например, танцор, который взят в красную рамку:

Если нас интересует функциональное рассмотрение, то надо переходить в 4D — увидеть это не статичной картинкой, а добавить к пространственным трём измерениям ещё и измерение времени: проиграть в уме видео, хотя бы и короткое.

А теперь представим, что ещё звучит громкая музыка, вас время от времени толкают, все танцоры довольно быстро перемещаются, а внимание ваше должно переключаться не только на отдельных людей, но и как-то выделять части этих людей (не разбирать на части, а выделять части вниманием!). Мышление в этой ситуации наблюдения за танцующими людьми-артистами мало отличается от того, что происходит в какой-то организации как системе-создателе. Мышление для описания работы ткачей в лёгкой промышленности (создатели тканей) или инженеров-атомщиков (создатели атомных электростанций) будет устроено примерно так же, как мышление о танцорах: шанса представить взрыв-диаграмму не будет, придётся представлять что-то движущееся/меняющееся и выделять это «что-то» вниманием.

Подробней о том, как обсуждать работающие системы с людьми, будет в руководстве по методологии, но уже сейчас можно заметить, что разговор о методах для живых людей, как ни странно, помогает разобраться с выделением системных уровней (определяемых функционально, во время использования/работы системы) и для неживых систем. Когда новичка инженера-менеджера просишь представить функциональный двигатель автомобиля во время работы этого двигателя::подсистема в едущем автомобиле::система, то он невольно представляет у себя в голове именно взрыв-диаграмму, и это ошибка уже с первой секунды размышления о функциональных/ролевых частях системы. А вот разобраться с примером такого мышления на людях быстрее как раз потому, что их трудно разобрать «в уме» на составные конструктивные части.

Роли/«ролевые объекты»/«функциональные части» людей-агентов, реализуемые конструктивными/материальными частями организма и работающие под управлением «программы выполнения метода»/мастерства::система (с учётом мастерства задействования инструментария метода), получаются не «сборкой конструкции», а обучением. У младенцев не так много человеческого в поведении. Врождённых ролей у младенца практически нет, младенца можно назвать «личностью» только в общефилософском смысле, а интеллект младенца крайне ограничен, вменяемость минимальна (ибо нет возможности понимать и произносить речь), методологический кругозор нулевой. Первые годы у людей быстро растёт интеллект (в том числе вменяемость/persuadability — вместе с появлением речи, вместе с активированием режима медленного мышления S2, дающего возможность понимать и принимать аргументацию), а затем познаётся/learn множество самых разных ролей, превращая «почти личность» ребёнка в полноценную взрослую личность.

Напомним: агент-человек состоит из (отношение композиции) организма и личности, где организм — это hardware/аппаратура универсального создателя/constructor (и вычислителя/computer в составе создателя), а личность — это софт/программы/«набор всех видов мастерства». Личность даёт агенту мастерство выполнять те или иные методы, то есть проявлять видимое культурное («по известному в человеческой культуре методу») поведение.

Артист и танцор проявляются как роли в момент танцевания, они распознаются по их поведению в каком-то агенте при помощи исключительно внимания. В

агенте::создатель есть программа/софт-на-нейроносителе/мастерство выполнения культуры/стиля/практики/метода художественного выступления (для роли артиста) и танцевания (для роли танцора).

Вы сразу отличите танцора во время танца от слесаря или землекопа во время слесарных или земляных работ — по «проявляемому поведению»/«способу/методу выполнения работы»/«паттерну поведения». Метод/культура/стиль поведения Васи в ходе (работ) танцевания зрительно сильно отличается от метода/культуры/стиля поведения Васи в ходе слесарных или земляных работ. Помним: роль исполнителя какого-то метода работы мы распознаём через наблюдение выполнения работ по этому методу (наблюдается изменение работами состояния важных объектов) каким-то агентом, знакомства с рассуждениями агента о его методе и демонстрации тем самым предметов интереса этого метода, другого пути нет — что будет делать агент (каким методом он будет работать) из роли чаще всего невыводима, а вот роль из метода работы — выводима. Скажите, что агент (человек, робот, предприятие) делает и говорит/пишет — и можно будет сказать, к какой культуре поведения он принадлежит, какая у него роль. Роль часто получает своё название-термин от имени метода её работы, от ожидаемого паттерна поведения этой роли. **Это мы обсуждали, когда говорили о мантре системного мышления: сначала «в каком окружении что делает» для целевой системы, потом «каким способом/методом создаёт», а уже потом — «кто делает» для роли создателя и агента-актёра этой роли.**

Совместное танцевание двоих танцоров::роль в «социальных танцах»::культура/стиль/практика/«метод поведения» относительно простое по сравнению с поведением команды каких-то агентов-людей в проекте создания и развития какой-либо системы и тем более в организации (каком-то предприятии), где одновременно ведётся множество проектов. Почему мы пишем «метод поведения», а не «метод работы»? Мы используем более абстрактный тип для «работы» просто потому, что как-то не принято социальные танцы считать «работой» в смысле job, а метод — это way of working, в русском языке тут легко перепутать «работу по методу как задействование ресурсов» и «работу как занятость на должности». Но мы ещё можем и учесть то, что говорим не о «танцевальных методах работы», а о «танцевальной культуре» — культура как раз будет методом работы, который распространён в каком-то сообществе. **Важны не использованные слова-термины, важен донесённый смысл, важно понимание! Но кроме слов для донесения понимания у нас ничего нет, так что слова-термины тоже важны!**

В культуре/методе социального танцевания обычно танцуют двое танцоров, «танцевальная пара», иногда социальные танцы даже называют «парными танцами». Танцевальная пара — это два танцора с заранее известными специфическими ролями (лидер/ведущий и фолловер/ведомый/следующий, часто «партнёр» и «партнёрша», но это танцевальные роли, а не гендерные, да и иногда бывает танец на троих — double trouble, один лидер и два фолловера[270]). Лидер и фолловер даже не разговаривают друг с другом словами, хотя в некоторых танцах могут и разговаривать, например в «Caroline shag»::стиль/культура фолловер

[270] https://vk.com/video52259628_456244076 и https://vk.com/video73069173_456240091, https://vk.com/video-18525054_456240224, методика double hustle https://1drv.ms/b/s!Ag1bdTHZ9O7JgQ84GsQZqL_x8m9V

может объявить прямо голосом фигуру, которую должен станцевать лидер.

При этом лидер и фолловер имеют понятные друг другу предметы интереса (им обучают в школах танца), они играют в ходе перформанса в «понятную ролевую игру»[271]. В большинстве командных проектах по созданию и развитию систем, и тем более в каждом предприятии со множеством проектов, людей с разными ролями больше двух или даже трёх, при этом методы/деятельности/культуры/практики этих людей существенно разнообразней, а взаимодействие — сложней.

Системно думать и о проекте, и о предприятии — это прежде всего вниманием выделять тамошние роли/«функциональные части» как выполняющие какие-то функции/методы/практики/деятельности. Для этого удобно использовать выход/исполнение/переформанс социального/парного танца как переходную по сложности систему.

Танцор::роль — даже не верхнеуровневая подроль участника сообщества социальных танцев. Танцор — это роль того агента, который выполняет танцевальные, а не бытовые движения, будь то социальные, сольные или ритуальные танцы (например, танцы шаманов). Всё выступление в целом делает артист::роль, который выполняет действия по методам (это означает, что этим методам его хорошо бы научить, а не ждать, что он эти методы как-то прихватит сам из «насмотренности»):

- уместности своего «выхода»/выступления («был на похоронах, там играла музыка, я не растерялся — потанцевал»)
- отслеживания реакции зрителя
- выбора соответствующей стилю танцевания одежды,
- реализации артистической/художественной идеи,
- начала и конца танцевания (возможно приглашение партнёра, проводы партнёра с танцпола на место в зале, настройка на конкретного партнёра в ведении, распознавание окончания вступления и конца трека в музыке, и т.д.)

Подролью/подсистемой артиста будет танцор, который и выполняет танцевальные движения, это более узкая роль, более мелкая часть личности (помним, что личность реализуется даже не столько телом, сколько «программой в мозгу». Мастерство (аналог компьютерной программы, реализованной мозгом-вычислителем) танцевания::культура/«метод/способ действий» для исполнения роли «танцор» реализуется меньшей частью мозга, чем общая программа выполнения действий по методам «художественной деятельности» надролью «артист».

Мы не знаем точно, как реализуется «мозговая программа» в мозге, но условно можно считать, что «по размеру» больше нейронов занято всем мастерством художественной речи, пения, музицирования/«игры на музыкальных инструментах», пантомимы, танцевания, рисования и т.д., то есть занято всей артистической частью личности, чем только мастерством танцевания, танцевальной частью личности. Артист::роль не только умеет танцевать::«метод выполнения танцевальных движений под музыку», но он ещё много чего другого умеет — артистическая деятельность явно не сводится к только танцам. Скажем, артист театра или кино обычно танцевать умеет — но на сцене или в кадре танцует редко.

---

[271] Социальный танец может пониматься как «игра в загадки», https://vk.com/wall-179019873_1550

Представим системное рассмотрение культуры/«методов действия агента» социального танца в момент работы/функционирования танцора (помним, что **метод работы в момент его воплощения в физическом мире, то есть когда какой-то физический/конструктивный объект-агент выполняет роль, отыгрывающий этот метод/способ/стиль/вид труда — это работа/действия/функционирование**).

Вот табличка системных уровней как уровней ролей/«функциональных/ролевых частей» члена сообщества социальных танцев, которая примерно поясняет тут рассказываемое (а что написано в заголовках таблички, будет понятно чуть ниже по тексту руководства)[272]. Можно также рассматривать эту табличку как стек танцевальных методов/культур/практик. В руководстве по методологии будет подробно рассмотрена и эта табличка, и ещё примеры подобных табличек — чтобы лучше понять принципы описания методов.

Культура	Роль адепта	Учитель (создатель)	Составляющие методы и их предметы методов	Конструктивы	Мастерство
Общая культура	Культурный человек	Деятель культуры			
Танцевальная субкультура/контркультура (танцевальная жизнь)	Неформал/«занимающийся»/член сообщества	Амбассадор/организатор сообщества	Ритм (паттерны во времени) культурной жизни социального танцора. Удержание стиля жизни "неформала" по аналогии со стилем в танце. Хип-хоп: peace, love, unity, having fun. Социальные танцы – соотношение между формальным обучением в школе и челленджах, занятиями сампо, походами на вечеринки, участием во флешмобах/ралли и буткемпах, участием в конкурсах.	Тусовка/сообщество	
Участие в вечеринке/соревновании/шоу	Участник мероприятия	Орг/концепт мейкер	Траффик (придерживаться), сет/танда (танцевать), приглашение, окончание танца, танцпол/сцена/подиум/круг, амфитеатр/столики/стулья/зрит.зал (учитывать). Музыкальный/стилевой формат (использовать), гендерный баланс (выбор танц.роли и времени перформанса), уровень танцпола (учитывать возможности перформанса).	Покрытие пола, саунд система, зона бара/чиллаут, зона на краю танцпола. Команды/пары	Прикладное танцевальное мастерство
Культура перформанса/выступления/выхода/танды	Артист	Хореограф/режиссёр	Последовательность [импровизационная или поставленная] фигур в одном или нескольких танцевальных стилях (создавать, исполнять). Место танца [слот, ронда, линия, хоровод] (занимать).	Исполнитель танца (или группа/пара), костюм, реквизит	
Танцевальный стиль	Танцор стиля (би-бой, сальсеро) или мультидансер	Гуру стиля действующий или исторический	Фигуры/степы/элементы стиля: паттернированные перемещения шагами в ритм (исполнять, модифицировать, нюансировать, композировать). Стайлинг: паттерны постур, украшения (придерживаться, применять каноны)	Тело в различных позах (сочетание положения танцевальных частей тела и реквизита)	
Танцевальное движение (танцевание)	Танцор	Учитель танцев	Танцевальные движения: конструкторы шага, ритма, украшений, постуры, вращения, ведения, верёвочек руками, положений в паре; понятие нюансировки (составлять, контролировать, исполнять, удерживать, сочетать).	Танцевальные части тела (40 танцевальных ног) и реквизит (веер)	Общетанцевальный интеллект (готовность к новому танцевальному стилю)
Общее движение, body control (соматодинамика)	Движенец (так называют в некоторых сообществах)	Тренер	Двигающиеся массы части или всей связной сомы (разгон, торможение, движение по инерции///импульс, скорость) и тела, путь (пройти), ось (держать) [сома – настоящая чувствуемая изнутри физика, тело – виртуальная воспринимаемая снаружи физика]	Массивные части тела и экзотела (включая инструменты/реквизит и окружение)	Телесный интеллект (готовность к новой культуре движения)
Ритмика (движение вовремя)	Ритмик	Ритм-инструктор	Пульс, доли, метры, ритмы, onset (что-то происходит вовремя), offset, inter-onset interval, полиметры, полиритмы. Жанры/стили в ритмике. Сольмизация (такадими, думтаки).	Моменты во времени, события как изменения.	
Усилия в масштабах тела (соматостатика)	Телесный практик	Инструктор-телесник	Сома (структурировать), лента (усилить-ослабить, удерживать): усилие в диапазоне, равномерность усилия по всей длине, равномерность усиления-ослабления на скорости в диапазоне, фон ленты (чистим), пятно опоры (определяем) микрошевелёнка (вносим).	Опорно-двигательный аппарат (адаптируем к нагрузкам и скоростям)	
Усилия и амплитуды в частях тела	Физкультурник	Фитнес-инструктор (практика фитнеса)	Ансамбль (включаем, выключаем, держим, составляем), точка опоры (собираем),	Части опорно-двигательного аппарата (вокруг скелетных костей: кости, суставы, ткани вокруг), области тела	
Жизнедеятельность	Здоровый человек	Физиолог (практика лечения)	Организм в целом и его органы: мышцы, кости, фасции, нервы	Мышечные пучки, волокна, …	

Добавим для наглядности художественных деталей («рендеринг», как в компьютерном 3D-моделировании, когда добавляют эффекты освещения и помещают какие-то дополнительные предметы в картинку из САПР/CAD, «для живости восприятия»). Это нужно для «заземления», представления объектов в физическом мире.

В центре Москвы в одном из клубов с 22 часов 13 минут до 22 часов 18 минут играет громкая музыка, царит полумрак и двое «неформалов»/«занимающихся социальными танцами»/«членов субкультурных сообществ» в своих подролях «участник мероприятия» танцуют::работа сальсу::«стиль танцевания» посреди клубного танцпола. Почему нам так важно указать время и место? Чтобы было понятно, что речь идёт о физическом объекте, который существует не только в пространстве, но и во времени — и мы рассматриваем экземпляр исполнения культуры (метода/way of working, но поскольку тут сообщество — культура, а также ещё будет рассмотрен и стиль) какими-то «конкретными ресурсами, занятыми конкретное время», то есть работу/work как она понимается в операционном менеджменте. Это неважно, что предметная область не такая, чтобы в ней

[272] Табличка в полном разрешении есть в https://disk.yandex.ru/i/QbcxOA1uuf5AAA

привычно звучали слова из инженерии и менеджмента. Важно показать, что рассуждения ведутся одни и те же для любой предметной области — сила интеллекта как раз в этом. Системное мышление позволяет детально обсуждать самое важное, что происходит в самых разных предметных областях, оно для этого и предназначено.

Мы понимаем, что танцоры::роли наши вполне физичны, но дальше будем обсуждать культуру танцевания как подкультуру общей культурной жизни. В просторечье эта культура называется «танец», но мы не будем использовать это слово-термин, ибо «танец» — это и «культура танца в целом», и «исполнение/выход/перформанс танца», и «танцевальная работа». Сравните: танго::танец — это может быть и танцевальный стиль, и сообщество людей, практикующих танго, и отдельное исполнение танцевального стиля танго. Когда говоришь «танцевание танго» — сразу понятно, что речь идёт о поведении (функция/практика/культура/стиль/метод обычно именуется отглагольными существительными, в английском — -ing-формой глагола).

Физиолог::роль поглядывает на эту пару и явно доволен: партнёр::«танцевальная роль» не хромает, исполняющий эту роль человек **здоровый::«состояние организма агента»**. Все кости, мышцы, связки, фасции и т.д. целы, этот врач::«надроль физиолога» (употребляем «врач» чисто для красоты слога: как «тигра» называть более общим классом «зверь») недавно починил (терапия — это в инженерии «ремонт») всё, что касалось системного **уровня физиологии**. Остальных::роли в клубе это не волнует, но врача-физиолога (читать как «зверя-тигра») до сих пор волнует: он ищет свидетельств в наблюдении за танцеванием, что мышцы-кости-связки и прочее, чем он занимался в ходе лечения партнёра, работают как надо. Вроде как всё ОК! И врач уходит с вечеринки. Врачу даже неважно — это была сальса::стиль или танго::стиль, собственно танцевания::«метод танцевального движения под музыку» он и не заметил, слишком высокий системный уровень для его мастерства физиолога, для него это не «танцевальное движение», а «телесное движение».

Каждым системным уровнем мастерства как части личности участников этой танцевальной вечеринки (подролями участников, «функциональной декомпозицией» личности) занимаются в части создания и развития свои деятельностные/трудовые роли. Но на нижних уровнях врач::создатель и тренер::создатель занимаются не только и не столько личностью (мастерством) агента как целевой системой, сколько телом/организмом агента (хотя и врач и тренер ещё и учат мастерству работы с собственным телом: врач учит спать и есть правильно, тренер учит заниматься движением правильно — но следит и за тем, чтобы менялась «аппаратура», то есть увеличивалась сила мышц, появлялась гибкость суставов — и не только за счёт мастерства управления телом, но и за счёт биологических изменений тканей под воздействием тренировок).

Если мы хотим, чтобы универсальный агент (универсальный в смысле universal constructor, «программируемый на любые задачи преобразователь физического мира») хорошо работал как артист, в проекте его создания и развития должны хорошо отработать самые разные роли создателей. Инженеры делают свои системы командой. Людей тоже должны «делать» (обучать) команды! Тут речь идёт не только о каком-то разделении труда между подролями учителей, специализирующимися на обучении (в руководстве по инженерии личности будем говорить о культуртрегере, методологе, методисте, преподавателе,

декане/администраторе), но ещё и о ролях учителей (создателей мастерства), специализирующихся в прикладной (в определённой узкой предметной области) области того целевого мастерства, которому учим (например, мастерства танцевания).

Чтобы танцевать, кроме танцевального мастерства нужно ещё иметь здоровое (с точки зрения физиологии) тело. Если тело не здорово, своим делом должен заняться физиолог. Физиолог — это подроль врача, а исполнять роль врача может хоть сам агент, исполняющий и роль танцора, и какой-то другой нанятый им для исполнения роли врача агент с нужной степенью квалификации во врачебном деле. Врача не волнует сам танец как таковой. Партнёр (с точки зрения врача — это просто «здоровый человек, бывший пациент»::организм, его танцевальная роль неважна) движется? Движется. Ему не больно? Не больно. Он ничего себе этим движением не повредит? Не повредит. Всё, кости-мышцы-связки целы, и врач может предоставить тело другим специалистам, которые займутся другими делами, помогут мастерству танцевания дать его инструмент — опорно-двигательный аппарат (кости-мышцы-связки-фасции и всё то, чем занимается обычно врач).

Рассмотрение этой работы с организмом::аппаратура для мастерства::софт мало отличается для танцора и музыканта: в случае музыканта тело::организм нужен чуть меньше, чем танцору (редко музыкант работает ногами, чаще всего только руками, а в духовых инструментах — лёгкими и губами — но эти части тела тоже работают как «оборудование»), а ещё к телу добавляется дополнительное «оборудование», (музыкальный) инструмент (например, рояль или тромбон). Легко представить себе ремонтника и настройщика рояля, который ремонтирует и настраивает рояль пианисту:: «подроль музыканта» примерно так же, как врач и тренер тело «танцора сальсы»::«подроль танцора». Более того, ремонтник и настройщик рояля может даже не уметь играть, и не различать классическую и джазовую музыку! Хотя может и уметь, как врач или тренер могут уметь танцевать. Но это не обязательно. Более того, когда «инженер по САПР/CAD» устанавливает и настраивает «программу САПР/CAD», то эта ситуация полностью аналогична для инженера-проектировщика работе врача с телом танцора, работе ремонтника и настройщика рояля с роялем музыканта. В состав тела агента мы включаем часто не только его собственное тело, но и оборудование как экзотело. В состав мозга мы можем включать не только неокортекс, но и экзокортекс, а в состав тела — одежду, транспорт, инструменты. Отнять мою рубашку — это всё равно, что отнять продолжение моего тела, частная собственность священна! А если в качестве оборудования я использую завод (капитал! Средства производства!), то это тоже моя частная священная собственность, продолжение моего тела.

Тренер::роль (мы включили в эту роль и подроль инструктора-телесника и подроль ритм-инструктора, и даже подроль фитнес-инструктора — все их будет выполнять один агент в нашем примере, но «разделение труда» как раз о том, что исполнение каких-то подролей крупной роли может выноситься вовне, исполняться другими агентами) внимательно смотрит на эту же танцующую пару и видит двух **движенцев**::роль (так обычно называют себя люди, которых интересует телесное движение как таковое, вне привязки к виду спорта, или танцам, или паркуру). Он выхватывает своим вниманием **общие движения** пары (вращения, приседания, балансы и падения, шаги) — насколько они управляют телом (body control), то есть насколько движения тела равномерны, насколько удерживается в этих движениях равновесие, насколько неразболтанно выполняются ускорения и торможения в

движениях. Если в движении видны какие-то проблемы (теряется форма, отклонения от ожидаемой траектории), то **инструктор-телесник**::(подроль тренера) смотрит на внешние признаки того, что нагрузки в теле равномерно распределены: усилия в функциональных мышечных органах-лентах, соразмерных всей длине человеческого тела (от точки опоры до точки контакта с другим человеком) минимальны для выполняемых движений и равномерны. Если там проблемы, то внутри этого же организма танцующего человека **фитнес-инструктор** отслеживает, не перенапряжены или не перерасслаблены ли какие-то отдельные части тела — колени или шея. Мы перешли уже на другой системный уровень, где тело здоровое, но нам нужно поставить его под контроль сознания/body control: если танцор хочет передать движение от точки опоры аж до другой точки опоры (да ещё эта точка опоры может быть в другом теле! Танец-то парный), ему нужно иметь весьма специфическое внимание к телесным ощущениям напряжения и расслабления в соме (сома — потому как тренер воспринимает тело снаружи, а танцор воспринимает тело изнутри). А ещё тренеру нужно добиться в теле/организме агента-танцора правильной структуры опорно-двигательного аппарата (мышц-связок-фасций-костей). Танцору эта правильность позволяет провести танцевальное движение без лишней суеты, толчков, шевелений для удержания равновесия, прицеливаний и метаний перед приходом ноги на пол или руки в какую-то точку. Но мы пока даже не говорим о танцевальном движении, тренер работает с телесным уровнем, пока оторванным от специфики задействования тела в каком-то культурном/стилевом (бытовом, спортивном, танцевальном) движении.

Агент, совмещающий роли тренера, ритм-инструктора, инструктора-телесника и физкультурника (но помним, что может быть и разделение труда, никакого совмещения всех подролей в одном агенте, может действовать «тренерская бригада»), конечно, готовит «управление телом»/body control и занимается свободой сознательного движения не только для танца. Управление телом и свобода сознательного движения как предметы интереса важны и для ролей спортсменов-легкоатлетов, и для ролей музыкантов (пальцами шевелить при игре на музыкальных инструментах — мышечная работа!), и для ролей, занимающихся единоборствами, и для ролей, занимающихся паркуром. Но тут он сработал для танцоров, и он доволен: тела танцоров вполне свободно двигаются. То, что эта свобода движения ещё и обеспечивает качественное ведение-следование в паре[273], тренеру неважно, собственно танцевальное движение мы тут не обсуждаем. Конечно, тут рассматриваем движение именно танцоров, но всё-таки свободно и управляемо двигающееся тело может быть использовано в самых разных целях — в этом-то и прелесть системного мышления, что проблемы каждого системного уровня можно обсуждать в том числе в отрыве от проблем других системных уровней. Мастерство управления телом, да и само тело — это нижние системные уровни агента-танцора. Обычно «нижний системный уровень» означает меньшую специализацию: атомы неспецифичны, они потенциально могут участвовать как мельчайшие частицы в огромном количестве других систем, равным образом хорошо развитое тело и поставленное управление этим телом потенциально могут участвовать в самых разных других системах — вокалистах (пение — это мышечная работа), программистах квантовых компьютеров (сидеть восемь часов на рабочем

[273] видеоролик, где очень кратко рассказывается о том, как именно уровень телесных практик ответственен за то, будет ли танцор студнем, тяжёлым или лёгким буратино, или же всё-таки танцором: https://vk.com/wall-179019873_511

месте — двигательная активность, хорошо бы, чтобы после этих восьми часов ничего не болело и не было усталости в теле), танцорах, спортсменах и так далее.

Выделение системных уровней вниманием. Пример социальных танцев

И только на уровень выше появляется **танцевание/танцевальное движение**, когда **учитель танцев** (создатель мастерства танцевания в учениках) смотрит, как двигаются его ученики-**танцоры** (носители мастерства танцевания) — принимают эстетичные позы, осуществляют мягкое ведение и чуткое следование, не теряют баланс на поворотах. Предыдущие роли (фитнес-инструктор, ритм-инструктор, тренер) не видели танцоров, они видели просто двигающихся людей, они видели физкультурников (их создатель — фитнес-инструтор), телесных практиков (их создатель — инструктор-телесник), движенцев (их создатель — тренер и помогающий ему ритм-инструктор) — неважно, что там за тип культурного движения, это мог бы быть и паркур, могло бы быть и фигурное катание. Тут же речь уже идёт о специфическом типе культурного движения: **танцевальном движении** (а не движении в дзюдо, конном спорте или картинге).

Но на этом уровне ещё неизвестно — какого именно танцевального стиля/культуры это движение. Танцевать какой-то стиль (танго, балет, сальса) учить долго. Дают разные оценки для разных танцевальных стилей и для разных способностей танцоров, но вот балету вполне могут учить десяток лет, хип-хопу лет пять, социальным танцам — года три. Мы просто выносим за скобки, выделяем вниманием общий метод движения для всех танцев, отличающий их от всех других культурных методов движения, считаем, что танцевальному методу движения можно научить один раз — использовать далее во всех «танцевальных стилях»/«видах танцевальных методов движения». Например, нужно научить танцевать в паре, то есть уметь вести и следовать, синхронизировать своё движение с движением партнёра. Это нужно во всех стилях социальных танцев (впрочем, и не только социальных, танец в паре может быть и бальный, и эстрадный, и даже в балете[274]). Можно научить не просто «шагать», а делать специфический **танцевальный шаг**, у него есть своя специфика (он содержит в себе встроенную «иллюзорность», необычность/нереальность). Это тоже нужно для всех танцев, учиться этому нужно один раз, а затем использовать во всех танцах!

«Вынесение за скобки» обучения танцеванию как таковому в отличие от обучения танцевания какому-то конкретному стилю танцев — это и есть идея мультиданса::«метод танцевания». Наш танцор::роль — это мультиданс-танцор, он уже обучен достаточно, чтобы легко дальше осваивать самые разные танцевальные стили: у него поставлено управление телом в ходе общего движения, у него освоено общетанцевальное движение, и нужно просто дальше приобретать мастерство какого-то танцевального стиля.

Так что дальше за танцеванием нашей пары наблюдает **учитель сальсы::**«гуру конкретного танцевального стиля»**.** Он смотрит, как переступают в ритм ногами его ученики-**сальсерос** (те, кто танцует сальсу, один такой — сальсеро, двое — сальсерос, это множественное число в испанском языке). Учитель сальсы смотрит на ту же пару, в тот же момент времени, что и физиолог, что тренер, что учитель танцев общетанцевального уровня. Но учитель сальсы видит пару совсем другими глазами (то есть у него другие предметы интереса). Он следит за качественным

[274] https://en.wikipedia.org/wiki/Partner_dance

исполнением метода **«стильного танцевания»/«удержания танцевального стиля»** (танцевания с соблюдением конкретного стиля танца, в данном случае — кубинской сальсы), и его предметы интереса: какая постура (структура усилий в теле при сохранении характерных поз), как танцоры попадают в ритм музыки, перенося вес с ноги на ногу и совершая при этом довольно специфические движения именно сальсы (а не танго, не бальбоа, не кизомбы), как одеты танцоры, как они передвигаются по танцполу (стоят на месте, или «гуляют»). Это и есть танцевальный стиль: то, как шагают танцоры под музыку в конкретном танце, независимо от метода выполнения конкретной «танцевальной фигуры»/па/pas/степа/step/движения/move/«лексической единицы».

Так, в сальсе не столько лидер толкает фолловера на шаг, сколько лидер отталкивается от фолловера — и оба «отлетают» от точки контакта на примерно равное расстояние. Танец не сводится к тому, что лидер крутит вокруг себя фолловера, рисунок танца более-менее симметричный в плане движения обоих партнёров — оба проходят в танце примерно одинаковое расстояние. И это происходит во время всей длительности перформанса сальсы, в каждом движении, при огромном разнообразии танцевальных фигур — стиль соблюдается независимо от танцевальных фигур. Стилю (иногда говорят, «базе») танца можно учиться долго. Шагать можно как в армии, одинаково на каждом шаге, даже танцевальном шаге, а можно делать нюансы — и в простых шагах при хорошем владении базой открывается целый мир[275] разнообразия в шагах. На этих нюансированных шагах сальсеро строит множество движений себя и партнёрши, характерных именно для сальсы — вращений, перемещений.

Гуру сальсы смотрит, как наша пара исполняет показанные им фигуры/па: крутятся куда надо, шагают куда надо, в том ритме и при соблюдении той постуры, которая должна быть в сальсе в отличие от, например, танго или паркура с фигурным катанием.

Метод стильного (с удержанием специфического танцевального стиля/культуры движения) танцевания сальсы включает специфические па/фигуры/steps/moves (например, enchufla, sombrero[276]), которые исполняют танцоры, превращающиеся тем самым в сальсерос. Сальсеро (роль танцующего именно сальсу::стиль агента) — это надроль танцора. Это именно надроль, то есть «больше чем просто танцор»: сальсеро не только должен танцевать «вообще» (то есть уметь двигаться как-то танцевально под музыку/быть танцором), но ещё должен уметь добавлять в свои движения специфику сальсы, «быть на стиле». Это означает, что сальсеро имеет мастерство больше, чем «просто танцор» — это более высокий системный уровень. Инженер-программист — это больше, чем «просто инженер»: он владеет общеинженерной спецификой (знает системную инженерию), но кроме того ещё и умеет программировать. Это общее рассуждение про стеки системных уровней в их ролевом рассмотрении, оно похоже для самых разных видов систем.

Вспомним про разделение труда: разные подроли обычно могут выполняться отдельными людьми, но в силу специфики танцевания — не могут, требуется собрать все роли в одном агенте. Но вот роли создателей в силу углубления разделения труда — они могут быть разведены по разным исполнителям. Наладка работы организма врачом, настройка работы тела тренером, обучение танцам

---

[275] https://vk.com/wall-179019873_401

[276] https://salsaselfie.com/all-cuban-salsa-moves/

учителем танцев, обучение конкретному танцевальному стилю учителем стиля — они могут (и, по большому счёту, должно) быть сделаны разными агентами, каждый из которых имеет достаточно времени, чтобы овладеть мастерством создания и развития соответствующего целевого мастерства в танцующем агенте.

Как определить, тут подроли — это отношение специализации или части-целого? В функциональном разбиении легко запутаться, но по большому счёту работает принцип разбирательства с этими «подролями»: **если даны какие-то взаимоисключающие альтернативы методов ролей в одном агенте, то это специализации (скажем, «строитель стен — бетонщик для скользящей опалубки или каменщик для кирпичной кладки»), а если нужны для исполнения полного метода и одно, и другое («строитель — каменщик для кирпичной кладки» или «строитель — маляр для окраски фасада») это уровни части-целого, системные уровни, то есть отношение композиции.**

И, конечно, тут важно ещё понимать, какой проект — в разных проектах могут приниматься разные решения о том, как моделировать методы работы/действий и исполняющие их роли, как именовать эти методы и роли. Что в одном проекте «варианты»/специализации, в другом проекте вполне может оказаться уровнями/частями (и нюансами, почему так, надо интересоваться у участников этого проекта). **Вниманием вы выделяете объекты для того, чтобы договориться о совместной деятельности в проекте, а не как-то «абсолютно объективно, независимо от ситуации, одно моделирование для всех проектов». Сами понятия «метод», «роль» — они часть мета-мета-модели мира (описывается в наших руководствах), они «для всех проектов». А вот мета-модель, типы которой берутся из мета-мета-модели — она существенно зависит от предметной области, прикладная.**

Скажем, в прикладной области социальных танцев есть «сальсеро», а есть «тангеро» (роль, которая танцует стиль аргентинского танго), это обычно — взаимоисключающие варианты в одном исполнении. У артиста::роль тем самым могут быть подроли-специализации (подтипы) сальсеро и тангеро, задействуемые в разных проектах, а ещё подроль-специализация пианист как надроль музыканта. Подумайте, почему пианист — надроль музыканта, а не подроль (тут та же причина, почему сальсеро — подроль артиста, но надроль танцора).

Но если в перформансе артисту потребуется вдруг для какой-то цели станцевать два танца, это будут подроли-части мультитанцора::роль. У мультитанцора части — сальсеро и тангеро, у них часть — танцор.

Поясним это примером с более простой физической системой. У дома-надсистемы есть части «внутренняя стена» и «фасад», в обеих этих частях есть подчасть «кирпич»: рассуждение с ролями артистов то же самое — и помним, что это рассуждение «в типах», ибо в стене и фасаде конструктивно кирпичи разные, но одного типа «кирпич», а в мозгу мастерство сальсеро и тангеро имеют одно и то же мастерство танцора, но оно работает дважды с разными параметрами. Мастерство — это «программа» (алгоритм, выполняемый вычислителем организма агента и приданным ему инструментарием) для выполнения какого-то метода (в нашем случае — танцевания сальсы и танцевания танго). Поэтому уместны аналогии с программированием: мастерство как программа может исполняться дважды, проходя и сохраняя в памяти при этом два набора состояний от двух разных исполнений как физических процессов. Мастерство, как и программа, может

быть представлено разными физическими объектами, если она реентерабельна[277], то есть разные экземпляры состояний переменных/памяти программы при разных её исполнениях не влияют друг на друга. Это всё онтологические рассуждения, системное мышление невозможно без освоения моделирования.

Моделирование — это прежде всего присвоение типа какой-то мета-модели объектам модели, оно делается на многих мета-уровнях абстракции. Поэтому системное мышление, включающее в себя прежде всего системное моделирование и последующие операции с моделью, невозможно «вообще», без привязки к проекту: в разных проектах на один и тот же вопрос системное мышление будет давать разные ответы, приводить к разному системному описанию — ибо один и тот же тип мета-мета-модели может быть присвоен разным типам предметной области в зависимости от того, с каким разделением труда мы работаем в разных проектах.

Это общая ошибка: требовать однозначного варианта системного моделирования (присвоения объектам типа мета-мета-модели системного подхода) какого-то набора систем, не поясняя проекта, для которого это моделирование делается. Так, при создании электромобиля Tesla создавался прежде всего флот/fleet (общее название для системы из средств транспорта, если это пассивный транспорт, то есть тележки или вагоны — это будет пул/pool) электромобилей и служб их продажи и технического обслуживания. Но потом пришлось делать отдельный проект — проект электрозаправок Tesla на всей территории США, и далее всего мира. Дальше можно обсуждать вечно: какую же систему создаёт Tesla? Проблема того же сорта, что и проблема биологического индивида: назвать эту систему невозможно. То же самое будет, если рассмотреть Tesla как заводы, создаваемые Tesla же: и там примерно те же вопросы — скажем, человекоподобный робот Tesla Optimus, который будет замещать рабочих — относить его к условным «сотрудникам» или относить его к «оборудованию»? А тут ещё этого робота хотят продавать наряду с автомобилями, что запутывает вопрос. Вряд ли вы узнаете, как договорились в Tesla проводить границы систем, границы системных уровней, это не публичная информация — но там точно об этом договорились.

Моделирование ведётся не просто так: оно каждый раз должно облегчать коллективное мышление в проекте создания системы! Выборы типов объектов и типов отношений для моделирования объектов и отношений из реальных ситуаций реальных проектов трудны, и во многом субъективны. Нельзя заведомо указать, «в такой-то общей ситуации выбирай такие-то типы», так не получится. В онтологии выбор вида отношения из их множества возможных для выстраивания иерархии (классификация, специализация, композиция как основные) — это по большей части субъективные предпочтения в моделировании, отражающие самые разные отношения объектов окружающего мира. В разных ситуациях возможны разные выборы, нет однозначно верного! В системном мышлении нет «рецепта», но есть типы объектов, которые можно попробовать использовать с наибольшей вероятностью получения понятных и логичных для самых разных участников проекта рассуждений. Но перескочить это «с наибольшей вероятностью» и попасть в ситуацию «гарантированно получить» — это не удастся. Плюс делайте поправку на использование естественного языка: если что-то может быть какими-то участниками проекта понято неправильно (например, термин протрактован «по-

[277] https://ru.wikipedia.org/wiki/Реентерабельность

бытовому»), то это обязательно будет понято неправильно. Так что документируйте модели и добавляйте заземление, то есть примеры из реальной жизни. Если вы сделали совсем уж неправильный выбор, вы обнаружите ошибку раньше.

Агенты, выполняющие роли сальсерос в паре, **собранны**/«предельно внимательны, выполняют психопрактику удержания многопоточного внимания». Их учитель **собранности**::«метод из интеллект-стека фундаментальных методов мышления» не пришёл в этот вечер в зал, но это не значит, что его не было и изготовленное им мастерство собранности не работает в составе всех этих упомянутых ролей (собранность базируется на трансдисциплине, она фундаментальна, «везде», по другую сторону от прикладных дисциплин).

Взгляд системного мыслителя везде в случае агентов видит:

- **Методы работы**, исполняемые **ролями**. Если какой-то агент существует и что-то делает, то это и значит, что он выполняет работу по какому-то методу — вопрос только, какой это метод. Чаще всего это метод создания систем какой-то природы, если мы говорим об универсальном агенте как создателе/constructor/«enabling system». Создавать и развивать систему — это изменять состояния каких-то объектов физического мира, чтобы они в итоге стали какой-то системой (а для этого ещё и работать с системными описаниями). Это изменение состояния делается по какому-то шаблону поведения, который мы и называем методом/способом/практикой/культурой/стилем/ работы. Ту функциональную часть агента, которая выполняет работы по этому методу, мы и называем ролью. **Это рассмотрение времени работы роли.**
- **Создателя мастерства** для исполнения работ не абы как, а по выбранному методу. Ключевой вопрос выполняющему работы по какому-то методу агенту: «где ты этому научился, кто тебя учил?» (при этом сам агент может не осознавать, что он не «просто работает», а «работает по методу, метод для этой работы, то есть для получения ожидаемых результатов, может быть и другим». Тот, кто «научил методу» может быть тот же агент, что и выполняет работу, он «самонаучился». Вполне возможно самообучение агента (назовём его роль в обучении::метод — «ученик») по книгам, статьям, курсам, руководствам, стандартам, регламентам (учит наполовину автор описания метода работы, наполовину сам себя агент, выращивающий у себя мастерство работ по методу, а не абы как/ad hoc), это может быть в существенной мере задействование какого-то отдельного учителя (живого или AI, или целой группы учителей, причём там будет просматриваться обширный граф создателей мастерства — этих учителей тоже кто-то должен создать) вдобавок к усилиям самого студента (он просто будет меньше «сам себе преподаватель»). В случае агентов с достаточным интеллектом обычно говорят не «создание и развитие», а «обучение»/«развитие» (ибо создание агента проходит чаще всего в рамках совсем других проектов: личность обычно не «создают», а только «развивают»). Особенности создания мастерства будут в руководстве по инженерии личности. **Это рассмотрение времени изготовления/развития мастерства для выполнения роли.**

Ключевое тут:

- **Все роли разных системных уровней агента работают одновременно. Мышечная работа в частях организма, как она**

видится физиологу, работа мышечных лент движенца/«телесного практика», и так далее — включая само танцевание::«культура/практика/метод движения» танцора::роль, исполнение сальсы сальсеро, танцевальный выход/перформанс артиста на мероприятии — всё это происходит одновременно. Все эти разные деятельности/практики/способы/методы работы/действий ума и тела танцующей пары и каждого отдельного партнёра мы выделяем в физическом мире исключительно нашим собственным вниманием: что-то выхватывается из фона и становится фигурой, а всё остальное — фоном.

- **Разные трудовые роли создателей (время создания мастерства танцевания сальсы — все эти физиологи, тренеры, гуру стиля), разные проектные роли сальсеро (которые одновременно присутствуют в ходе танцевания — телесный практик, движенец, танцор стиля) обращают своё внимание на разное и сами они могут работать в разное время, причём это время создания соответствующего мастерства какой-то частной роли. Создатели (в случае мастерства — это учителя, в общем случае — инженеры) выделяют в работающей системе разные объекты из фона своим вниманием. Все создатели могут рассматривать свои создаваемые и развиваемые системы, не останавливая работы этих систем «разборкой на части», причём они могут действовать одновременно. В случае танцевания — все врачи и тренеры (инженеры тела) и учителя (инженеры личности) видят каждый своё (результаты работы соответствующих систем организма и результаты работы соответствующего мастерства) в танцующей паре.**

Системные уровни. Продолжение примера социальных танцев: уровни выше танцора.

На сальсеро::роль история ещё не заканчивается, системный мыслитель всегда смотрит в объекты всё большего и большего масштаба, более и более высоких системных уровней! Вверх от уровня сальсеро::роль идёт артист::роль — мы уже неоднократно это упоминали в предыдущих абзацах.

В клубе рядом с учителем танцевального стиля («изготовителем» сальсерос) стоит приезжий кубинец, **хореограф::учитель::создатель**, который только что провёл мастер-класс по тому, как строить **композицию** в фигурах/moves сальсы, и он наблюдает за следующим системным уровнем мастерства выполнения особого метода: метода действий артиста::роль, который исполняет какие-то действия (про танцы не принято говорить «работа») в ходе «танцевального выхода»/перформанса. В социальных танцах принято композицию/хореографию танцевания создавать «на лету» тем же агентам, которые танцуют эту композицию в ролях **артистов.** Подроли артистов, танцующих хореографию (композицию из фигур, соответствующую каким-то художественным/artistic критериям) на предыдущем системном уровне назывались «сальсерос» и их метод/стиль был «танцевание сальсы» — но там не шло речи о композиции (вернее «и этой композиции тоже», то есть речь шла о «сальсе вообще, всех вариантах композиции», а не о конкретном выходе с его уникальной

хореографией/композицией. Ещё уровнем ниже были просто «танцоры» и их поведение было «танцевание», и там ещё не было сальсы (вернее, «и сальса тоже, если надо»). Ещё уровнем ниже они же были «движенцы» и их поведение было «движение тела», никаких танцев (вернее, «и танцы тоже, если надо»). И так далее — у каждой роли создателя какого-то ролевого уровня его целевое мастерство::система и порождаемый этим мастерством метод/стиль::«функция мастерства» (которую переносят на роль агента метонимически — практику считают выполняемой не мастерством как частью личности, а ролью всего агента, «смотрит не глаз тигра, а смотрит весь тигр» — но это всё онтологические тонкости).

Так какой же системой занимается каждый создатель (врач, тренер, самые разные учителя) в агенте, танцующем со своей парой в танцклубе? Каждый вид/специализация создателя — своей системой! **Система в глазах смотрящего, вернее, в глазах создающего (ибо дело не в пассивном наблюдении/смотрении, а в активном создании).** Все виды создателей смотрят на одного и того же агента, танцующего в клубе — но видят абсолютно разное, говорят на разных языках разных предметных областей, создают в агенте разные части (в том числе физически разные — скажем, затрагивающие физически разные нейроны, кодирующие то или иное мастерство).

Это и есть проявление эмерджентности: на каждом системном уровне меняются объекты рассмотрения, появляются новые свойства, которых на более низких уровнях ещё не было, а на более высоких уровнях уже не будет. Поэтому на каждом системном уровне меняется предметная область, к её объектам/предметам применяются свои специфические/предметные методы, выполняемые своими предметно-специфическими ролями, и у этих ролей свои предметно-специфические интересы.

Хореограф::(роль создателя метода для конкретного варианта танцевального выхода) внимательно смотрит на пару — и видит (выделяет вниманием!) не мышцы, не отдельные движения рук или ног, но последовательность фигур/па/steps/moves, как-то привязанных к музыкальным акцентам, и этот нарратив (последовательность) собирается в целый перформанс/выход, длящийся пять минут (удивительно, но отдельного слова в социальных танцах для этого перформанса/выхода/исполнения нет, его тоже называют «танец»). Хореограф оценивает: получилась ли приличная хореография (иногда сокращённо называют «хорео»), в которой сбалансированы самые разные фигуры сальсы, есть ли в этой хореографии какие-то модные фишки (например, вставки из других стилей танца), есть ли какая-то интересная композиция. Это уровень **танцевального перформанса/выхода** в целом[278] — какая будет рассказана артистом::роль история/нарратив, как сочетается ритм движения исполняющих хореографию (последовательность определённых движений, предусмотренных хореографом) артистов с ритмом музыки в каких-то больших структурах (не на уровне одного бита, не на уровне одного движения, а на уровне музыкальных фраз и более длинных структур), будет ли набор фигур неожиданным, разнообразным и интересным, или скучным и однообразным.

Многозначное слово «танец» (который и особый метод движения, и стиль в рамках общего метода движения, и сам перформанс/выход/исполнение какой-то

[278] Уровень композиции подробней описан тут — https://vk.com/wall-179019873_432

хореографии) мы тут не использовали, а ещё для учебных целей приводили множество синонимов для терминов и явно оговаривали типы. Поэтому язык разговора о ситуации в клубе (танцевальном выходе/перформансе сальсы) получился немного корявым, зато более точным. «Танцоры из клуба» этот текст вряд ли одолеют, для его понимания надо пройти для начала «Рациональная работа», а затем «Системное мышление» вот прямо до этого места, которое вы сейчас читаете.

Неудивительно, что агенты учатся мастерству выполнения роли артиста-«танцора стиля»-танцора-движенца (выбранный какой-то путь от вершины к основанию иерархии часть-целое мы будем называть «системным стеком» — тут стек мастерства, определяющий стек возможного выполнения стека ролей в стеке методов) много лет до хоть какой-то сносной степени мастерства на каждом уровне. Эти агенты получают не инженерные точные объяснения, а только в большом количестве примеры вот этих самых «танцевальных выходов», и просто пытаются повторить уж что поняли, а также надеются, что каким-то образом разные упражнения (например, силовые тренировки, растяжки и т.д.) помогут им в перформансе.

Особо подчеркнём: мастерство какой-то степени нужно на каждом уровне! Ибо если ты очень хорошо строишь композицию, очень плохо исполняя стиль какого-то танца (например, сальсы), при этом будучи даже неплохим танцором и отличным движенцем, тебя за «своего» в сообществе танцоров какого-то стиля не признают. Аналогично, представьте себе борца греко-римской борьбы, который владеет своим телом в совершенстве, ибо «много занимался телом». Вот он выходит на танцпол — и увы, на его танцевание смотреть будет нельзя, а вот на борьбу — можно. Надо понимать, что там собственно танцевального, стилевого, хореографического/композиционного в движениях тела. Это же верно для многих и многих методов: человек с огромным интеллектом (нижний уровень мастерства для многих и многих прикладных методов) в любом прикладном методе, которому он не учился, будет выглядеть примерно так же нелепо, как отличный «движенец» в танцах, если он не учился именно танцам, а мастерство движения применял где-то в другой области (скажем, в борьбе или подъёме штанги). Это рассуждение верно и для инженера, который своё мастерство инженерии развивал, скажем, в области электроники, или программной инженерии, а затем вдруг взялся учить людей (методы инженерии личности) или учить целые организации (методы менеджмента). Ожидание успеха тут примерно такое же, как у борца или штангиста в танцах. Другое дело, что нужный уровень мастерства танцевания при обучении танцам агент с мастерством и телом штангиста или борца получит быстрее, чем «человек с улицы», не имеющий мастерства движения на уровне этого агента-борца или агента-штангиста. И нужный уровень мастерства обучения личности или мастерства обучения организации (менеджмент) агент с мастерством инженера получит быстрее, чем кто-то, кто не имеет мастерства системной инженерии (общая часть инженерного знания для всех видов создаваемых и развиваемых систем).

Мы не можем отменить то, как разговаривают люди, создать какой-то специальный язык (хотя инженеры, математики, физики ровно это и делают, создают терминологию своих предметных областей совершенно сознательно). Слово «танец» всё равно вам встретится, и тогда вам нужно самостоятельно определить — что это за объект, какого он типа, что имелось в виду. И это будет почти со всеми терминами, которые мы использовали в рассказе. Так, можно **сальсой** назвать и

стиль танца (те постуры — позы с предписанной мышечной работой в этой позе, ритмику шага и какие-то характерные/signature фигуры/steps/moves, которые в перформансе делают сальсеро), и конкретный перформанс в стиле сальсы (который длится пять минут)! Впрочем, и когда мы говорили о тренере, то описывали ситуацию, где как-то можно было легко перепутать тренера-как-роль, с тренером-как-человеком::агентом в роли тренера, который одновременно играет и роли инструктора-телесника и фитнес-инструктора. Композиция может оказаться результатом применения методов композиции (сама последовательность фигур в перформансе, или иногда даже описание последовательности фигур, а не сама последовательность фигур как динамический 4D объект), а может быть и сама методом работы, то есть поведением — то, что делает хореограф, «композитор перформанса». Нужно быть внимательным к типам каждого термина, при непонимании — переспрашивать!

Требуется разобраться в онтологии и семантике, чтобы рассуждать системно! Так, мебелью можно назвать и всю мебель квартиры, и стол со стульями, и только стул — всё мебель, но каждый раз приходится уточнять, что имеется в виду. А уж сколько разночтений можно найти в терминах «предприниматель» (мы не рекомендуем его употреблять, в голове у каждого агента — человека и AI — это абсолютно разные типы) и стейкхолдер (путаются роль и агент-в-роли, а также любая роль и внешняя роль проекта).

С термином «танец» то же самое: нужно всегда уточнять: любой танец, конкретный стиль, или конкретный перформанс/выход имеется в виду под словом «танец», а чтобы не запутаться и удерживать внимание на выбранном системном уровне и его понятиях — нужна собранность, и она рекомендует не просто рассуждать, размахивая руками, а записывать. Мы описывали ситуацию в клубе на десятке страниц, но выход в социальных танцах — это всего двое на пять минут. Если описывать предприятие, то легко обнаружить систему из двух тысяч агентов с их оборудованием, сотнями ролей и рассматриваемые сроки его проектов — год или даже десять лет. **Рассуждения про роли и их методы, а также развитие предприятия как создание его материальной базы (инструментов) и мастерства (это будет мастерство сотрудников и компьютерных программ предприятия) будут примерно такими же, какие мы привели для ситуации танца в клубе, только этих ролей и методов — тысячи. Без привычки точно присваивать типы объектов и документировать/записывать описания уточнённых объектов в их самых разных отношениях и с указанием типов — пропадёте.**

Но и хореограф с его методами хореографии/«танцевальной композиции»--это тоже не конец истории, вверх по уровням можно идти довольно долго, и каждый раз находить что-то интересное. **Запросы «для чего это надо» идут всегда сверху вниз по системным уровням, то есть от функций больших по размеру систем к функциям малых систем (скажем, запрос к владению телом идёт от хореографии, если там будет какой-нибудь хитрый поворот, выводящий тело из балансирования), а решение этих запросов (и наоборот, понимание, что в текущей «элементной базе» более нижних уровней нельзя решить запрос) — снизу вверх.** То есть для того, чтобы быть готовым к любой хореографии, надо быть неплохим движенцем/физкультурником, и дальше вниз — иметь неплохое здоровье, помним, что «здоровый человек» тоже может быть выделен вниманием в этой ситуации, внимание может (осознанно!

Контролируемо!) переключиться с «артиста» через много уровней вниз к «здоровому человеку» (больной человек сальсу не станцует! И греко-римской борьбой не сможет заниматься, не сможет выйти на «поединок» как аналог «танцевального выхода»).

Но для ответа на вопрос «что вообще тут в этом танцевании происходит», надо продолжать подниматься по системным уровням вверх.

Рядом (в этот же момент времени!) на краю танцпола стоит **орг (организатор) сальсатеки** (вечеринка сальсы — сальсатека, вечеринка танго — милонга, вечеринка хастла — дискотека) и оценивает сальсатеку как вечеринку/мероприятие в целом. Для него танцующая пара — **участники проекта «создание и развитие сальсатеки::вечеринка::мероприятие»**. Ибо танцевальная вечеринка состоит главным образом из вот этих танцевальных перформансов/выходов, их одновременно идёт на танцполе множество (танцуют множество пар), и ещё они следуют один за другим — с каждым новым музыкальным треком идёт новый танцевальный выход, скажем, полусотни танцевальных пар. Орг включает в своё рассмотрение и эти перформансы, и стоящих/нетанцующих партнёров и партнёрш с их дисбалансом (дисбаланс — это специфический термин вечеринки, когда партнёров и партнёрш не равное количество: не все смогут выйти на танцпол, даже если захотят!), и диджеев, и фотографов с видеографами, и даже гардеробщиков (ибо вечеринка начинается, как и театр, с вешалки).

Что обсуждается на этом **уровне мероприятия/вечеринки**? Почитайте социальные сети, в которых обсуждаются танцы. Вы обнаружите, что обсуждается больше всего именно этот уровень, обсуждение ведут не столько танцоры::роль или «танцоры стиля»::роль, сколько «участники вечеринок»::роль (но все эти роли — это одни и те же люди-агенты, причём исполнение этих ролей одновременно):

- сколько вы танцуете или не танцуете (потому как не с кем — не нашлось пары из-за гендерного дисбаланса, или партнёрша отказала в танце, или было слишком жарко, или ещё что-то произошло[279]),
- какое у вас после вечеринки настроение, на какую вечеринку нужно идти, чтобы нормально потанцевать (то есть получить хорошее настроение — какие орги на своих вечеринках гарантируют хорошее настроение, а кто не может ничего гарантировать?),
- какая должна на этих вечеринках играть музыка, чтобы под неё было интересно танцевать, и у всех участников этих вечеринок было хорошее настроение,
- сколько это всё должно стоить, кого можно было бы пускать бесплатно.
- … и так далее: страсти кипят, всё крутится вокруг плохого или хорошего настроения на вечеринках, адекватности потраченных на мероприятие денег и удовольствия от мероприятия, альтернативного поведения (куда можно ещё сходить за подобные деньги для бóльшего удовольствия).

Обратите внимание, собственно движение танцевания::метод, а также создание хореографии/композиция::метод (то, что происходит в ходе танцевальных выходов/перформансов) в социальных сетях почти совсем не обсуждаются! Вечеринку «танцем» в быту (это ведь только мы не используем этот многозначный термин) обычно не называют, танцевальные перформансы/выходы (система, в которой участвует танцевальная пара в ролях артистов, исполняющих хореографию

[279] Пример тем, обсуждаемых на этом уровне — https://vk.com/wall-179019873_1740

в двух танцевальных подролях — лидер и фолловер) — это тут лишь часть того, что происходит на вечеринке. Вечеринка::мероприятие — это более высокий системный уровень, там кроме лидеров и фолловеров участвуют множество других ролей.

Тут тоже легко запутаться в терминах. Вечеринку часто называют «танцы» — тот же «танец», просто во множественном числе! И паттерн/метод/стиль действий походов на вечеринки назовут «занятия танцами» («занятие» в смысле траты времени) и учебные занятия по достижению танцевального мастерства тоже называют «занятия танцами». Всё путается, включая «время изготовления мастерства участия в вечеринке» и «время использования мастерства участия в вечеринке». Мастерство «участия в вечеринке» (по выполняемой практике) или мастерство «участника вечеринок» (по отыгрываемой роли) включает в себя и мастерство выполнения многочисленных перечисленных ранее функций/методов, начиная с функционирования организма (предмет интереса физиолога), заканчивая культурой/методами участия в вечеринке, выполняемыми ролью «участник вечеринки» и заключающейся в уместном там поведении. Это время использования мастерства участника вечеринки в его подроли артиста. А время изготовления? Помним о всех этих тренерах, инструкторах, гуру стиля, хореографах — и человеке::агент, который ещё только будет участвовать в вечеринке, но пока не имеет мастерства в задействовании осознанного или неосознанно выбранного варианта метода участия.

Следующий системный уровень уже не имеет тех, кто им специально занимается, но иногда какие-то организаторы вечеринок, преподаватели или даже отдельные танцоры вдруг выступают как **«амбассадоры (посланцы) субкультуры»**::роль и начинают обсуждать, как они развивают сальса-субкультуру с её многочисленными **неформалами**::роль. Участники вечеринок сальсы и ведущие связанный с этим особый образ жизни люди тут — неформалы::роль, имя нашей танцующей пары опять поменялось! Это системный **уровень субкультуры** конкретного стиля танца, в нашем случае сальсы, но обычно в таких субкультурах танцуют какой-то «зонтик» танцевальных стилей, собираемых вокруг одного танца. Так, в субкультуре сальсы часто танцуют и кубинский сон, а также неожиданно доминиканскую бачату и бачату сеншуал, в субкультуре кизомбы танцуют не только кизомбу, но и сембу, урбан киз, таррашинью, таррашо, неожиданно гавайскую компу, а также афро-хаус (в виде «анимации/развлечения» участников вечеринки — guided line dance[280]).

Сообщество/тусовка как групповой/коллективный агент, выполняющий роль субкультуры и практикующий поддержание/репликацию и развитие мемома социального танца для какого-то танцевального стиля, включает в себя в качестве своих частей все вечеринки «зонтика» танцевального стиля (в данном случае — «зонтика» сальсы). Например, в Москве одновременно идёт пять-шесть вечеринок. Это сообщество включает и все танцевальные школы, где работают тренеры, гуру стиля, хореографы (в Москве для сальсы их более десятка). Школы учат (репликация мемома субкультуры), вечеринки развлекают (использование мастерства самых разных субкультур/«методов действий» её адептами::«носитель мемома субкультуры»), а ещё проводятся конкурсы и фестивали. Субкультура/сообщество тем самым вполне материальна. Скажем, сто лет назад субкультуры сальсы как всех этих постоянно проходящих вечеринок-

[280] https://en.wikipedia.org/wiki/Line_dance

школ-конкурсов с их многочисленными участниками в самых разных ролях (не все там танцуют!) ещё не было, а через сто лет, наверное, уже не будет. В Арктике и Антарктике субкультуры сальсы нет и сейчас. Субкультуру можно считать не только методом действий, но функциональным объектом, который занимает определённое место в пространстве-времени, то есть совпадает с конструктивом: сообществом, «людьми, которые практикуют какие-то определённые методы, носители мемома субкультуры», ибо на этом уровне мы больше говорим о создании и развитии какого-то мастерства, какого-то практикования этого мастерства (действий, определяемых мастерством), эволюции практики, поэтому меняется язык разговора. О субкультуре можно думать как о материальном объекте, она изменяется во времени, хотя масштаб времени опять поменялся. Это не доли секунды на телесную работу опорно-двигательного аппарата или смены ощущений сомы, не секунда-другая на отдельное движение, не три минуты выхода/перформанса, не три часа вечеринки. Изменения субкультуры (изменение реплицирующегося мемома с идеями какого-то стиля социальных танцев, этот мемом и определяет субкультуру — как геном определяет вид живого существа, а не феном, не организм, так и тут — не член субкультуры определяет субкультуру, а мемом, отреплицированный в его голове) занимают даже не месяцы, а годы. Субкультура эволюционирует вместе со всеми своими методами/стилями поведения, которые выполняются ролями коллективных и индивидуальных агентов и даже частей агентов.

Системное мышление нам помогает снизить сложность мышления в том числе и о субкультуре, запредельно сложном многоуровневом объекте. Мы должны просто думать там отдельно о каждом системном уровне, управлять своим вниманием при переходе от уровня к уровню, отслеживать внимание разных других ролей к разным явлениям на разных уровнях (прежде всего к тому, что там делают: внимание к методам/практикам всех ролей).

Все эти роли и их методы действий (часто говорят «методы поведения», но метод — это тоже поведение, метод исполняется, это паттерн действий, так что тут некоторый онтологический дребезг из-за неразличения метода и работы как разных видов поведения) рассматриваются на каждом уровне в их тесной взаимосвязи, но всё-таки отдельно:

- субкультура сальсы (реализуемая коллективным агентом неформалов::«участники субкультуры», при этом субкультура тут термин предметной области, это не «подметод», а агент. Метод поведения самой субкультуры — «бытование», специфических терминов тут нет, слишком высокий системный уровень),
- частью этой субкультуры является вечеринка в нашем клубе в конкретный вечер (реализуемая коллективным агентом «участники вечеринки»),
- частью вечеринки является пара артистов в составе мини-мероприятия «танцевальный перформанс/выход» (каждый артист реализуется отдельным агентом/личностью и на этом уровне есть ещё танцевальные роли лидера и фолловера в паре),
- частью артиста является сальсеро (собственно сальсу как танцевальный стиль танцует именно сальсеро),
- частью/подролью/составляющей сальсеро является танцор (вот он «просто танцует, причём красиво», и внешне в мире это тот же агент/личность, что и реализующий артиста),

- частью танцора является движенец (двигается как раз он, и это внешне тот же агент, выделить движенца можно только вниманием!),
- … и так далее, дело может не остановиться даже на здоровом человеке в целом (и вместо тренера или учителя тут будет врач, занимающий организмом в целом), ибо проблемы могут быть и на уровне каких-то отдельных частей организма (например, неврологические проблемы, или проблемы мышечной ткани, обусловленные генными болезнями). Ниже физиологического уровня будет уровень биохимии в работе мышц (и там будут чинить эти «поломки» биохимики), а сами мышцы будут разбираться в контексте целого организма, для которого нужно состояние «здоровый человек» (занимается этим врач-физиолог).

Если на каком-то системном уровне обнаруживается проблема — можно звать какого-то «создателя мастерства в каком-то методе» (консультанта, тренера, инструктора, наставника): где-то один такой создатель научит танцшколу методам зарабатывания денег (слово «метод» в речи опускается как название типа, «научит зарабатывать деньги»), а где-то другой создатель научит движенца методам движения руками-ногами (слово «метод» в речи тоже будет опущено, как тип мета-мета-модели, будет «научит двигать руками-ногами»). Эти «создатели и развиватели» мастерства работают во время создания, а не во время вечеринки и рассматриваемого нами в этом подразделе танцевального выхода/перформанса пары артистов сальсы, причём думать об исполняющих все эти многочисленные роли субкультуры коллективных агентах и индивидуальных агентах-личностях можно абсолютно одинаково, и это существенно экономит мышление.

Есть ли системные уровни выше уровня субкультуры? Конечно. Например, **уровень (всей танцевальной, но мы можем эту специализацию и пропустить) культуры в целом** (включая бальные танцы, сольные танцы, но может быть и театр, и моду, и живопись с графикой, поэзию и т.д. — то, что в быту называют «культура», имея в виду «что-то гуманитарное, направленное на развлечение». Да, это респектабельное занятие для профессионалов, но всё-таки речь идёт о развлечении — хоть речь идёт о Достоевском, хоть о Майкле Джексоне, вся «культура» — это сфера работы профессиональных развлекателей. Конечно, профи в области культуры будут возражать, что они не развлекают людей, а делают что-то более возвышенное, но давайте присмотримся внимательней — речь идёт о порождении и репликации каких-то мемов, и всё.

Этот уровень культуры/«общества культурных людей» (развлекающих и развлекающихся) включает в себя все субкультуры/сообщества, тоже вполне материальный объект. В расширительном толковании инженерию как метод изменения мира к лучшему тоже можно назвать одним из методов человеческой культуры, но это уже расширительное толкование «культуры», близкое к тому, что употребляем тут мы, говоря о практике/культуре/методе работы как типе мета-мета-модели — только «культура» тут употребляется в значении «мемом, определяющий мастерство человечества», то есть набор теорий/знаний/объяснений/алгоритмов, определяющий умение человечества (а сейчас «агентечества» с учётом AI) делать то, что сейчас умеют делать все разумные агенты Земли в целом.

Будьте внимательными: в речи слово-термин «культура» может быть использовано по-разному:

- Отсылка к методу (культура тут — синоним метода/практики/стиля работы/действий), культура::поведение.
- Отсылка к коллективному агенту (множество людей — общество или сообщество, например, культура неандертальцев — это все неандертальцы).
- Отсылка к знаниям/объяснениям/алгоритмам/дисциплинам методов, накопленных коллективными агентами («культура человечества» — это все знания/объяснения/теории, известные человечеству), иногда даже не все знания, а только лучшие из них (скажем, квантовая физика будет отнесена к культуре человечества, а вот теория флогистона — нет). Тут ещё близко понятие «материальной культуры», ибо кроме знаний методов нужно ещё и рассматривать инструментарий этих методов. Вот этот инструментарий в совокупности — материальная культура.

Следите по контексту, какой онтологический тип скрывается за каждым словом!

Хотя трудно представить роли людей, которые профессионально занимаются танцевальной культурой в целом и уж тем более общечеловеческой культурой в целом для целей их изменения (культурологи тут не в счёт, они «аналитики»: всё понимают, ничего не делают — да и если бы хотели, ничего сделать не смогли бы: эволюционирующие коллективные сообщества уровня выше организации не так-то легко изменить. Мало кто может похвастаться «вкладом в культуру» на очень высоких системных уровнях, например, уровне общества или человечества).

Чем выше по системным уровням, тем труднее найти людей, которые пытаются что-то делать на этих уровнях (чиновники могут утверждать, что «развивают культуру», но это утверждение обычно нельзя проверить, а часто «настоящая культура» наоборот, развивается мимо чиновников, мимо цензуры — и субкультура не менее часто называется поэтому **контркультурой**[281], она и оказывается главной движущей силой культурного развития).

А выше культуры? Есть, конечно: цивилизация в целом (включая не только культуру из «домов культуры», но производственную культуру, политическую культуру, культуру исследований, культуру воспитания детей). Но мало кто::агент будет утверждать, что он «создаёт и развивает» цивилизацию в целом. На уровне цивилизации в целом с её необозримым мемомом отдельным индивидам и даже коллективам (предприятиям) или отдельным сообществам (даже политическим партиям) можно работать только маленькими отдельными кусочками, небольшими наборами мемов/знаний отдельных методов/практик, которые практикуют ограниченное число агентов.

В проектах неизбежно рассмотрение иерархий целевой системы по отношению композиции, причём границы отнесения каких-то объектов то ли к подсистемам целевой, то ли к окружению — они обсуждаемы. И неизбежно рассмотрение самых разных создателей. А если речь идёт о людях и AI как создателях, то неизбежно создание и развитие мастерства путём обучения, плюс ещё и организация этих создателей в рамках какого-то оргзвена (например, предприятия в целом). Поэтому занимаемся:

- Системной инженерией как общим методом создания каких-то принципиально многоуровневых целевых систем
- Инженерией личности — созданием из универсальных создателей с

[281] https://ru.wikipedia.org/wiki/Контркультура

нейросетевым вычислителем (людей, но сейчас и AI) прикладных создателей с мастерством выполнения какого-то прикладного метода/способа/культуры/практики/стиля работы, это принципиально многоуровневая работа, личность многоуровнева, и пример социальных танцев должен был как раз это проиллюстрировать (обратите внимание, сколько перечислено учителей у артистов-танцоров!).
- Системным менеджментом — организацией из создателей более низкого системного уровня (например, отдельных людей, отдельных AI, но это могут быть и отдельные подразделения) систем-создателей более высокого уровня (например, подразделений, но это могут быть и отдельные предприятия).

Разделение труда между создателями. Наша система. Граф создателей.

Особенность системного рассмотрения в том, что объекты всех системных уровней существуют и взаимодействуют в момент работы **нашей (кто — «мы» в этом «нашей»? Это оказывается важно! Разные «мы» — разные ответы!)** целевой системы. У **нас** в примере социальных танцев из предыдущих разделов целевая система неявно была задана как выполнение конкретной работы по методу создания перформанса сальсы, который выполняет пара артистов::роли, задействуя свою подроль сальсеро и танцевальные подроли лидера и фолловера. Артисты реализуются парой людей-агентов (один агент в роли лидера/ведущего, другой в роли фолловера/ведомого/следующего) в одном из клубов в Москве.

Но это не факт, что автор руководства прописывал именно «пару артистов» как целевую систему в каком-то проекте! И вряд ли автор писал этот текст как амбассадор субкультуры, называя целевой системой сообщество сальсы (чаще говорят «сообщество танцоров сальсы», хотя мы помним, что там кроме сальсерос довольно много и других ролей — орги мероприятий, диджеи, фотографы) как часть сообщества социальных танцев!

С другой стороны, текст оказывается очень удобен для того, чтобы выбрать какую-то целевую систему: все возможные роли как кандидаты в целевые системы в нём как на ладони. При этом речь идёт о времени танцевания (а не подготовки/создания танцевального выхода). В это время системные уровни существуют и работают одномоментно для ролей агентов-участников танцевания, но разные роли вокруг целевой системы выделяют своим вниманием разные объекты, каждая на своём системном уровне (некоторые роли ещё и роли учителей::создатели, в танцевании они не принимают участия!). Происходящее на других системных уровнях в ходе танцевального выхода (использования мастерства, которому учат) этими ролями создателей учитывается, но не в центре их внимания, не предмет интереса.

Знание о существовании множества системных уровней в самых разных системах позволяет разделить сложный проект по созданию и развитию каких-то систем на множество отдельных проектов, выполняемых какими-то отдельными ролями, а роли поручить разным агентам.

Производитель кирпичей (завод! Но мы назвали его просто по роли) берёт глину, изготавливает кирпичи. Каменщик (человек! Но мы назвали его просто по роли) изготавливает из кирпичей стену. Прораб (человек в роли менеджера, тут помянута должность) из стен изготавливает здание, координируя («назначения работ»::метод менеджером::роль, при этом помним, что в бытовой речи типы мета-мета-модели не произносим) подчиняющихся ему в силу должности каменщиков.

Застройщик (компания) из зданий, строительство/изготовление которых координируется прорабами, создаёт целый квартал. Строительный девелопер (компания) из кварталов создаёт какую-то зону застройки, городской архитектор (подразделение мэрии как учреждения) из этих зон застройки создаёт город (или с нуля, или модернизацией уже имеющегося).

Это всё было описанием проектов, происходящих с системами разных системных уровней. Разные агенты (люди, AI и их организованные коллективы), помянутые в предыдущем абзаце, работают с городом в разных ролях на разных системных уровнях: кто с городом как целым, кто со зданиями города, кто с глиной кирпичей города. Хотя фраза «глина кирпичей города» уже «не звучит», по системным уровням кирпичи и тем более глина кирпичей уже далеко от города. Обычно **осмыслены фразы, где связываются системы смежных системных уровней, а не далёкие друг от друга по иерархии отношений композиции системы**. «Глина кирпичей» (тут отношение композиции/«часть-целое»!) или «здания квартала» (тут тоже отношение композиции/«часть-целое»!) тут много осмысленней, это как раз системы смежных системных уровней.

Мы уже упоминали эту особенность системного мышления, когда обсуждали хвост коровы Маргариты, которая сама входит в стадо как его часть. Говорить, что «хвост — это часть стада» верно для математики и физики, но неверно для системного мышления. Говорить, что танцы — это биохимические процессы в мышцах, это формально верно, но неверно для системного мышления. Говорить, что город состоит из кирпичей и бетонных панелей, а ещё дорог между группами кирпичей и группами бетонных панелей — формально верно, но так говорить в системном мышлении неправильно.

Если понимать, что для создания и развития систем разных системных уровней требуется разное мастерство (то есть думать системно), то первое же, что можно делать — это поручить изготавливать системы разных уровней не одному агенту, а целой группе агентов, каждый из которых успевает за долгое время своей жизни и работы приобрести глубокое мастерство в предметной области.

Роберт Хайнлайн писал: «Любой человек должен уметь менять пеленки, планировать вторжения, резать свиней, конструировать здания, управлять кораблями, писать сонеты, вести бухгалтерию, возводить стены, вправлять кости, облегчать смерть, исполнять приказы, отдавать приказы, сотрудничать, действовать самостоятельно, решать уравнения, анализировать новые проблемы, вносить навоз, программировать компьютеры, вкусно готовить, хорошо сражаться, достойно умирать. Специализация — удел насекомых»[282].

Этот афоризм можно понимать как гимн человеческому интеллекту: потенциально да, любой человек может освоить все эти методы, даже «вносить навоз» (в том числе в промышленных масштабах[283], а не в масштабах любительского огородничества[284]). Но увы, за время человеческой жизни нельзя научиться работать по всем этим методам профессионально, а также нельзя успевать отслеживать все изменения в непрерывно улучшающихся методах работы, чтобы удерживать своё мастерство современным. Решение тут — **разделение труда**, обычный путь цивилизации. Один «неразбиваемый на частные методы»

---

[282] Роберт Хайнлайн, «Достаточно времени для любви», 1973

[283] https://direct.farm/post/vneseniye-organicheskikh-udobreniy-5847

[284] https://navozov.ru/blog/navoz-kak-udobrenie-vidy-i-sposoby-primeneniya/

метод/дело/практика/деятельность/занятие, который выполнялся одним человеком в прошлом, сегодня начинает выполняться хорошо, если это выполнение ведётся дюжиной разных ролей (часто больше), в которых специализируются разные агенты (люди и AI, и даже целые организации). Эти агенты также включают в свой состав разные всё более специализированные для выполнения этих стремительно расходящихся ролей инструменты. В этих более и более узких методах/«видах труда/деятельности/инженерии» индивидуальные и коллективные агенты достигают высоких уровней мастерства, ибо они имеют достаточно времени на достижение этого мастерства, они фокусируются на выполнении одного узкого метода работы, у них есть время для:

- **Развития** — смены мастерства в одном поколении метода на мастерство нового поколения метода, скажем, переход от метода «программирование на языке Fortran» в 80х годах 20 века к методу «программирование на Matlab» в начале века, а затем к методу «программирование на Julia» в 20х годах 21 века. Развитие — это «научиться что-то делать новое», подразумевает два метода: старый (возможно, отсутствующий, если «никогда этим раньше не занимался») и новый.
- **Совершенствования** — достижение высокого уровня мастерства, беглости в выполнении работ по какому-то методу текущего поколения, «набитие руки». Например, после перехода с одного языка программирования на другой как «развития», то есть изменения метода работы (и знаний/дисциплины, и инструментария), надо всё-таки потратить время на совершенствование/«беглость и безошибочность» в использовании этого языка, «тренировку» уже имеющегося мастерства, уменьшение процента ошибок и увеличение скорости того, что уже умеешь делать. Совершенствование — это про один метод, который надо не просто освоить, но добиться в нём высокой степени мастерства.

В примере социальных танцев из предыдущих подразделов как раз описано современное разделение труда: нет одного «создателя мастерства артиста». У агентов, которые танцуют в клубе сальсу, множество самых разных создателей, выполняющих самые разные роли — от врача, через тренера/инструктора фитнеса и учителя танцев к гуру танцевального стиля и хореографу. Но на уровень выше (создание и развитие вечеринок::мероприятий) разнообразие ролей ещё выше: совсем верхнеуровневые роли оргов вечеринок, фотографов и видеографов, артистов для шоу и учителей для мастер-классов (часто проводятся прямо на вечеринках), аниматоров для линейных танцев. А ещё выше там роли амбассадоров культуры, и это тоже могут быть отдельные профессионализирующиеся в этой роли агенты.

О **глубине разделения труда** говорят как о раздаваемом по разным агентам для последующего коллективного выполнения исходной роли числе уровней подролей, на которые распадается исходная роль и числе подролей на каждом уровне. Разделение труда, оказывается, довольно глубоко даже в танцах! Там уже ушли от ситуации «один учитель танцев учит одного танцора». Нет, подготовкой одного артиста-танцора занимается много агентов-создателей, имеющих разную специализацию. Инженерия личности (инженерия набора мастерства в человеке) требует не меньшего числа участников команд с разной специализацией, чем инженерия какой-нибудь программной или аппаратной «железной» системы (например, робота).

Системные уровни хороши как раз тем, что позволяют обсудить: кто::создатель что::система делает, и что можно не делать, и что нельзя не делать. Всё это будет упрощено (моделирование оставляет только важное!) разложено по полочкам и документировано (мозгу не верим, он забывчив, верим записанному), и сверхсложное станет более простым.

Так, сверхсложный город можно обсуждать как состоящий из кварталов/зон застройки, кварталы как состоящие из зданий, здания как состоящие из стен, и никакого винегрета типа «кварталов из стен». С городом просто понимать идею системных уровней, с танцевальной вечеринкой — сложней, с предприятием — ещё сложней. Но разобраться с предприятием::создатель::система и выпускаемыми им системами::«целевая система» всё-таки можно, мышление о предприятии будет тем же самым: разные системные уровни выпускаемых предприятием систем и систем-создателей подсистем систем на этих уровнях (а внешние роли для предприятия ещё и занимаются созданием надсистем). Главное, что разобраться с тем, чем заняты самые разные агенты в предприятиях как коллективных агентах, можно: просто помним о разделении труда, поэтому выстраиваем размышление от целевой системы (обязательно в окружении!) в момент её работы к целевой системе из конструктов, затем методов создания целой системы из конструктов, затем роли для этих методов и дальше — агенты с нужной степенью мастерства, а далее агенты, организующие агентов-сотрудников. Мы наблюдаем агентов при бытовом внимании к предприятию, а системное мышление говорит, какая должна быть траектория внимания по объектам, связанным с предприятием, если пользоваться системным моделированием. Поскольку объектов оказывается много, то предлагается документировать результаты системного моделирования, не ограничиваться «мышлением в уме», «мышлением проговариванием», и даже «мышлением письмом», но проводить «системное мышление моделированием».

В этом **прелесть системного мышления: один раз понял, а затем используешь это мышление для всех самых разных систем, которые тебе встречаются. Окружающим агентам эти системы кажутся абсолютно разными, но системный мыслитель находит все эти системы довольно похожими друг на друга, мышление системного мыслителя про эти системы — беглое, он разбирается только с прикладными нюансами проектной ситуации, а в общих чертах любая проектная ситуация для него выглядит одинаковой, он уже знает про там происходящее в объёме знаний мета-мета-модели, «в общих чертах». Это экономит очень, очень много времени! У системного мыслителя всегда есть чеклист: на что обратить внимание в незнакомой ситуации. Поэтому по-настоящему незнакомых ситуаций не случается, растерянности перед хаосом окружающего мира нет.**

Совершенно неважно, какая система, если речь идёт о системном мышлении. Мышление будет устроено одинаково, внимание будет удерживаться системными уровнями, хотя содержание мышления будет абсолютно разным для разных видов систем, разных системных уровней. Типы мета-мета-модели одни и те же (мышление ведётся в типах! Мета-мета-модель — это типы из нашего руководства, типы понятий интеллект-стека фундаментальных методов мышления), а вот типы мета-модели (предметной области, метаУ-модель из общего учебника какой-то дисциплины, метаС-модель ситуационная, как сложилась в какой-то организации и отражено, например, в регламентах и корпоративных стандартах) будут

существенно отличаться.

Пример космической ракеты легче понимать, чем танцевальный пример, ибо ракета не живая, и не учится, хотя AI в современной ракете уже не факт, что не учится примерно так же, как люди-агенты в танцевальном примере. Но мышление о космическом корабле устроено так же, как мышление про танцоров: если корпус корабля изготовлен из неправильного для успеха системы материала (например, из алюминия), то нужно опускаться на уровень рассмотрения материала, и решать проблему (например, брать сталь[285], как это сделал SpaceX с ракетой Starship). Иначе из-за проблем с материалом нарушится работа всех остальных более высоких системных уровней ракеты, она не сможет летать, или будет летать не очень надёжно.

Проблемы возникают из-за неправильной совместной работы многих системных уровней, помним о конфликтах систем разных системных уровней. Обычно задачи, которые нужно решать в ходе создания и развития систем, приходят с более высоких уровней, в конечном итоге от надсистемы, которая требует от целевой системы выполнения какой-то функции — и это рассмотрение идёт на много уровней вниз.

Ракета получает свою функцию летать с какими-то характеристиками с более высокого системного уровня. Например, можно рассмотреть космическую компанию вроде SpaceX, ставящую задачи для связки ракеты и космического корабля, или заказчика полётов корабля у такой компании, например, телекоммуникационную компанию, которая желает запустить спутник. Но вот для того, чтобы выполнить такое задание, нужно решить много-много проблем на низлежащих системных уровнях (их довольно много: скажем, из какого материала делать сопла двигателей? А чем охлаждать эти сопла? А как должны быть устроены насосы?), согласовать между собой взаимодействие всех частей ракеты, и частей этих частей, и так до уровня исходных материалов (из каких материалов делать насосы ракеты? А трубопроводы?). Чтобы сообразить, что много-много решаемых текущих проблем — это совсем не те решаемые проблемы, которые нужно решать (они неважные, их решение ничего не даст!), нужно подниматься на много-много уровней вверх — и затем спускаться опять вниз по системным уровням, чтобы выйти на действительно важные решения. Помним также, что конструкция системы должна отражать многоуровневую, а не одноуровневую оптимизацию. Глупо делать корпус ракеты из стали и иметь поэтому запас прочности корпуса при высоких температурах, а затем не использовать этот запас прочности! Все оптимизации конфигурации системы — многоуровневые.

Полезное упражнение тут — это представить авиалайнер как 6 млн индивидуальных деталей, летящих с одинаковой скоростью 890км/час в одном направлении на высоте 10 км. Это шутка, но это и чистая правда! Если не вводить тут рассмотрение на разных системных уровнях, оставаться редукционистом, то остаётся думать про самолёт именно так: 6 млн деталей, которые были собраны вместе и теперь согласованно летят на десятикилометровой высоте. Вспомните картинку аккуратно выложенных рядом деталей разобранного автомобиля, работу автомобиля невозможно обсуждать, обсуждая его детали и их взаимодействие! Невозможно организовать производство автомобиля или самолёта, если рассматривать одноуровневое разбиение «автомобиль — все его детали списком». Невозможно

[285] https://worldsteel.org/steel-stories/innovation/spacex-relies-on-stainless-steel-for-starship-mars-rocket/

организовать предприятие, если его рассматривать, например, как «предприятие — все его люди списком».

Если мы используем системное мышление, то мы сможем организовать работу примерно полумиллиона человек на самых разных заводах и в конструкторских бюро мира, которые изготавливают эту сборку из 6 млн деталей, которые мы называем «авиалайнер». Системное мышление использует системные уровни, чтобы не потерять ни одной части в системном разбиении, но каждый раз иметь дело с ограниченным числом частей-подсистем в каждой системе на каждом системном уровне.

Конечно, в случае авиалайнера и предприятия (и вообще по факту сегодня — любой другой системы) описание системного разбиения поддерживается компьютером, коллективное внимание команды этих проектов может быть удержано только с использованием компьютера. Даже если вернуться к бумажной инженерной работе, не удастся разработать и изготовить успешную систему таких масштабов. Поэтому системное мышление сводится по большому счёту к системному компьютерному моделированию, без компьютера удержать внимание в проекте просто нельзя, **системное мышление требует технически обеспеченной собранности. Не пишешь результаты моделирования — системно не мыслишь!**

Окружение, включая надсистему, а также целевая система и её подсистемы, входят в одно системное разбиение, уровни которого определяются на момент, когда целевая система существует, готова и работает, внося свой вклад в появление нужных системных свойств у надсистемы. Никакая система не нужна сама по себе, она нужна только для того, чтобы стать частью надсистемы! Поэтому системное мышление начинается всегда с того, что целевая система определяется как **«чёрный ящик»** (непонятно, как устроенный объект), выполняющий какую-то свою работу в надсистеме: рассмотрение от границы целевой системы начинается вверх по уровням, а не вниз/внутрь целевой системы, в состав её частей. Рассмотрение частей системы, знание о подсистемах — это рассмотрение **«прозрачного ящика»**, у которого известен состав частей и как они взаимодействуют. Рассмотрение устройства/состава целевой системы, то есть проход от границы системы вниз по системным уровням — это всегда второй шаг. Первый шаг — «чёрный ящик» в системном окружении, второй шаг — «прозрачный ящик». И так с системами каждого системного уровня — на много уровней вверх и вниз.

Создатели::система имеют дело/взаимодействуют с целевой системой по мере того, как она постепенно («разово» — это так мыслили раньше, сейчас системы эволюционируют, а не «рождаются, живут, умирают») замышляется, проектируется, изготавливается, ликвидируется. Если речь идёт о времени создания и развития системы, то системы, изменяющие описания и материалы целевой системы, чтобы получить из них готовую работающую систему, называются создателями/«системами создания»[286] (enabling systems по ISO 15288:2023, constructors в физике). Эти системы постепенно (в ходе «непрерывного всего» — непрерывной разработки, непрерывного изготовления, непрерывного введения в эксплуатацию) создают и развивают целевую систему. Создатели проводят

[286] в более ранних версиях руководства это были «системы обеспечения», enabling передавалось как https://en.wiktionary.org/wiki/обеспечение — ударение можно ставить двумя способами, обеспéчение или обеспечéние.

инкременты (части системы, содержащие новую функциональность/фичи/features/возможности) целевой системы через разные состояния (инкремент «замыслен», «спроектирован», «в виде закупленного сырья», «изготовлен», «проверен», «принят в эксплуатацию») к готовности эксплуатации/использования, а потом выводят из эксплуатации/использования, в том числе и ликвидируют (это может быть непросто! Например, оставляют «зелёную площадку» после полной ликвидации атомной электростанции), или вместо ликвидации ремонтируют, или иногда модифицируют и возвращают в использование. Подробней это рассказывается в руководствах по методологии и системной инженерии.

Создатели не входят в окружение целевой системы (не являются частями каких-то надсистем целевой системы) и тем самым не входят в одно системное разбиение с целевой системой, но входят в самые разные иные системные разбиения. Почему так? Сама целевая система не живёт, себя не кормит, не выращивает — для этого и требуются системы создания, сами работающие во время создания и развития системы. А окружение? Это все системы, которые окружают целевую в какой-то её конфигурации (помним, что система развивается/эволюционирует, поэтому конфигурация как состав системы меняется — посмотрите ваш телефон, он расскажет, какая его версия и какая версия операционной системы, какая версия его приложений) во время её работы/эксплуатации, когда она уже создана, работает, ещё окончательно не ликвидирована. Конечно, живые системы являются и создателями/родителями целевых (бабочка-1 родитель/создатель другой бабочки-2), но там есть особенности — геном (наследственный материал) находится в биологической системе и в создателе, и в целевой системе в ядре каждой клетки, а в техноэволюции мемом находится только в создателе (информационная модель системы), а то и распределён по многим создателям, а целевая система его не имеет.

В стандартах классической системной инженерии предпочитают о подобных системах говорить enabling system (например, в серии стандартов, основанных на ISO 15288:2023), но вот David Deutsch и Chiara Marletto предложили говорить о таких системах constructor[287] (одно из словарных значений как раз «создатель»): основное тут то, что подобного сорта система поддерживает свою идентичность в ходе каких-то однородных изменений, производимых создателем в окружающей среде. Например, молекула катализатора производит множество актов катализа, оставаясь неизменной (enable chemical reaction, слово «enable» в английском языке тут хорошо подходит, поэтому enabling system в английском языке не вызывает вопросов). Или станок производит множество деталей. Или преподаватель обучает/«изготавливает» множество студентов. Или фирма рубит в лесу множество деревьев и изготавливает из них множество досок. Или кошка рожает множество котят, а самореплицирующийся робот производит/изготавливает множество себе подобных «роботят» (эти эксперименты уже идут[288]).

У создателей, как и у любых других систем, есть свои надсистемы и свои подсистемы. Раньше системы создания называли системами ведения жизненного цикла (lifecycle enabling system). Иногда «lifecycle enabling system» переводят и как «системы обеспечения жизненного цикла», но слово «обеспечение» как перевод enabling часто путают с обеспечением/снабжением или с

[287] https://www.constructortheory.org/
[288] https://reprap.org/wiki/RepRap

неглавными/вспомогательными «системами обеспечения», поэтому в текущей редакции руководства мы не используем слово «обеспечение» для жизненного цикла, а также стараемся избавиться и от термина «жизненный цикл», который был одним из основных терминов прошлого поколения системного подхода: слишком много от этого ошибок.

Ещё раньше системы создания называли просто системами жизненного цикла, иногда даже сразу предприятиями (enterprise), ибо проектами создания чаще всего занимаются предприятия. Сейчас создатели упоминаются как системы, участвующие в проектах создания и развития систем в отличие от проектов, занимающихся эксплуатацией систем. «Создание» — это по факту однократное (хотя там внутри может быть множество попыток с разными прототипами, просто эти прототипы не попадают в целевое окружение, они ещё могут не иметь минимальной нужной функциональности) выполнение работ по выпуску системы как MVP[289] (минимальный жизнеспособный продукт, minimal viable product), а «развитие» — это отсылка к длительному/многократному практикованию метода изменения конфигурации системы с целью улучшения её приспособленности/fit к «эволюционной нише», то есть использованию в каких-то надсистемах. Поэтому «создание» — это чаще всего выпуск MVP, а «развитие» — это выпуск множества версий системы после того, как она создана, то есть выпущена версия MVP.

Создатель — это не только развитый интеллектуально агент (человек, робот) или даже коллективный агент (предприятие). Формально это может быть и просто какой-то станок, который вытачивает целевую деталь — эта деталь потом станет подсистемой работающей целевой системы. Станок не участвует в эксплуатации целевой системы, он не входит в её эксплуатационное/операционное окружение, не рассматривается вообще во время эксплуатации. И даже метод работы станка для получения ожидаемых (а не абы каких) результатов работы обычно так не называется, чаще говорят «функция». Зато он участвует в создании системы — так что этот станок будет создателем/constructor/enabling system/системой создания. И, конечно, станок будет входить в какое-то предприятие, но предприятие будет для этого станка далёкой над-над-над-надсистемой, через много системных уровней (например, системные уровни станок-сектор-отдел-служба-предприятие). А если рассмотреть станок-2, принимающий участие в создании станка-1, который принимает участие в создании целевой системы, то это будет фрагментом **графа создателей, в которой основное отношение — создания/enabling в системной инженерии или transformation в constructor theory, а не часть-целое/композиции**. В графах создателей могут быть и люди, и предприятия, и AI. Например,

- Консультант службы продвижения (внешний контрактор, предприятие) создаёт (выполняя работы по методам методологии и методики, это методы инженерии личности)
- учебный курс по продажам и
- преподавателя/наставника для «менеджеров по продажам»/продавцов.
- Преподаватель/наставник для менеджеров по продажам учит этих менеджеров по продажам (продавцов), используя материалы руководства/курса/учебника/регламента по продажам (создаёт мастерство продавца в агентах, которые будут дальше выполнять роли менеджеров по

[289] https://en.wikipedia.org/wiki/Minimum_viable_product

продажам)
- Менеджеры по продажам (то есть продавцы, но названные чуть более торжественно, «менеджерами») учат людей-сотрудников предприятия-будущего-клиента тому, как стать клиентом предприятия, и этого мало, они ещё и учат пользоваться продуктом, чтобы обеспечить постоянный доход от сервисных платежей и обновления версий продукта! Надо, чтобы после покупки люди ещё и пользовались продуктом! Так что продавцы создают в сотрудниках предприятия-клиента ещё и мастерство пользования продуктом.

Это очень маленький граф создателей клиентуры, в жизни такие графы содержат существенно больше узлов. Скажем, консультанта тоже должен кто-то «создать», например, контрактовать, а если у продукта на предприятии будет много пользователей, то надо научить их всех, для этого потребуется обучить преподавателя из числа сотрудников предприятия-клиента, а ещё надо как-то встретить сотрудников предприятия и продавцов — и вот это оказывается работой по совсем отдельным методам.

Важно, что целевая система определяется для множества ролей, для большого коллектива. Об этом подробней будет говориться в следующих разделах. А как называется какая-то система, которая прямо сейчас находится в центре нашего внимания, но не целевая? Если мы эту **нашу систему/system-in-hand** назовём целевой, то нас никто не поймёт, будут считать (справедливо), что мы тянем одеяло на себя. Например, **нам** (в том числе вариант «мне и мне», или даже «нас тут пять ролей в команде из троих человек») поручили разработать винтик для большого и сложного станка. Да, целевой системой **для всей команды** является отгружаемый клиенту станок, который войдёт в остальное оборудование завода клиента как системное окружение, но мы-то в центре внимания вот прямо сейчас в разговоре держим этот винтик — и до системного уровня окружения станка на заводе клиента нам как до Луны, слишком высоко! Что, нам считать станок над-над-надсистемой для этого винтика::система? Да, конечно. Считать винтик целевой системой? Нет, конечно. Целевая система уже есть, станок, это на каком-то среднем масштабе времени, всего проекта создания и развития станка, требующего кооперации и договаривания многих людей. Целевая система для этого и была введена как тип: это исходная точка для договаривания многих ролей, часто многих ролей, исполняемых командами агентов.

Для таких ситуаций рассмотрения отдельными агентами из больших систем создания, состоящих из множества цепочек создания и появляется вид системы, который называется термином **наша система** (engineered system в классической системной инженерии, managed system в системном менеджменте, MySystem, OurSystem, система в руках/system in hand). «Наша система» обозначает систему в частном/преходящем/вре́менном фокусе внимания в системном разбиении целевой системы в её окружении в момент эксплуатации или где-то в обширном графе создателей. Скажем, «целевая система — самолёт, наша система — пятый винтик в топливном насосе двигателя целевого самолёта», или «целевая система — самолёт, наша система — уникальный станок::создатель для нарезки «высокочастотного кабеля авионики»::подсистема этого самолёта», или «целевая система — процветающее трудолюбивое общество Остазии, но нашей системой сегодня является общество Евразии и Океании, которое мы всячески ослабляем и дестабилизируем».

Метафорически, если «целевая система» в большом коллективном проекте — это

«начало координат»/«Северный Полюс», то «наша система» (engineered system) — это конкретный какой-то географический пункт для нашей команды, чьи координаты определяются по отношению к этому началу координат. И мы:: «команда проекта нашей системы» должны чётко определить положение этого пункта по отношению к началу координат, определяя граф создателей, иначе будет трудно договариваться с другими командами в проекте: мы для них будем не «сотрудниками, работающими на общую цель, успешность целевой системы», а «заботящимися о собственном благе, пришедшими откуда-то зачем-то», с чем бы ни пришли. Дальше об этом поговорим подробней, но пока несколько примеров (и небольшие нюансы могут приводить к большим изменениям в этих примерах! Помним, что при рассмотрении вымышленных учебных примеров не мышление прикручивается к ситуации, а ситуацию крутят так, чтобы она подошла к произвольно взятому мышлению, так что аккуратней с обсуждением примеров — лучше берите реальные ситуации, которые не дадут вам изменять ситуацию вместо изменения мышления о ситуации):

- В парикмахерской целевой системой является причёска (платят деньги за причёски!), сама парикмахерская — система создания для причёски, наша система — это точильный станочек для заточки ножниц, который мы устанавливаем в углу парикмахерской. И горе нам, если заточенные на нашем станке ножницы плохо постригут клиента, и причёска будет из-за этого некачественной! Мы выявляем сценарии рационального использования станка, выбираем из разных вариантов видов станков, имеющихся на рынке, закупаем, настраиваем и т.д. — занимаемся «нашей системой». Кто это «мы»? В данном случае «я, работающий на должности завхоза парикмахерской, мне поручили этот проект».
- У нас мастерская инженеров-менеджеров (частная!), мы усиливаем интеллект, тем самым занимаясь производством мастерства мышления в незнакомых ситуациях, в которые попадают инженеры-менеджеры в своих рабочих проектах. Мы по факту изменяем свойства мозга наших инженеров-менеджеров так, чтобы в проблемных ситуациях включалось не бытовое мышление, а наш мастерство мышления по руководствам программ развития. Моя система — это та часть интеллекта, которая ответственная за системное мышление. Я как наставник по системному мышлению делаю этот кусочек мозга, который должен сработать в окружении кусочков мозга, вместе которые реализуют мастерство интеллект-стека фундаментальных методов мышления. А если я буду в роли разработчика полной программы развития, то у меня «наша система» — это весь усиленный интеллект, включая системное мышление, но не ограничиваясь им. «Наша система» в этом случае будет совпадать с «целевой».
- Наш небольшой отдел IT-компании занят производством софта поддержания общей среды данных. Этот софт используется в проектных организациях, занятых проектированием объектов капитального строительства (мостов, эстакад, электростанций и т.д.). Конечно, сам софт как таковой никому не нужен, мы по факту сдаём работающий софт с людьми. Поэтому «нашей системой» («мы» для этого «наше» — это наш небольшой отдел IT-компании) в этом проекте будет «служба управления инженерными данными проектной организации»::оргзвено, а целевой системой будет объект капитального строительства (например, мост). Нам придётся показать, как служба управления инженерными данными, которую мы помогаем сделать проектной

организации, поможет мосту стать лучше, дешевле, быть построенным быстрее. А для этого мы показываем, как подсистема нашей системы (софт, который производит наш отдел) выполняет уникальные функции в службе управления инженерными данными, которая выполняет уникальные функции в проектной организации, которая проектирует стройку моста и проектирует сам мост. И показываем, как наша работа связана с мостом, демонстрируем для этого документированный граф создателей. Тут важно, что наша система — это служба, а софт — лишь подсистема этой службы (обычно беда, если наша система — это софт, ибо проектом службы тогда непонятно кто занимается, а софт не будет принят, ибо принимать будут работу службы, в жизни изменения от работы службы, нет изменений от работы софта, если им некому пользоваться).

- … таких примеров множество. Речь может идти или о подсистеме целевой системы, или о каком-то создателе в их графе (иногда довольно далеко по путям в этом графе от целевой системы), или о подсистеме какого-то создателя в их графе. Нюансы тут важны, и помним, что «объективного» выбора нашей системы тут нет, и нет «пошагового алгоритма выбора»/«волшебного рецепта». Просто нужно учитывать, что во внимании удерживаем целевую систему, её надсистему, её подсистемы, а также системы в графе создателей. И ещё понимаем, где там наша система, за которую ответственны «мы» как «маленькая команда в большом проекте» (иногда это команда из одного человека, вас лично). И у нашей системы есть своя надсистема, свои подсистемы, свои системы создания: по отношению к ней мы тоже управляем вниманием в проекте её создания и развития.

Вопросы. Целевая система и соотношение систем

Девушка пришла к парикмахеру перед тем, как отправиться на вокзал встречать жениха. Поезд, проносящийся мимо платформы на всех парах, и растрёпывающий готовую причёску, по отношению к причёске входит в:
--- Целевую систему ### Нет, целевая система тут сама причёска — обсуждаем её изготовление и последующее поведение при эксплуатации.
--- Надсистему ### Нет, надсистема тут сама девушка (причёска её часть).
--- Создателя ### Нет, причёску делал парикмахер.
+++ Систему в операционном окружении ### Да, когда причёска была готова и эксплуатировалась (in operation), поезд оказался в её окружении и провзаимодействовал с ней.

Чем являются процессор, экран и оперативная память по отношению к смартфону?
+++ Подсистемы ### Да, это части смартфона

--- Надсистемы ### Нет, они не включают в себя смартфон.

--- Системы в окружении ### Нет, они не окружают смартфон, взаимодействуя с ним. Они внутри смартфона — они его части.

--- Системы создания ### Нет, они не занимаются созданием и развитием смартфона: замысливанием, разработкой, изготовлением или ликвидацией самого смартфона в его начальном виде и затем добавление в новых конфигурациях смартфона различных новых фич.

Чем являются процессор и видеокарта по отношению к оперативной памяти персонального компьютера?
--- Подсистемы ### Нет, они не части оперативной памяти

--- Надсистемы ### Нет, оперативная память в них не входит.

+++ Системы в окружении ### Да, во время работы (operations) они взаимодействуют с оперативной памятью, тоже находясь внутри компьютера.

--- Системы создания ### Нет, они не занимаются созданием и развитием оперативной памяти: замысливанием, разработкой, изготовлением или ликвидацией оперативной памяти в её начальном виде (MVP), а затем выпуском новых и новых конфигураций с новыми и новыми фичами оперативной памяти.

Представьте себе кассу супермаркета. Кто из нижеперечисленных является создателем кассового аппарата?
--- Кассир ### Нет, кассир не изготавливает и не чинит аппарат, он работает с ним (operations), внося вклад в его функционирование. Кассир тем самым в операционном/эксплуатационном окружении кассового аппарата.

+++ Ремонтник фирмы-производителя кассовых аппаратов ### Да, ремонтник изменяет кассовый аппарат, он система создания.

--- Поставщик чековой ленты для кассовых аппаратов ### Нет, поставщик чековой ленты находится в окружении кассового аппарата, он нужен в ходе функционирования (operations) готового кассового аппарата.

Организатор мероприятия готовит фортепианный концерт с использованием системного подхода. Он включает в свою целевую систему:
+++ рояль ### Да, рояль входит как часть в концерт как целевую систему.

+++ пианиста ### Да, пианист входит в концерт.

--- настройщика роялей ### Нет, настройщик роялей тут система создания, он один из тех, кто «изготавливает» концерт и его части.

--- рояльный ключ (для настройки рояля) ### Нет, рояльный ключ только готовит рояль к эксплуатации, в ходе концерта он не используется. Он из системы создания концерта.

+++ зрителей ### Да, для организатора мероприятия зрители — важнейшая часть концерта.

+++ зрительный зал ### Да, зрительный зал — это часть концерта.

--- программу концерта ### Нет, это описание концерта.

--- городской квартал, где расположен зрительный зал ### Нет, это окружение. Организатор мероприятия не изготавливает городской квартал, его интересы сосредоточены на концерте.

--- организатора мероприятия (себя самого) ### Нет, это система создания.

--- текст речи, которую он произнесёт для зрителей перед концертом ### Нет, «текст» указывает на то, что это какое-то описание (концерта или даже чего-то другого).

Настройщик роялей готовит рояль к концерту с использованием системного подхода. Он включает в свою целевую систему:

+++ рояль ### Да, это и есть его целевая система — он готовит к эксплуатации именно её, со всеми её подсистемами/частями.

--- пианиста ### Нет, пианист в окружении рояля — целевой системы настройщика. Пианист в настройке не участвует (хотя настройка делается для него).

--- настройщика роялей (самого себя) ### Нет, настройщик роялей тут в системе создания рояля, он один из тех, кто «изготавливает» рояль, готовит его к эксплуатации.

--- рояльный ключ (для настройки рояля) ### Нет, рояльный ключ только готовит рояль, он из системы создания рояля.

--- зрителей ### Нет, для настройщика рояля зрители лишь в рабочем окружении рояля, они не в его составе.

--- зрительный зал ### Нет, зрительный зал в окружении рояля.

--- программа концерта ### Нет, это описание концерта, в состав рояля не входит.

--- городской квартал, где расположен зрительный зал ### Нет, это совсем уже далёкий уровень надсистемы/рабочей среды (operation environment) для рояля — к настройке рояля это не имеет отношения (у настройщика нет интереса к городскому кварталу).

--- организатора мероприятия ### Нет, это одна из систем из графа создателей концерта как окружения рояля.

--- текст речи, которую организатор произнесёт для зрителей перед концертом ### Нет, это какое-то описание концерта — надсистемы для рояля.

Вася — разработчик программного обеспечения, осуществляющего подготовку трёхмерных моделей к печати на 3D принтере. Что является Васиной целевой системой?
--- Исходный код, записанный на диске его компьютера ### Нет, это описание его целевой системы. Целевой системой будет полученная из исходного кода исполняющаяся программа в момент её использования/эксплуатации.

--- Программа, запущенная на исполнение для тестирования в его лаборатории ### Нет, это не время эксплуатации (operation). Это модель (натуральная, всё равно как физическая модель самолёта для продувки в аэродинамической трубе) его целевой системы, а не сама целевая система. Ещё неизвестно, как поведёт себя программа при запуске её на компьютере разработчика 3D-модели.

--- Программа, записанная на диске компьютера пользователя 3D принтера ### Нет, программа для разработки 3D-моделей, а не печати этой модели на принтере. Но это не главное. Главное, что даже записанная на диск программа не работает, это только описание программы. Работает (то есть целевая система) запущенная на исполнение на этом компьютере программа.

+++ Программа, запущенная на исполнение на компьютере разработчика 3D-модели ### Да, это и есть целевая система для разработчика программного обеспечения, готовящего трёхмерные модели к печати на 3D-принтере.

--- Программа, запущенная на исполнение на компьютере, управляющем 3D принтером ### Нет, речь идёт не о печати, а о подготовке моделей к печати.

--- Компьютер пользователя 3D принтера ### Нет, это окружение для программы 3D-печати, а не для программы подготовки 3D-модели к печати.

--- Программно-аппаратный комплекс с 3D принтером ### Нет, речь пока не про печать, а о подготовке модели для печати.

--- Ничего из перечисленного ### Нет, это не так — правильный ответ в списке есть: программа, запущенная на исполнение на компьютере разработчика 3D-модели.

Вася является разработчиком программного обеспечения, управляющего печатью моделей на 3D принтере. Что является Васиной целевой системой?

--- Исходный код, записанный на диске его компьютера ### Нет, это только описание его целевой системы.

--- Программа, запущенная на исполнение для тестирования в его лаборатории ### Нет, это не время эксплуатации (operation). Это модель (натуральная, всё равно как физическая модель самолёта для продувки в аэродинамической трубе) его целевой системы.

--- Программа, записанная на диске компьютера пользователя 3D принтера ### Нет, это только описание, ещё не программа. Работающая программа лежит не на диске, а расположена обычно в оперативной памяти.

--- Программа, запущенная на исполнение на компьютере разработчика 3D-модели ### Нет, эта программа должна управлять принтером, а не помогать разработчику 3D-модели.

+++ Программа, запущенная на исполнение на компьютере, управляющем 3D принтером ### Да, это и есть целевая система — программа, управляющая печатью моделей на 3D принтере.

--- Компьютер пользователя 3D принтера ### Нет, Вася разрабатывает не компьютер, а программу. Компьютер тут по большей части в окружении (кроме той его части, которая совпадает с выполняющейся программой).

--- Программно-аппаратный комплекс с 3D принтером ### Нет, это надсистема: целевая система (управляющая принтером программа) является частью этого программно-аппаратного комплекса.

--- Ничего из перечисленного ### Нет, это не так, правильный ответ есть: программа!

Проектный институт «Главжелдормонтажавтоматика» ведёт проект по разработке системы управления путевой инфраструктурой на железнодорожной станции. Что является целевой системой проекта?
--- Система энергообеспечения станции ### Нет, это система в окружении — она не входит в состав системы управления путевой инфраструктурой, она снаружи её.

+++ Система управления всей путевой инфраструктурой на станции ### Да, это и есть целевая система. Что создаём (в том числе проектируем создание), то и целевая система.

--- Железнодорожная транспортная система ### Нет, это надсистема для целевой системы, которая создаётся (система управления путевой инфраструктурой на одной станции) причём на много системных уровней выше.

--- Команда проекта ### Нет, это система создания.

--- Объектный контроллер системы управления, реализующий логику безопасности верхнего уровня ### Нет, это подсистема целевой системы.

--- Железнодорожная станция ### Нет, это надсистема, система управления путевой инфраструктурой только её часть.

--- Ничего из перечисленного ### Нет, правильный ответ тут есть — это та система, проект которой разрабатывается.

Рекурсивное применение системного мышления: рекурсивное управление вниманием

Понимание того, что любая система входит в системное разбиение и принадлежит какому-то системному уровню, позволяет системному мыслителю применять одно и то же системное мышление рекурсивно/recursive: проводить одни и те же рассуждения для каждого системного уровня, для каждой подсистемы в этом системном уровне, если идти по системным уровням вниз от целевой, и каждой надсистемы, если идти по системным уровням вверх. Неважно, какая это система — для неё может быть (и даже должно быть!) развёрнуто полное системное мышление (как прописано в системной мантре). А потом, после всестороннего обдумывания ситуации с этой конкретной рассматриваемой системой, можно вернуться к системному разбиению в целом (время использования) и графу создателей, чтобы выбрать следующую систему для рассмотрения — и так с самыми разными системами самых разных системных уровней и самых разных узлах графа создателей.

Системное разбиение (system breakdown structure) — это прежде всего средство для управления вниманием. Внимание выхватывает для подробного рассмотрения какой-то один объект-фигуру, а всё остальное остаётся фоном, насколько огромным или разнообразным ни было бы это «всё остальное». Внимание позволяет резко упростить сложность мира, временно игнорируя незначимые детали — оставив в обсуждении только важное. Системное мышление заставляет всё время концентрироваться на главном, опуская неважные детали, это **концентрация внимания/focusing**. И оно же заставляет не забывать о целом, когда внимание обращается к частям — **деконцентрация внимания**. Помним, что основной акцент в системном мышлении — это «наверх» по системным уровням: не от целевой системы к её подсистемам, а от целевой системы к её окружению, к надсистеме! Если у вас во внимании какая-то фигура, то её окружение будет находиться в фоне! **Деконцентрация внимания, обращение внимание на фон оказывается даже более важным, чем игнорирование фона, фокусировка внимания!**

Нужна осознанность в управлении вниманием: вы осознанно концентрируете и деконцентрируете внимание, осознанно перемещаете это внимание с объекта на фон в моменты деконцентрации (много объектов в поле внимания, высокая алертность к появлению нового в широком поле вашего внимания), и концентрируете внимание, отслеживая находящиеся в его фокусе объекты на игнорируемом фоне. Системное мышление управляет вниманием, описывая его движение в понятиях/типах системного подхода (система, функция/метод, роль, конструктив, эмерджентность и т.д.). Понятия/типы системного подхода — это типы мета-мета-

модели для выявления/discovery и последующего удержания вниманием в окружающем мире (физическом мире и мире описаний) объектов этих типов.

Вы знаете, что нужно искать в физическом мире объекты с типами «система» разных видов (целевая, надсистема, подсистема, наша, создатель). Как эти типы относятся друг ко другу (какие между ними типы отношений) как раз и описано в нашем руководстве по системному мышлению, это и есть мета-мета-модель мира, выраженная текстом руководства на естественном языке (а потом, после освоения работы по руководству, это будет частью модели языка и мира, реализованной нейросетью вашего мозга, это наше руководство тем самым описание вашего будущего мастерства системного мышления).

Дальше вы активно ищете в жизни объекты этих типов, ибо уже имеете ожидания по поводу этих объектов. Активно — это расспрашиваете о состоянии этих объектов, придумываете и создаёте такие объекты, если убеждаетесь, что их нет, проверяете догадки рациональным рассуждением и даже экспериментами.

Концентрация-деконцентрация внимания в ходе системного моделирования соответствует движению внимания по системным уровням (уровням размера систем), а сам фокус внимания — это выбор конкретной целевой системы, её подсистем и надсистем, системы в каком-то месте графа создателей, выявление среди них «нашей системы».

Системный мыслитель хорошо ориентируется в сложном мире: ни на секунду он не теряет контекста/окружения рассматриваемого объекта внимания, оставаясь способным обсуждать как самый маленький винтик в самом маленьком приборе, так и совсем огромные системы планетарного масштаба. От этих «скачков масштаба» он не сходит с ума, для него это самая обычная процедура концентрирования внимания на всё более и более малой части мира, переставая воспринимать подробности из окружения этой части, или наоборот, деконцентрирования внимания на всё более большой части мира, теряя при этом подробности маленьких частей этой большой части мира.

Системный мыслитель выбирает/select какую-то систему, рассматривая её в составе надсистемы и в целом в системном окружении («в контексте», причём в момент работы готовой целевой системы), затем может рассмотреть эту систему в свою очередь как набор частей — «зуммировать»/«zoom in» на очередной уровень детальности, увеличив подробность рассмотрения этой части, как в современных фотоаппаратах. Когда рассматривают внимание, используется метафора «камер внимания» с zoom объективами:

- Внимание активно/деятельно: камеру сначала надо навести на какой-то кусок мира, который собираемся рассмотреть, затем её настроить на нужный зум. Это изменение состояния измерителя ещё перед замером: нужна работа по какому-то методу, чтобы направить камеру на объект! Измерение надо готовить, только потом измерять! Чтобы описать то, что за углом, надо подойти к этому углу и заглянуть за угол, сделать активное действие (и метод должен предусматривать, что важный объект действительно окажется за углом, а не, например, прямо сверху — и надо идти к углу и заглядывать за угол, а не задирать камеру прямо в вверх). И там, уже за углом, надо понимать, высматриваешь силуэт большой горы или песчинку на этой горе, настроить фокус: это задействование знаний, мета-мета-модели не только управления камерой внимания, но и знания о снимаемом объекте. Если тебе

нужно внимание на Петю, то плохо спутать его с Колей. Надо знать объект внимания!

- Только после того, как камера наведена приблизительно на объект, можно рассматривать то, что получено из этой камеры и выбирать какие-то объекты из её поля зрения, присваивать тип и проверять, на нужных ли объектах у нас внимание. Опознание наличия или отсутствия объектов каких-то типов может быть только после проведения измерения и при наличии знания об этих типах (знания мета-мета-модели).
- В общем случае камер у агента может быть много: внимание «многопоточно», с каждой камеры идёт какой-то поток информации, и работу каждой камеры надо настраивать и использовать мета-модель, а ещё нужно как-то объединять в мышлении результаты работы камер внимания.

Системный мыслитель может легко выбрать для каждой камеры внимания нужный масштаб рассмотрения ситуации, выбрать нужные ему системные эффекты (эмерджентные свойства) в качестве своих предметов интереса на выбранном системном уровне. И делает это системный мыслитель осознанно: он хорошо знает, что использует навигацию по системным уровням и при каждом мета-системном переходе (с одного системного уровня на другой) у него в рассмотрении появляются новые системные эффекты/эмерджентности.

Вот пример рассмотрения системных уровней для мультимодальной транспортной системы[290]:

В транспортной системе мы сначала можем обсуждать мульти-модальные[291] перевозки и конкуренцию независимых друг от друга мономодальных транспортных систем. Так, трубопроводный транспорт конкурирует в перевозке нефти с железнодорожным транспортом — для их владельцев они враги-конкуренты в операционном окружении, но для желающего перевезти нефть из одной точки мира в другую они части одной мульти-модальной транспортной системы (помним, что

[290] Leidraadse (2008), Guideline Systems Engineering for Public Works and Water Management, 2nd edition, http://www.leidraadse.nl/

[291] https://ru.wikipedia.org/wiki/Мультимодальная_перевозка

разные роли выделяют системы по-разному, как им удобно для их деятельности. Хотя для совместной работы в команде какого-то проекта им придётся договориться). Когда мы обсуждаем транспортные системы — это планетарные масштабы, или масштабы какой-то страны.

В одной из подсистем транспортной системы можно выбрать для обсуждения железнодорожную систему — поезда, энергетику железной дороги, управление движением поездов и т.п. Если взять одну из подсистем железной дороги — систему железнодорожной станции, то в ней можно дальше рассмотреть её собственные подсистемы — систему посадки пассажиров, информационную вокзальную систему, систему питания пассажиров (catering), систему продажи билетов. Часть этой системы продажи билетов — её подсистема автоматов по продаже билетов. Эти автоматы тоже каждый могут быть рассмотрены как отдельные системы. Винты, которые крепят печатную плату контроллера к корпусу этого автомата — это тоже системы. И даже в винтах можно найти разные части — головку с шлицами под отвёртки разной формы, резьбу.

Вот так, в паре абзацев и одной маленькой картинке мы проходим рассмотрение ситуации от планетарных или страновых масштабов до маленького винтика, и при этом не сходим с ума, не теряем нити рассуждений, чётко понимаем каждый раз предмет обсуждения и масштабы проблем. Это не поменяется и при обратном проходе, от частей винтиков до транспортной системы в целом. Перемещение по разным системным уровням, концентрация внимания на разных в них системах, деконцентрация внимания — это чрезвычайно мощный инструмент мышления.

Бессмысленно рассматривать винт в автомате по продаже билетов как непосредственную составную часть транспортной системы — это с точки зрения формальной логики будет правильно, но абсолютно бессмысленно, как «хвост стада коров». Системный подход, вводя системные уровни, делает рассуждения осмысленными: все люди получают возможность договориться, обсуждая проблемы только каждый на «своём» системном уровне (на уровне, для которого у них есть интересы, предпочтения, намерение действовать, чтобы изменить ситуацию к лучшему с точки зрения этих ролевых предпочтений), но при этом они учитывают проблемы смежных уровней — более высоких чем их целевые системы и более низких. Так организованное с разделением на системные уровни и по разным ролям коллективное мышление — это огромное достижение цивилизации.

Примерно так же мы можем обсуждать создание и развитие мультимодальной транспортной системы, указывая входящие в неё подсистемы разных уровней: кто::создатели и как::методы строит железнодорожную систему, кто::создатели и как::методы изготавливает винт крепежа платы контроллера к корпусу автомата по продаже билетов, кто и как завинчивает винт крепежа платы (это будет другой создатель, нежели создатель, изготавливающий винт). Всё обсуждается как «системы», и системы создания обсуждаются по отношению к системам в окружении, ибо если не знаешь, что::«целевая система или наша система» изготавливаешь — то тогда и не знаешь, какие системы-создатели это изготавливают, обсуждение невозможно. **Окружение сначала, целевая система и её устройство потом, методы создания («как делаем целевую систему») ещё позже, а системы создания (кто::роли, кто::агенты их играет, кто::менеджеры будет их организовывать) — будут рассматриваться последними.**

Боинг 747-8 состоит из 6 миллионов независимых видов деталей (входят в состав целевой системы), которые производили полмиллиона человек на 5400 фабриках (системы создания), за один год заказывалось (последний самолёт выпущен в декабре 2022) 783 миллиона частей самолёта (входят в состав целевой системы)[292]:

А теперь окружение этих боингов, время эксплуатации самолётов: аэропорты с авиадиспетчерскими и системами посадки-высадки пассажиров, воздушные коридоры (тоже системы! Они оборудованы радарами и другой инфраструктурой для их отслеживания. Они материальны, занимают место в пространстве). Сам по себе самолёт вне всего этого бесполезен, как пробка от бутылки бесполезна без бутылки, как бутылка бесполезна вне ситуации её использования.

В современных системах число отдельных элементов, которые нужно согласовать между собой (в проектировании), а часто и создать с нуля (в конструировании) достигает десятков миллионов в «железных» системах, а если речь идёт об электронных системах, то и триллионов: на одном электронном чипе Cerebras число отдельных транзисторов — 2.6 триллиона штук, при этом каждый транзистор имеет своё уникальное назначение внутри чипа, выполняет свою уникальную функцию[293]. И ещё эти чипы — не самый высокий системный уровень, выше их уровень печатной платы, ещё выше — суперкомпьютера на основе этих плат.

Можно оставить надежду о создании таких сложных объектов без какого-то их иерархического рассмотрения и управления коллективным и личным вниманием посредством документированной системной иерархии (то есть иерархии систем по отношению композиции). **Управление коллективным вниманием создателей в привязке этого внимания к системным уровням, выделяемым в иерархии по отношению композиции систем — самая важная часть**

---

[292] http://787updates.newairplane.com/787-Suppliers/World-Class-Supplier-Quality

[293] https://www.anandtech.com/show/16000/342-transistors-for-every-person-in-the-world-cerebras-2nd-gen-wafer-scale-engine-teased

системного мышления. Это коллективное внимание также направлено на отношения создания между создателями и создаваемыми системами, тут мы говорим о графе создателей.

Системные уровни появляются в результате роста сложности систем, это свойство эволюции (биологической/дарвиновской, меметической, техно-эволюции). Сложность систем будет только расти, это бесконечный рост. При этом каждый создатель-команда и создатель-предприятие может быть устроен достаточно просто (хотя это тоже системы, сложность создателей тоже растёт — предприятия объединяются в эко-системы, растут в суперхолдинги, число системных уровней там тоже увеличивается по мере эволюции). Но разбираться с проектами создания сложных систем (включая сами системы создания) можно потому, что одному создателю или даже его части (например, команде проекта внутри предприятия) не приходится заниматься всей системой в целом на всех системных уровнях — нет, каждый создатель работает с какими-то частями системы, и это существенно упрощает создание. Кто-то делает двигатель ракеты, кто-то делает компьютер ракеты, кто-то делает корпус ракеты, но нет фирмы, которая делала бы вот это всё — и даже выплавляла бы сталь для корпуса ракеты и очищала кремний для чипа компьютера.

Разные фирмы специализируются на разном мастерстве, но организованные согласно какому-то графу создатели — они вместе создают удивительно сложные целевые системы, работающие в удивительно сложном окружении. Всё это возможно благодаря системному мышлению инженеров и менеджеров этих предприятий-создателей.

Задания по системным уровням

Поставьте отметку о выполнении:

1. Написан пост с моделированием системных уровней мастерства вашего хобби (за основу поста взято описание из разделов с примером социальных танцев).
2. Написан пост с моделированием системных уровней вашего основного рабочего мастерства, (за основу поста взято описание из разделов с примером социальных танцев).

8. Графы создателей

Отношения создания

Простейший граф создателей метафорически (то есть весьма вольно) можно проиллюстрировать диаграммой:

На диаграмме целевая система (обозначена красным кружком, «начало координат» для системных описаний) находится в своём системном окружении/среде/environment, то есть входит в надсистему целевой вместе с другими какими-то системами. Надсистема обозначена объемлющим кружком в окружении, куда входят ещё другие кружки с кружками внутри — системы в окружении целевой с их подсистемами. Помним, что целевая система проходит техно-эволюцию. Она имеет свои подсистемы (кружки внутри красного кружка). Каждый уровень группировки частей в целой системе (обозначены как более мелкие кружки в объемлющих их более крупных кружках, и так на нескольких уровнях вложенности) — это системный уровень. Системный уровень подсистем целевой системы — три кружка-подсистемы в кружке целевой системы, два кружка вокруг целевой и сама целевая система внутри надсистемы — это системный уровень, на котором находится целевая система. Дальше мы не детализируем именно системные уровни, но просто отмечаем отдельные надсистемы.

Целевая система проявляет свои внешние свойства в надсистему «в ходе»/«во время» её работы/operations, это обозначено словами «феном[294] тут». Слова run-time часто используются в программной инженерии, чтобы обозначать время функционирования/работы/эксплуатации готовой системы. Но мы обозначили на диаграмме его другим, не менее употребимым словом — Ops-time, время работы операторов системы, когда система в рабочем состоянии, функционирует.

Целевые системы создаются и развиваются не сами (они саморазвиваться обычно не умеют, ибо не совсем живые, а в классической инженерии чаще совсем неживые), а создателями, выполняющими методы визионерства, разработки, архитектуры, DevOps/«инженерии внутренней платформы создания» и т.д. Системы создания включают в себя агентов с их инструментами. Агенты в узком смысле (автономные интеллектуальные вменяемые агенты) изображены кружочками с «лицами», а их явно неживое оборудование (от отвёртки до датацентра) — кружочками без лиц, организации изображены состоящими (композиция) из умных агентов и их оборудования и инструментов. Обратите внимание, что и во время

[294] https://ru.wikipedia.org/wiki/Феном

эксплуатации/operations какие-то агенты на нашей картинке входят в окружение (например, пользователь::внешняя-роль-из-надсистемы компьютера::система входит в его операционное окружение, в отличие от компьютерного завода, который будет создателем компьютера и будет рассмотрен во время создания/dev-time).

Эти агенты-создатели создают и развивают (то есть часто просто как-то изменяют состояние, а не создают с нуля — скажем, красят, настраивают, добавляют функции, ремонтируют) целевые системы, подсистемы и надсистемы целевых систем. Агенты в самых разных проектных ролях имеют предметами своего интереса самые разные эмерджентные характеристики самых разных систем на разных системных уровнях, для этого они пытаются как-то спроектировать и предсказать по проекту успешность целевой системы (в эволюции — fit/«соответствие нише»), для этого они делают «умные мутации» в ходе развития системы. И для этого они совместно редактируют мемом системы в ходе её создания и развития. Это обозначено словами «мемом тут» во времени создания (слова «и развития» мы просто опустили для краткости, но развитие не менее, а часто более важно, чем создание MVP) напротив слов «феном тут».

Техно-эволюция, как и биологическая эволюция, включает «развитие вида», а не только «рост/изготовление одного организма/«экземпляра продукта». На каждом шаге техно-эволюции есть проект/design текущего поколения продукта::система с уровнем детальности, достаточным для изготовления (например, инструкции для станка с ЧПУ, а также инструкции сборочному роботу на машинном языке ЧПУ и робота). Этот проект/design сам является развёрнутыми разработчиками изобретательскими идеями (то, как функциональные объекты реализуются конструктивными объектами, как они расположены в пространстве, сколько стоят), и вот они в биологии находятся в геноме, внутри целевого организма (но в генной инженерии ещё и в памяти создателей, например в компьютерах лаборатории!), а вот в техноэволюции аналог генома — мемом, хранится у создателей. И в ходе «умных мутаций» (потенциально успешных изменений в идеях мемома) создаётся целевая система с набором её особенностей в готовом виде: мемом порождает феном, как и в биологии.

Слова для времени создания и развития из программной/software инженерии — dev-time (от «development»), часто design time. При рассмотрении создателей мы не рассматриваем работающую/эксплуатирующуюся в окружении целевую систему, а рассматриваем систему в момент её создания (замысливания/«стратегирования использования», проектирования, изготовления, ввода в эксплуатацию системы как MVP, а дальше поток инкрементального развития функциональности и конструкции). Остальные методы времени создания (инженерные обоснования, принятие архитектурных решений) тоже присутствуют, но они тоже опущены для краткости, главное, что понятно: речь идёт о ходе/времени создания, dev-time, а не ops-time. Системы создания и системы из системных уровней внутри и снаружи целевой системы рассматриваются в разные времена/realms, что отражено пунктирной вертикальной красной линией, которую пересекают стрелки отношения создания.

Есть время готовки борща (создатели — повара, рецепт борща как мемом), есть время есть борщ (целевая система — борщ с его свойствами как феном, окружением тут выступают ситуация обеда в ходе подачи, рот-язык-зубы в ходе еды, а также желудок в ходе переваривания. Но это уже всё ops-time, а dev-time — это изменение

состояния свежих овощей, сырого мяса, воды создателями борща на кухне, до конечного состояния «борщ в тарелке, готов к использованию»). Есть время изготовления ракеты, есть время полёта ракеты. При этом работа инженеров с ракетой абсолютно не похожа на работу космонавтов в летящей ракете. **В системном мышлении принято чётко различать время, которое обсуждается, и главное время тут — использования/функционирования системы (все функциональные описания — в нём). Но есть ещё и время создания системы (все конструктивные описания — в нём), оно не главное, но тоже есть!**

Конечно, есть проблемы и единства рассмотрения этого времени, так называемая **проблема DevOps**, когда разработчики/создатели системы никак не связаны с операторами/пользователями и поэтому делают систему, которой невозможно пользоваться. Эта проблема решается прежде всего организационными мерами, но сегодня часто задействуют и технические меры: операторы и даже пользователи вообще исключаются как люди, заменяются роботами, так называемый подход NoOps[295]. В любом случае, в системном мышлении принято не столько считать всё происходящее в разработке и использовании принадлежащим к одному физическому времени, сколько различают «логические» времена создания (development, design, construction, implementation, enabling — везде в центре методы работы создателей, а целевая система тут пассивна, ещё не готова к работе) и времени эксплуатации (run, operation, use — функции/методы самой целевой системы, а создатели тут уже не работают, пассивны).

Можно тут обсуждать и цифровых двойников, но основная их роль — это «автоматизированное управление», замена оператора по настройке-подстройке параметров уже работающей целевой системы автоматом или составной конструкцией из человека (который крутит какие-нибудь ручки или меняет ненадёжные элементы конструкции в физическом мире) и информационной управляющей системы (которая говорит, куда какие ручки покрутить, что из элементов конструкции стало настолько ненадёжным, что хорошо бы заменить — софт занят «предиктивной аналитикой», например, для «ремонта по состоянию»). Цифровой двойник работает во время использования, а не во время создания.

Между системами в окружении (целевой, надсистемой, системами в составе надсистемы из ближнего окружения и т.д. — если встретилось слово окружение/среда/environment, надо всегда помнить, что это «операционное окружение»/«operations environment», то есть рассмотрение времени работы) и их создателями тоже отношение создания (development, design, construction, implementation, enabling в инженерии, в constructor theory — универсальное transformation), когда один создатель::система описывает и/или меняет другую систему. И таких систем можно рассмотреть целый граф по отношению создания, а если поглядеть на все такие цепочки как пути на графе, то это будет граф создателей: узлы — это системы (целевая и создатели) а рёбра — отношения создания. Внутри одного времени — отношения часть-целое/композиции, через границы времён/realms — отношения создания (X::система создаёт/«изменяет состояние» Y::система).

На диаграмме показан вариант такого графа создателей. Для каждого создателя тоже было его создание, и его эксплуатация/использование/работа/operations.

---

[295] https://ailev.livejournal.com/1367897.html

Поэтому на диаграмме представлено несколько разных «времён» рассмотрения (realms), и что для создателя будет его ops-time, для целевой системы будет dev-time. А что для создателя его dev-time, то для создателя создателя — ops-time.

Разных создателей может быть много, пути на графе создателей могут быть длинными. Можно в мышлении двигаться по этим путям довольно далеко от целевой системы: каждого создателя тоже надо кому-то создать, и при системном моделировании мы в каждом проекте просто останавливаемся на той длине рассматриваемого пути на графе создателей, которая позволяет более-менее уверенно оценивать успешность целевой системы.

Топ-менеджеры в своих проектах регулярно работают с цепочками создателей из графа создателей на шесть-семь звеньев — и когда берут какие-нибудь примеры на три или даже четыре звена, удивляются, что модель плохо соответствует жизни. Скажем, вы рассматриваете продавцов, но не учитываете, что реально вы в ситуации какой-нибудь дилерской сети и ещё с «агентами у клиента», а не прямых продаж — и вы их всех зовёте «продавцами». Всё, вы потеряли одно звено цепочки создания, модель будет плохой.

На диаграмме показан сереньким «создатель создателя создателя целевой системы», чтобы не забыть про наличие именно длинных цепочек создания, а не одного отношения создания между двумя системами. Стрелки направлены в среднем слева направо, это обычное умолчание для показа времени с прошлым слева и будущим справа: сначала как-то появляется создатель, и только потом — создаваемая им система. Вместе же все цепочки создания, моделируемые как пути на графе — граф создателей, обычно направленный/directed ациклический граф[296].

Отношения создания — это не отношения часть-целое! Enabling/construction это не composition/part_of! Кастрюля, в которой варится борщ — это не кастрюля в составе борща, или борщ в составе кастрюли! Это кастрюля для создания борща!

А теперь поставьте крестик на любом из кружков этой диаграммы, которым в составе большой команды проекта будет заниматься ваша маленькая команда (возможно, в ней будете только вы один) — это будет «наша система» (system-in-hand, engineered system, MySystem, OurSystem). И повторите все рассуждения про целевую систему для нашей системы — ни на секунду не забывая про целевую систему и отношения нашей системы и целевой системы!

Запутались? Запишите все эти системы в каком-нибудь редакторе текстов или другом моделере, как шахматист записывает шахматную партию. Думайте не «в уме», думайте над текстом, или аутлайном, или таблицей!

Системное мышление подразумевает использование моделей/описаний, внимание должно удерживаться не в мозгу мыслителя, а документами (сегодня — электронными, в том числе информационными и имитационными моделями, вчера — бумажными документами). Мышление — это всегда мышление письмом и письменным моделированием!

Наша диаграмма (тип изображения, в котором квадратики или кружочки обозначений объектов соединены стрелками для обозначения отношений) графа

[296] https://en.wikipedia.org/wiki/Directed_acyclic_graph

создателей в нашем руководстве используется исключительно в целях объяснения небольшого неизменяемого и никак не привязанного к проектам набора понятий. Она не будет модифицироваться в ходе проекта, она содержит очень мало деталей. Это иллюстрация в учебник, не рабочий инструмент системного моделирования. **Мы не рекомендуем диаграммное моделирование в рабочих проектах, но мы требуем вести в проектах обязательное системное моделирование в виде текстов, аутлайнов, таблиц: форматы, которые удобно менять/редактировать, производить в них поиск, наращивать их объём без боязни запутаться в хитросплетении связей[297].**

Документирование в системном мышлении важно. Внимание, которым управляют без записей, управляется ненадёжно. Люди забывчивы, поэтому документируйте/записывайте всё (всё-всё!).

Как моделировать изменения состояний важных объектов в проекте, будет рассказано подробно в руководстве по методологии.

Моделирование: граф создателей

Заполните табличку, строчки которой описывают пути на графе создателей, в которые входит «наша система». Сделайте это для трёх и более известных вам проектов (можно брать подпроекты одного большого проекта, можно брать независимые проекты). В табличке есть места для четырёх создателей, если вы готовы рассмотреть больше, пишите остаток цепочки в графе для четвёртого создателя в их цепочке.

Номер п/п	Название проекта	Целевая система	Наша система (может совпадать с целевой, либо быть её частью, либо входить в граф создателей)	Надсистема нашей	Первая в цепочке создающая система (непосредственно меняющая нашу)	Вторая в цепочке создающая система	Третья в цепочке создающая система	Четвертая в цепочке создающая система

Концепция использования

Знание о существовании различных видов систем (надсистемы, подсистемы) в их

[297] Подробней про системное моделирование на примере предприятий см. доклад А.Левенчука, 2022, "Как мы наладили обучение организационному моделированию", видео с 6:18:13, https://www.youtube.com/watch?v=lpgpnCoV14w&t=22693s, слайды https://disk.yandex.ru/i/ixpCGYY4kROrmA. Про вред от визуального моделирования см. книгу А.Левенчука «Визуальное мышление. Доклад о том, почему им нельзя обольщаться», 2018, https://ridero.ru/books/vizualnoe_myshlenie/

относительном положении от целевой системы в системном разбиении (указание на системное разбиение — это было указание на время использования) позволяет более строго/точно выделять целевую систему в мире. Понятие системы в физике как раз означает какую-то часть мира, отделённую границей от остального мира (окружения/среды, а когда говорят больше об описаниях/текстах, то используют слово **«контекст»**).

Мы будем выделять систему из мира вниманием, при этом **границу будем считать границей нашего внимания, а не какой-то материальной средой.** Так, компьютер берём нашим вниманием вместе с его корпусом (корпус — не граница системы! Граница проходит там, где молекулы корпуса кончаются и начинаются молекулы воздуха вокруг корпуса, и эта граница нематериальна, она «в уме», это граница внимания), дом вместе с его внешней стеной, кабель вместе с его оплёткой, клетку вместе с её мембраной.

Дальше мы вводим понятие «**чёрного ящика»** (black box): это какая-то система, которую мы представляем без знаний о внутреннем её устройстве — мы только можем описывать функцию::поведение «чёрного ящика»::система, проявляемое на внешней его границе, то есть на границе занимаемого системой места в физическом мире. Мы ничего не знаем о внутреннем устройстве, о подсистемах «чёрного ящика». А если мы заглядываем внутрь границы системы и говорим о том, как она устроена, то будем называть это **«прозрачный ящик»** (transparent box, иногда говорят «белый ящик»). Бывает и **«серый ящик»**: мы знаем очень немного про то, как устроена система внутри её границы, но всё-таки знаем.

Мы описываем систему как чёрный ящик минимально четыре раза, это и есть «системное рассмотрение»:

- **Функционально:** как роль (функциональный/ролевой объект) и его функцию во взаимодействии с окружением во время эксплуатации/работы/функционирования. Забивало — прикладывает усилие от руки к забиваемому острому предмету.
- **Конструктивно:** как конструктив, который мы создаём и развиваем во время создания. Молоток — вот этот, который мы купили в магазине (и будет реализовывать во время эксплуатации забивало).
- **Пространственно:** как место в пространстве, которое занимает этот чёрный ящик в момент эксплуатации. Тот объект, который лежит в верхнем ящике шкафа у правой стенки, а в момент эксплуатации на рабочем месте номер пять в помещении номер четыре.
- **Стоимостно:** Как совокупная стоимость владения чёрным ящиком. Вот эта штука, стоит 1000 рублей купить и практически нисколько эксплуатировать.

Важно, что все эти рассмотрения про один и тот же объект-систему и согласованы между собой, то есть они непротиворечиво описывают одну и ту же систему (это делается через 4D экстенсионализм — проверяется, что описываемый объект занимает одно и то же пространство-время), а ещё они не лезут внутрь системы (тут в примере мы не говорим, что там внутри забивала-молотка — не поминаем его части: ручку и боёк).

При системном рассмотрении мы учитываем дополнительно:

- **Граф создателей:** кроме рассмотрения системы как «чёрного ящика» в момент его работы, мы учитываем, что кто-то эту систему создаст и будет

развивать.

- **Эволюцию:** рассматриваем не только однократное первичное создание системы, но и развитие системы: «чёрный ящик» будет развиваться/модернизироваться, речь не идёт о каком-то однократном ходе его появления. Нет, будет MVP и множество инкрементов, **«никакая версия системы не является последней».**

«Описываем систему» — это или

- «прямая инженерия», то есть проектирование/design в части придумывания того, какая нужна система в составе надсистемы.
- «обратная инженерия»/reverse engineering уже существующей системы, если такое описание недоступно, но для чего-то нужно.

В части «прямой инженерии» не надо считать, что «мы ничего не знаем о системе, пока её нет, мы же не можем заглянуть в будущее» (упоминаем это только потому, что слышали это от многих инженеров-менеджеров). Почти вся инженерия — это проектирование/design систем, которых ещё не существует, но это не мешает описывать несуществующую систему, то есть придумывать гипотезы о том, какая несуществующая система будет успешна. Чтобы описывать несуществующую систему, нужно высказать гипотезу о том, какая будет эта система в 4D — придумать такую систему, которая бы выполняла поведение, приводящее надсистему к выполнению какой-то важной функции в будущем. Мышление о здании, которое построено будет в следующем году, вполне возможно — строители делали это и пару тысяч лет назад, никаких затруднений в том, чтобы представить «в уме» работающей в будущем какую-то систему, которая ещё не изготовлена. В прямой инженерии описывают систему, которой ещё нет в настоящем, это обычное дело, любое проектирование — как раз гипотезы о том, каким будет будущее!

Первое же системное рассмотрение — это рассмотрение системы как чёрного ящика на предмет выполнения им функции («нанесения непоправимой пользы») в надсистеме в ходе эксплуатации. Онтологическая модальность этого рассмотрения и в случае обратной инженерии, и в случае прямой инженерии — это модальность[298] веры/doxastic modality, то есть это гипотеза. Она может как выдержать проверку логикой и экспериментом, так и не выдержать, этим занимаются в рамках **методов инженерных обоснований**:

- **Проверка логикой: мы показываем, что в описании системы нет логических противоречий.** Если будет сказано, что наш «чёрный ящик» белого цвета вплоть до зеркальности, чтобы лучше отражать солнечные лучи и не перегреваться, и одновременно будет сказано, что наш «чёрный ящик» модного тёмно-зелёного цвета, чтобы выглядеть эстетично, то это явное противоречие, надо менять описание, устранять противоречие (повторим: неважно, описываем мы при этом уже существующую систему, или будущую). Инженерия в том и заключается, что люди в таких ситуациях **договариваются, устраняют противоречие**.
- **Проверка экспериментом:** мы можем выполнить какие-то измерения в реальном мире и проверить, совпадают ли в каком-то доверительном интервале эти измерения у уже существующей (изготовленной/воплощённой) системы с проектными значениями для неё.

[298] https://en.wikipedia.org/wiki/Modal_operator

В любом случае, мы и в прямой, и в обратной инженерии придумываем описание «чёрного ящика» как гипотезу (верим, что она верна), а потом критикуем эту гипотезу и в случае нахождения ошибок улучшаем, улучшаем и улучшаем эту гипотезу о «чёрном ящике» — работаем с этим описанием, добиваясь его непротиворечивости и точности в эксперименте (обратная инженерия) и в предсказании (прямая инженерия, но и тут будет эксперимент — создание системы и замеры на изготовленной системе для подтверждения гипотезы).

Описание в его функциональной части должно демонстрировать, что система «наносит непоправимую пользу», работая в составе надсистемы, в конструктивной части — что оно реализуемо (можно сделать!), в части пространственной — что система может поместиться там, где она должна работать, в части совокупной стоимости владения — что строить и эксплуатировать систему будет выгодно.

«Предпринимательская гипотеза» — это как раз оно, гипотеза о том, что наш «чёрный ящик» будет полезен и дёшев, поэтому будет хорошо продаваться и инвестирование в его разработку и изготовление прямо сейчас даст прибыль позже. Это **визионерство**::метод/практика (слово «предпринимательство» мы избегаем употреблять, хотя речь тут идёт строго о «предпринимателе по Шумпетеру», но предпринимательство мало кто понимает «строго как у Шумпетера», и каждый тут фантазирует что-то своё. Так что мы табуируем термин «предприниматель», практику «предпринимательство», а бытовое понимание того, что там происходит, раскладываем на несколько ролей. **Визионер**::роль — это и есть «предприниматель по Шумпетеру», но это не основатель фирм, не человек с особым складом ума, склонным к риску, не богатый человек, не изобретатель, ничего из обычных ассоциаций с «предпринимателем». Но он оценит, будет ли проект прибыльным: выдаст об этом гипотезу. И если он::роль будет считать, что прибыли не будет — проекта не будет).

Если мы меняем модальность системного описания «чёрного ящика» с доксической (веры) на деонтическую (запреты и разрешения, предписания), то описание чёрного ящика называют **системными требованиями** (system requirements). И раньше в конечном итоге разрабатывали именно системные требования, а сейчас разрабатывают **концепцию использования/concept of operations,** уточняемую и детализируемую до **сценариев использования/use cases** — они отличаются прежде всего вот этим онтологическим статусом, но не только. **В концепции использования и дальше в сценариях использования описывают поведение системы как «чёрного ящика», то есть описывают функции системы, её роль в окружении — описывают на статусе гипотезы, что это правильно угаданное поведение успешной системы.**

Примерно до 2015 года в системной инженерии даже был отдельный метод **инженерия требований**, сейчас его нет. Проверьте: до 2015 года в год выходил чуть ли не десяток учебников по вариантам метода инженерии требований, а потом как отрезало — по инерции ещё выпускают книжку в год, но это просто «старички» удовлетворяют спрос других «старичков». Более подробно эта история перехода от инженерии требований к разработке концепции использования рассказана в руководстве по системной инженерии.

Главное тут:

- Исполнители роли инженеров по требованиям (часто они назывались **аналитиками**) вклинивались между разработчиками и внешними

проектными ролями на основании идеи, что «не надо отвлекать разработчиков от разработки, да ещё эти разработчики с клиентами разговаривать не умеют, это должны делать специально обученные люди». Выяснилось, что эти «специально обученные люди» (аналитики) просто создают ситуацию испорченного телефона (клиент говорит одно, аналитик слышит другое, документирует третье, разработчик вычитывает из требований четвёртое), вреда это приносит больше, чем вред от «отвлечения разработчиков от дела», а время **«вычитывания из клиента» и «вычитывания из требований»** примерно одинаково. Дополнительно появляется задержка во времени на передачу информации через дополнительное звено и ещё утеря контекста и обоснований появления тех или иных требований.

- Требования появлялись как раз из концепций использования, которые постепенно детализировались. Самые разные требования для тех или иных фич медленно собирались вместе в огромный строго согласованный между собой «монолит» (как это сейчас говорят архитекторы), утверждались — и дальше этот «монолит» передавался в разработку для «безусловного удовлетворения». Вот это «собрать всё, утвердить, передать в работу» оказалось дико медленно, хотя и было сильно лучше исторически предыдущей ситуации, когда проектирование системы шло вообще без требований. Тогда было всё так плохо, что об этом лучше не вспоминать. «Много лучше с требованиями, чем без них» — это было чистой правдой! Но и с требованиями получилось не так хорошо. Первая задержка шла от согласования разнородных требований между собой: участвовали внешние проектные роли, инженеры по требованиям, разработчики, архитекторы. Эти требования утверждались, и все понимали, что они — огромная ценность, ведь в них вложено огромное количество труда, их нельзя менять, они же с таким трудом были согласованы! Поэтому при обнаружении очевидных ошибок в требованиях, или при обнаружении изменения ситуации, при которой надо бы было менять требования — требования предпочитали не менять. И делали систему заведомо хуже, чем могли бы сделать. То, что «у нас есть процедуры изменения требований» — это фикция и отговорки, эти процедуры были запретительно дорогими по времени, к тому же за невыполнение требований наказывали, а за выполнение кривых требований вроде как наказать нельзя.
- После того, как требования попадали к разработчикам, они пытались выполнить обратное действие: понять, что там реально будет полезно внешним проектным ролям, которых разработчики в глаза не видели, а видел их только аналитик. При этом беда была не только в том, что при обнаружении ошибки в требованиях было сложно их менять, но и в том, что сами разработчики считали, что они должны с этими требованиями сработать однократно, а испытания планировались не для того, чтобы продемонстрировать пригодность системы для клиента (потом разобрались, назвали их **приёмкой/валидацией/validation**), но для показа того, что «требования удовлетворены» (эти испытания назвали **проверкой/верификацией/verification**). Проблема была в том, что замысел, проектирование, разработка, изготовление, испытания (и проверка, и приёмка) считались однократными. Это приводило к невозможности улучшения системы: ни оперативно добавить новую фичу, ни оперативно

исключить ненужную. Неоперативно и крайне нервно — можно, «есть процедуры изменения требований». Но вот оперативно — нет, нельзя. Только вслушайтесь: «у нас неверная гипотеза, давайте её по-быстрому поправим» против «нам дали неверные требования, давайте не будем эти требования выполнять». Системы, которые делались на базе концепций использования и детализации до сценариев использования, **оказывались лучше** из-за того, что и концепцию использования, и дальше сценарии использования вроде как можно править по ходу разработки, «в рабочем порядке», если нашлись проблемы, а вот требования «утверждены» и поэтому править можно их только «в особом, медленном и трудоёмком порядке». Ну, закалённым инженерам старой школы этот «медленный и трудоёмкий порядок» был нипочём, а инженеры новой школы лишь усмехались и тихо говорили «для бешеной собаки семь вёрст — не крюк», а потом демонстрировали невероятные для «старичков» скорости разработки.

- Работа с концепциями использования и сценариями использования, по которым требования не разрабатывались, а которые использовались разработчиками непосредственно, происходила ещё и быстрее: разные сценарии использования правили и реализовывали разные команды людей-разработчиков, а чтобы результаты работы этих людей объединились между собой — за этим прислеживали архитекторы (понятие архитектуры тоже изменилось, ибо оказалось нельзя изменить только работу с требованиями). Это означает, что командам не надо было ждать друг друга, пока соберутся и затем утвердятся все требования в одном «монолите». Нет, все высказывали свои гипотезы в виде концепции использования (описывали систему как чёрный ящик), критиковали их, проверяли путём создания части системы, реализующей сценарии использования, затем улучшали эти сценарии — и так система создавалась и развивалась непрерывно, бутылочное горлышко «собрать все требования вместе и потребовать их однократного выполнения» исчезло, а работа по созданию системы была распараллелена не только в части собственно проектирования, но и в части понимания того, что должно быть спроектировано — что там за поведение (функции) должен выдавать «чёрный ящик».
- Отказ от требований привёл ещё и к тому, что начали активно использовать A|B тестирование[299], когда гипотез выдвигается сразу несколько (требования обычно требуют что-то одно!), и проверяются они все вместе, а потом выбирают вариант, который оказался по каким-то критериям лучше. **Если у тебя «гипотезы», а не «требования», то ты с ними поступаешь другим образом: не столько «удовлетворяешь», сколько «проверяешь и постепенно корректируешь».**

Концепция использования (а раньше — сделанные на её основе требования, от разработки которых отказались) прежде всего содержит информацию о функциях системы по отношению к её рабочему/целевому/операционному/функциональному окружению, поэтому она состоит из самых разных моделей, которые описывают поведение системы на её границе во взаимодействии с системами снаружи (системами в составе надсистемы). Наиболее подробные модели поведения называют сценариями использования. В некоторых школах системной инженерии сценарии использования считают отдельными от концепции использования (ибо

[299] https://en.wikipedia.org/wiki/A/B_testing

они разрабатываются позже сжатых описаний функциональности системы в концепции использования), в некоторых — входящими в концепцию использования, просто сама концепция использования потихоньку меняется в ходе проекта: она конкретизируется, уточняется, детализируется, в неё входит всё больше всё более детальных сценариев использования по мере развития системы. Мы принимаем второй подход: сценарии использования входят в состав концепции использования, это один из видов моделей, которые в неё входят. Подробней об этом — в руководстве по системной инженерии и предлагаемой им литературе.

Когда говорим о концепции использования как об описании «чёрного ящика», то говорим именно про описание времени эксплуатации в части получения необходимой функции от системы. Система **должна бы** при эксплуатации в части её поведения или осмысленно мигать лампочками, или давать какой-то результат вычисления, или нагреваться, или оставлять бороздку правильных размеров: всё, что предполагается, что будет делать система (гипотезы!), чтобы в момент её эксплуатации говорили, что она делает то, что от неё ожидают и не делает того, что от неё не ожидают, то есть описывается ожидаемое поведение успешной системы. **Ожидаемое поведение системы — это предсказание того, что должно бы произойти при эксплуатации. В случае требований раньше говорили поэтому о функциональных требованиях (деонтика), а сейчас это просто гипотезы о функциональности (докса), поэтому нет «должна», только «должна бы» (а жизнь потом покажет, оправдалась ли гипотеза).**

«Нефункциональных требований» не было, хотя такой термин часто встречался в литературе, но обычно разъясняли, что его использовать неправильно. Чаще говорили просто о других видах требований — например, требованиях качества (-ости/-ilities, типа доступн**ость**, ремонтопригодн**ость**, надёжн**ость**), которые интересны не только разработчику, но и другим ролям в проекте, прежде всего роли архитектора. Сейчас стало очевидно, что архитектура имеет дело с архитектурными характеристиками (прежде всего эти -ости), но они относятся к системе не в части выполнения своих прикладных функций, а к общему какому-то поведению и в момент работы/operations (например, характеристики надёжности работы) или даже на момент создания и развития (например, характеристики возможности лёгкого изменения в ходе непрерывных улучшений, evolvability). И эти характеристики хотя и разные, но всё-таки более-менее одинаковые у самых разных видов систем. Например, масштабируемость/scalability: насколько легко поднять производительность системы, если это надо — нужно будет только добавить какие-то дополнительные модули (скажем, добавлять ещё пару колёс на каждую новую тонну грузоподъёмности тележки, или придётся всё разрабатывать по-новой с самого начала — ибо для увеличения грузоподъёмности менять надо в тележке и раму, и колёса, и вообще всё?). Архитектурные характеристики — описание поведения системы при работе в необычных условиях или не в момент эксплуатации: способность работать под высокой нагрузкой, ремонтопригодность, доступность по цене/affordability (помним, что совокупная стоимость владения затрагивает и стоимость расходников и обслуживания при эксплуатации, но ещё и стоимость изготовления включает не только стоимость материалов, но и стоимость работ системы создания, то есть затрагивает описание ещё и систем не из окружения, а из графа создателей), лёгкость монтажа. Руководство по системной инженерии подробно разбирает работу архитектора, архитектурные характеристики, способы достижения приемлемых значений для метрик, которыми измеряют архитектурные характеристики, даёт современную дополнительную

литературу на эту тему.

Конечно, во многих и многих моделях пересекаются и концепция использования (что делает система на её внешней границе, чтобы вписаться в окружающие её системы в момент её работы, принося какую-то пользу этому миру), и архитектурные характеристики (надёжность, лёгкость изменений системы разработчиками, устойчивость к разрушению, и прочие характеристики, непосредственно не связанные с функциями системы), и **концепция системы** как описание внутреннего устройства системы, «прозрачного ящика». Как концепция использования при детализации превращается в набор сценариев использования, так концепция системы при детализации превращается в **проект/design системы**, который уточняется вплоть до описаний прозрачного ящика, достаточно детальных для изготовления системы.

Если говорить о таком же наборе описаний, но на уровне надсистемы (скажем, описывать не концепцию использования шестерёнки в механических часах, а концепцию часов с шестерёнкой в их составе как концепцию системы), то можно обнаружить, что концепция использования шестерёнки по факту — часть концепции часов, ибо в устройстве часов важна шестерёнка и что она делает безотносительно устройства самой шестерёнки.

Конечно, терминология для всех этих описаний чёрного и прозрачного ящика может меняться, особенно если смотреть не на описания, а на документы, выражающие эти описания: легко концепцию использования насоса, описывающую поведение насоса, назвать «опросным листом», рассылаемым поставщикам насосов с вопросом «можете ли вы изготовить насос с вот такими характеристиками?» В «опросном листе» не будет ни слов «концепция использования», ни более древнего слова «требования». Если и будут какие-то указания на то, что именно должно бы быть в составе насоса, то не будет и слов «концепция системы». Архитектурные характеристики (типа надёжности) будут или не будут указаны, название документа будет «опросный лист». Вы должны уметь поглядеть на такой «опросный лист» — и определить тип описания: «опросный лист»::«концепция использования» насоса (а если там есть архитектурные характеристики, то отметить и их тоже).

Почему нужно выделять отдельно описание чёрного ящика как концепцию использования (что делает система вовне, от границы системы в окружение), а не сразу рассказывать про концепцию системы — как она устроена внутри себя (какие подсистемы выполняют какие функции и из чего они сделаны)? **Различение чёрного и прозрачного ящиков нужно, чтобы иметь возможность предложить самые разные варианты того, как устроена система, из чего она сделана — предложить разные аффордансы для выполнения функции системы.**

Скажем, концепция использования дверного ключа будет описывать, что ключ переводит дверь из состояния «заперта» в «отперта» и обратно. Это будет главное, функция ключа. Архитектурные характеристики (срок службы до первой поломки, ремонтопригодность и т.д.) тут не так важны, важно для чего нужен ключ, какие изменения (поведение — это изменения, «в один момент времени было одно состояние, в другой момент времени стало другое состояние») вызывает система в своём окружении. Тут важно, не какие изменения вызывает окружение в системе, а какие изменения вызывает система в окружении. Подробней об этом будет говориться в руководстве по системной инженерии, рассказ о действиях системы на

окружение — в том числе реагировании на воздействия извне, а не о действиях окружения на систему.

Если вы подробно описали ключ и его функциональность как «чёрный ящик», дальше вы можете предлагать разные конструкции ключа, которые оказываются неразделимыми с конструкцией замка. То есть вы понимаете, что ошиблись и меняете рассмотрение ключа::система на замок::система, в который входит ключ::подсистема, а уже замок::система входит в дверь::надсистема. Далее рассматриваете варианты

- механического замка с поворотным ключом,
- механического замка с нажимным ключом,
- электрического замка с программируемым электронным ключом-«таблеткой»,
- электрического замка с управлением через приложение в смартфоне,
- электрического замка с управлением через биометрию (отпечаток пальца или распознавание лица), и так далее: вариантов много, можно выбирать.

Главное — это соблюсти функцию замка (ибо оказывается, это делает не ключ, а замок) в его окружении: перевод двери из состояния «заперта» в состояние «отпёрта» и обратно. А дальше рассматриваем и ключ тоже, только там будет перевод замка из состояния «заперт» в состояние «отпёрт» и обратно. И вот тут возможны варианты: например, перевод замка в состояние «заперт» может быть сделан ключом, а может быть сделан захлопыванием двери. В замке может быть переключатель режима захлопывания двери и запирания ключом, но и ключ может быть хитрым — этот переключатель может быть в ключе, если ключом служит приложение в смартфоне!

Всегда рассматриваем вначале функцию::поведение системы в мире, потом выявляем аффорданс для этой функции (и тут возможны варианты, и легко можно обнаружить смену целевой системы, как в примере с ключом: оказывается, надо было рассматривать замок как целевую систему, а ключ как подсистему). По возможности откладываем рассмотрение того, как устроена система как роль/«функциональный объект», ибо если непонятно, что система должна была бы делать (непонятная функция системы) — бесполезно обсуждать устройство системы, которая будет выполнять эту функцию.

Концепция использования — это набор моделей системы, модели абстрактны и не явлены никак в физическом мире. Чтобы на них посмотреть, или кому-нибудь переслать, нужно иметь их на носителе (неважно, бумажном, электронном или оптическом), то есть иметь документацию концепции использования (а если вы находитесь в предприятии, где придерживаются идей уже устарелой системной инженерии, то вам потребуется ещё и для концепции использования разработать требования и документировать уже требования). Документация концепции использования может найтись (или быть подготовлена) в самом разном виде: документы стандартов, технические задания, файлы с диаграммами сценариев использования на каких-то языках, фрагменты базы данных с информационной моделью. Если во всех этих документах содержится описание поведения целевой системы как «чёрного ящика» в окружении в момент эксплуатации, то это и будет концепция использования. И если вы при этом слышите слово «требования», то первым делом интересуйтесь насколько это гипотезы, а насколько и впрямь «очень

надо именно так, можем обосновать», интересуйтесь сценариями (поведением), и проверяйте, чего в требованиях может не хватать или быть искажено, если их готовили «аналитики». Бойтесь «испорченного телефона», поинтересуйтесь ситуацией использования системы в ходе её работы, поймите сценарии/поведение, не принимайте «требования» слепо на веру — помним, что это просто «гипотеза, которую утвердили и назвали требованием, а потом гипотеза может оказаться неверна — и что будем делать?». Не верьте, если вас будут заверять, что «это требования, выполните их, и всё у вас будет хорошо». Это рассуждения из старой системной инженерии, старого системного подхода.

Концепция системы и архитектура

Очень часто те люди, которые описывают систему, хотят указать не только внешние свойства системы, описать не только границы системы и её поведение как чёрного ящика, но заодно и указать какие-то детали внутреннего устройства системы: определить (define) части системы (подсистемы), указать на процесс взаимодействия подсистем. В этом случае о системе говорят как о «**прозрачном ящике»,** в нём можно считать известными какие-то подсистемы, свойства и поведение этих подсистем. Это можно считать указаниями по концепции системы: какое поведение ожидается от внутренних частей системы (время их работы, функции подсистем), какими частями конструкции системы (конструкция подсистем) это поведение может быть реализовано, при этом важно ещё учитывать и пространственные аспекты, и аспекты полной стоимости. **Концепция системы** описывает «прозрачный ящик» — это минимально четыре взаимоувязанных обязательных вида системных описаний, но чаще всего этих видов описаний по разным предметам интереса больше.

В любом случае, начнём хотя бы с двух: функциональное и конструктивное описания в их взаимоувязке. Скажем, вы готовитесь изготовить дверь, и вдруг вам внешние проектные роли говорят: «замок в двери должен быть из титана».

Слово «должен» указывает на требования, а не гипотезу, и это настораживает. А вдруг вы найдёте материал лучше? Для чего это вдруг потребовалось указывать, чего в поведении двери боится тот, кто описывает эту дверь? Какой испорченный телефон тут сработал, кто задал такой сценарий использования, что появилось это требование? Откуда вообще в двери замок, ибо вы и не подозревали, что дверь должна была бы запираться и отпираться: этого не было в концепции использования. Мы понимаем, что это описано **ограничение** свободы разработчика: один разработчик::роль указал другому разработчику::роль, что там должно бы быть внутри системы, чтобы при разработке концепции системы это ограничение было учтено. Вопрос: откуда разработчик::роль взялся во внешних ролях проекта. Иногда разработчиков целевой системы на зарплате у организации-заказчика называют «инжиниринг заказчика», и им платят зарплату как раз для того, чтобы они создавали такие ограничения свободы внешних разработчиков — чтобы, например, у внешних разработчиков не возникало соблазнов излишне удешевить-упростить или задрать цену-усложнить конструкцию системы.

Обычно организация проекта (например, коллектив из нескольких команд) согласовывает с внешним заказчиком системы функции/поведение, которые должна бы выполнять целевая система как чёрный ящик, свойства этого чёрного ящика, т.е. разработчики согласовывают концепцию использования (чаще всего даже не всю, а какие-то отдельные сценарии использования по отдельным «фичам»,

которые будут реализовываться какой-то отдельной командой), а архитектор системы согласовывает архитектурные характеристики и их приемлемые значения для всех команд разработчиков. Не факт, что эти гипотезы по концепции системы и приемлемым архитектурным характеристикам формулирует заказчик! Это совместная с заказчиком работа. Как устроена система внутри, то есть какие функции выполняют подсистемы и какими физическими объектами реализована конструкция системы — это коллектив проекта определяет самостоятельно, заказчика больше интересует обычно надсистема, она для него «прозрачный ящик», а целевая система — «чёрный ящик». Более того, это часто делается не за один приём, а долго — и продолжается даже после того, как первая версия (MVP, minimal viable product) уже изготовлена и начала использоваться. Часто самые интересные идеи и у агентов-заказчиков в самых разных их ролях, и у агентов-исполнителей этого заказа будут появляться уже после появления MVP, в ходе последующего развития системы.

Решения «прозрачного ящика» по тому, как система делится на конструктивные части (модули) и как эти модули будут взаимодействовать, чтобы значения архитектурных характеристик удерживались в приемлемых пределах (например, система в ходе улучшений продолжала оставаться легко изменяемой) называют **архитектурой.** Обратите внимание: архитектура — это решения по устройству системы в части конструкции (модули/конструктивы и способ их связи), равно как и концепция системы — это тоже решения по устройству системы, но включают и функциональные объекты, и конструктивные как аффордансы для этих функциональных объектов, а также пространственные (компоновка/размещение/location/allocation) и выходят на совокупную стоимость владения, и ещё могут включать какие-то важные другие аспекты (скажем, отдельно выделенные этические и юридические рассмотрения, или эстетические и художественные рассмотрения, в каждом проекте могут появляться какие-то особенности).

Концепция системы — это не архитектура (хотя ещё десяток лет назад считалось, что концепцию системы разрабатывают архитекторы), а архитектура — не концепция системы. Функциональность системы в надсистеме определяется решениями в концепции системы, а вот важные характеристики вроде «способность системы к развитию, лёгкость внесения изменений в конструкцию» определяются архитектурными решениями.

Архитектурные решения/architecture decisions оформляются самыми разными способами, например, описаниями с предписанной структурой на нескольких страницах в виде **записей архитектурных решений/architecture decision records**. Архитектурные решения представлены списком таких записей, или какой-то таблицей с материалом таких записей. Архитектура — это абсолютно не «диаграмма». Диаграммы в проекте чаще всего не архитектурны, а если и обнаруживаются где-то в архитектурном решении, то оказываются использованы в чисто иллюстративных целях, а не выражают архитектурные решения. Не надо путать **системную архитектуру** со **строительной архитектурой**, где прежде всего речь идёт о внешнем виде здания, а главное для архитектора умение — это умение рисовать.

Подробней об архитектурных решениях и их записи говорится в «Системной инженерии», но главное пока запомнить: **вы ничего не можете сказать про «прозрачный ящик» (ни обсуждать концепцию системы, ни обсуждать**

архитектуру системы), пока вы не выясните, что там с «чёрным ящиком», то есть поведением системы в её окружении. Если вы не знаете, что делает сепулька, то вы ничего не сможете сказать о том, как она может быть устроена!

Это главное, что нужно запомнить в системном мышлении: первое же рассмотрение системы — это концепция использования, а не концепция системы, не архитектура системы. Сначала смотрим от границы системы в окружающий мир, в среду, причём в момент эксплуатации (если вы разрабатываете систему, то это надо вообразить, ибо эксплуатация будет в будущем), и рассматриваем поведение/функцию системы. А потом только начинаем смотреть на прозрачный ящик. Если мы начинаем рассматривать шестерёнки в механических часах, а потом вдруг смотрим в окружение часов, то можем увидеть, что часы эти расположены в ракете, и должны бы отсчитывать миллисекунды, штатно работая с перегрузками в 30G, ибо там всё вибрирует рядом с двигателем, около которого установлены эти часы. Можете забыть с этого момента про шестерёнки, это было зря потраченное время! **Сначала смотрите вокруг, от границы системы наружу в момент эксплуатации, а потом только внутрь системы в момент эксплуатации, и потом только внутрь системы в момент её создания!**

В любом случае, все эти описания системы (концепция использования, концепция системы и архитектурные решения) — гипотезы, готовьтесь их менять, торгуйтесь, предлагайте свои варианты, внимательно выслушивайте варианты, которые предлагают внешние проектные роли, обязательно занимайтесь обоснованиями. «Я так хочу» — это не аргумент, ибо сейчас хочу так, завтра хочу по-другому, или «вы не так меня поняли, я хотел другого», или вас в этой роли подменит какой-то другой агент, и он будет хотеть по-другому — это же «хочу» должно быть информированным мнением роли, а не простым желанием агента.

Моделирование: концепция использования и концепция системы

Запишите для целевой или вашей системы сверхкраткую концепцию использования («чёрный ящик») и сверхкраткую концепцию системы («прозрачный ящик»). Важно, чтобы вы отличали: описание «чёрного ящика» времени использования и описание «прозрачного ящика» времени создания и развития. Полнота и точность описаний тут не важны.

Система	Концепция использования	Концепция системы
Пример. Компьютерная мышь	Компьютерная мышь работает на ровной поверхности, к компьютеру присоединяется через радиоканал, от компьютера может быть в работе на расстоянии до 15 метров, работает полгода без перезарядки, использует стандартный драйвер мыши	Компьютерная мышь имеет питание, реализованное батарейным отсеком и сменными батареями, координатное устройство, реализованное лазером и лазерным датчиком с контроллером, корпус под выбранную руку (правую или левую) из

	операционной системы, рассчитана на пользование правой рукой, скролл инерционным колёсиком, шум от правой и левой кнопки мыши незначительный, но есть тактильное ощущение «клика».	ударопрочной пластмассы, механизм скроллирования с инерционным колёсиком и оптическим датчиком вращения колеса, средства скольжения по поверхности — тефлоновые накладки на корпус, радиопередатчик для радиоканала на интегральной микросхеме и радиоприёмник на интегральной микросхеме, вставляемый в разъём USB компьютера.

Рекурсивное рассмотрение в системном мышлении

Если рассмотреть надсистему нашей целевой системы как чёрный ящик, то она будет целевой для команды важнейших внешних ролей проекта. И описание надсистемы как чёрного ящика будет для этих внешних ролей концепцией использования их целевой системы, которая для команды — целевая.

Когда из концепции использования ещё делали требования, то это описание часто так и называли: **требования внешних проектных ролей** (stakeholder requirements/stakeholder needs, не путать с system requirements, **системными требованиями**, которые называют просто **требования**, опуская слово «системные»), но чтобы меньше путать это описание надсистемы-как-чёрного-ящика с описаниями целевой системы-как-чёрного-ящика, это описание надсистемы как чёрного ящика чаще называют **потребностями** (stakeholder needs, нужды/потребности внешних проектных ролей), а ещё всё чаще и чаще называют их **ролевыми интересами или предпочтениями**. Скажем, мы::команда делаем шестерёнку к часам, то есть целевая система команды — шестерёнки. Потребности/needs — к часам::надсистема, а сценарии в концепции использования будут для шестерёнки::«целевая система».

Тем самым в проекте обычно **в концепции использования описываются два «чёрных ящика»**:

- **надсистема** — описывается интересами/предпочтениями в каких-то характеристиках надсистемы, важных для внешних проектных ролей (потребностями, stakeholder needs). По предметам интереса к надсистеме могут быть разные предпочтения, вот надо сделать гипотезу о том, какие значения характеристик всех устроят, а дальше не бояться корректировать эту гипотезу по ходу проекта. Исходить надо из того, что надсистема — это не просто что-то «данное в ощущениях», неизменяемое, «спущенное с небес». Нет, окружение целевой системы тоже сделано создателями, тоже

непрерывно меняется, и можно выдавать гипотезы о том, как оно там должно быть устроено. Современная инженерия основывается на том, что вы непрерывно корректируете ваше понимание мира, корректируете ваше знание о ситуации использования целевой системы, знание о возможностях команд, которые создают надсистему подкорректировать окружение целевой, знание о том, как лучше удовлетворить потребности путём модификаций целевой системы так, чтобы она улучшила работу надсистемы. Вам надо не только учесть разнообразные предпочтения внешних проектных ролей в отношении надсистемы как окружения целевой системы, но и учесть возможные изменения в этих предпочтениях.

- **целевая система** — описывается как «чёрный ящик» преимущественно постепенно детализирующимися сценариями использования. По большому счёту тут то же самое рассмотрение, что и для надсистемы, в ISO 15288:2023 это называется **рекурсивным применением системного мышления**.

А ещё описываются **два «прозрачных ящика», но один в концепции использования, а другой — в концепции системы**:

- **Надсистема** — описывается прежде всего концепцией использования целевой системы с акцентом на роль/функцию целевой системы в надсистеме. Показывается, каким образом наличие целевой системы в составе надсистемы приводит к удовлетворению интересов/предпочтений/потребностей/needs внешних проектных ролей. Это и есть важное для нас «устройство надсистемы», прозрачный ящик. Концепция использования — это в какой-то мере концепция надсистемы (рекурсивное применение системного мышления! На каждом уровне происходит одно и то же) с точностью, достаточной для описания целевой как чёрного ящика, выполняющего какие-то функции в надсистеме так, что становится понятным удовлетворение интересов внешних проектных ролей. Для настенных часов надсистема — интерьер и люди в интерьере, надо описать, что там окружает часы и как взаимодействуют (например, мешает ли людям громкое тиканье часов по ночам).
- **Целевая система** — описывается концепцией системы, в том числе архитектурой как дополнительными решениями по конструкции системы (с ожидаемым влиянием архитектурных решений главным образом на архитектурные характеристики/«предметы интереса», а не функциональность). Конечно, концепция системы и в том числе архитектура показывают, как удовлетворяются интересы не только внешних проектных ролей (их интересы больше описывает концепция использования), но и внутренних/командных проектных ролей, то есть интересы визионеров, разработчиков, архитекторов, инженеров производственной платформы целевой системы. Например, для часов с заданной точностью хода на их внешней границе надо принять решения по их внутреннему устройству: будут ли использованы в часах драгоценные камни для уменьшения трения и тем самым повышения точности хода, или в настенных часах это — роскошь, требуемая точность хода будет достигнута просто путём использования твёрдых сплавов для осей и подпятников. Или вообще настенные часы делать не механическими, а электронными, чтобы достичь требуемой точности хода.

На диаграмме целевая система и её подсистемы показаны красными кружками, отношение «часть-целое» показано вхождением кружков друг в друга. Окружение — синие кружки, ближайшая надсистема — синий большой кружок, в который входит и целевая система, и несколько других систем. Конечно, есть и ещё какие-то системы в окружении, которые не входят в состав надсистемы. Системных уровней много и внутри границы целевой системы, и вовне границы целевой системы.

Внешние проектные роли времени использования показаны синими человечками, но это не люди. Это роли, исполнять их могут и люди, и люди с компьютерами, и команды, и фирмы со всеми их людьми и оборудованием. В рассмотрении operations такие внешние роли, например, «пользователи», при этом мы рекомендуем общий тип «пользователь» нашей мета-мета-модели всегда заменять на тип роли из мета-модели (предметной области целевой системы) — в играх это игрок::пользователь, в бухгалтерских системах бухгалтер::пользователь и аудитор::пользователь. Но внешними проектными ролями могут быть и вредители/хакеры, и инспекторы внешних надзоров — кто угодно, кто взаимодействует с целевой системой в operations time. Эти роли показаны в кружках, чтобы подчеркнуть, что это тоже системы в окружении (роль — ролевой объект, который играется каким-то агентом как конструктивным объектом).

В рассмотрении времени создания (construction time) создатели::системы, мы просто не рисуем кружки и их вложенные/внутренние части.

На всей диаграмме отношения создания (между правой и левой частями диаграммы), и взаимодействия между системами одного уровня при их работе (между кружками одного уровня) не показываем. То, что делить на части системы можно очень по-разному (скажем, функционально и конструктивно: как ложку — «держало» и «хлебало» будут двумя функциональными частями ложки, но конструктивная часть цельнолитой или даже деревянной ложки будет одна изготавливаемая «деталь»-конструктив), тоже не показываем, хотя в концепции системы множественность согласованных между собой разных видов описаний в рамках одного системного описания будет принципиальной. Обычно

рассматриваются ещё и разные варианты согласованных декомпозиций (включая ещё и размещение, и ценовую декомпозицию — что там сами изготавливаем, что отдаём). И дальше изо всех согласованных между собой вариантов разных декомпозиций оставляем только «наименее плохой вариант из согласованных», ибо «лучший вариант» недостижим в эволюции, всегда удаётся достичь только локального оптимума.

Важно для этой иллюстративной диаграммы то, что «синенькие» внешние роли во время создания и развития занимаются устройством «синенького» (надсистемы) на диаграмме, а вот «красненькое» для них — «чёрный ящик». А «красненькие» внутренние проектные роли (роли команды проекта) вынуждены и выяснять прежде всего синенькое, но и дальше идти внутрь «красненького», то есть рассматривать сначала окружение «чёрного ящика» (ближайшую надсистему как «прозрачный ящик», но и надсистемы в дальнем окружении), а затем делать «чёрный ящик» прозрачным — формулировать гипотезы не только о функциональности целевой системы как чёрного ящика (концепция использования), но и о конструкции (концепция системы, в том числе начальные архитектурные решения).

Команда проекта создания и развития целевой системы создаёт её концепцию. При создании концепции чаще всего начинают с описания функционального разбиения, получив понимание, что там происходит внутри системы в терминах ролей подсистем (подсистем как ролевых/функциональных объектов), а также функций подсистем. Например, «чтобы часы показывали время, когда на них поглядят ночью, мы сделаем постоянно светящиеся стрелки, днём этого не будет видно, зато ночью всё будет хорошо» или «чтобы часы показывали время, когда на них поглядят ночью, мы предусмотрим, что при падении освещённости в окружении часов будет включаться подсветка» или «для удешевления часов мы не будем учитывать, что на них будут глядеть ночью, если очень надо, пусть поставят какую-нибудь подсветку комнаты, чтобы осветить циферблат». Эти описания будут концепцией использования подсистем (рекурсивное применение системного мышления!), при этом сами подсистемы ещё придётся проектировать, придумывать метод изготовления и материалы, из каких надо будет эти системы изготавливать (надо будет найти аффордансы, которые станут конструктивами системы). Скажем, в примере с «часами ночью» придётся придумывать, из чего делать подсветку циферблата — светодиоды, лампочки накаливания, люминофоры. Что-то подойдёт (аффорданс — это «подходяшка»), что-то не подойдёт для того, чтобы выполнить ожидаемую функцию подсистемы в составе целевой системы.

Системное мышление рекурсивно, оно применяется одинаково на каждом системном уровне, каждой командой-создателем систем. И каждая команда имеет свои «системные координаты» вокруг своей целевой системы и отдельно ещё и «их системы», поэтому самое важное — это договориться, что же это за системы! Договариваются, описывая прежде всего поведение системы по отношению к надсистеме (сначала вверх по уровням!), стараясь описывать систему как «чёрный ящик», чтобы потом можно было пробовать в качестве гипотез самые разные варианты устройства этой системы, то есть рассматривать самые разные варианты системы как «прозрачного ящика» (вниз по уровням — потом!).

Мы в разделе объясняем рекурсивность только на примере двух уровней (система-надсистема), но эта рекурсивность используется на всех уровнях — и для системы с подсистемами, подсистем с под-подсистемами и так далее до элементов систем, и

для надсистем и окружения надсистемы, тоже на много уровней. На всех уровнях вокруг любой системы можно (и даже нужно) развернуть полное системное рассмотрение примерно так же, как развёрнуто оно вокруг целевой системы.

Системное мышление говорит, что концепция использования (раньше — потребности и требования) описывает две разные системы (надсистему как «прозрачный ящик» и целевую систему как «чёрный ящик»). Но методы/способы того, как именно выявить интересы/потребности/предпочтения внешних проектных ролей в каких-то важных характеристиках надсистемы и сформулировать концепцию использования, как их не пропустить в разговорах с внешними проектными ролями, затем не потерять в коллективном внимании участников проекта создания и развития системы, как оттрассировать (trace, соотнести) поведение целевой системы к потребностям (там же эмерджентность, появляются новые свойства, например, надо обосновать, почему использование шестерёнок из материалов с небольшим тепловым расширением приводит к увеличению точности хода в механических часах), какие нужны ещё дополнительные методы, которые позволяют не просто управлять вниманием к системам, а ещё создать и развивать эти системы, причём в конечном итоге сделать их успешными — это говорится в руководстве по системной инженерии, это выходит за рамки руководства по системному мышлению. В руководстве по системному мышлению говорится только о том, что «если вы хотите описать систему как чёрный ящик, то это вы хотите разработать её концепцию использования».

Как именно разрабатывать концепцию использования, какой метод брать? Как выглядят результаты разработки разными методами/способами/стилями (есть много вариантов)? Это всё будет в руководстве по системной инженерии, а для систем разной природы (роботов, мастерства, организации и т.д.) будет ещё и уточняться в каких-то руководствах по прикладным инженериям и менеджменту.

Наше руководство пока касается только понятийного наведения внимания на системы и системные описания, оно пока не про то, как делать системы (методы/практики/культуры работы), а про то, как думать о системах, когда их делаешь. В нашем руководстве описываются типы мета-мета-модели объектов методов работы, например типы «система», «системное описание», «концепция системы». Руководство по системному мышлению наставляет о том, что именно следует системно моделировать в проекте, оно про мышление моделированием: какие объекты в жизни надо брать для описания в них важного (моделирования), что в них важного (какие модели нужны).

Но как именно моделировать, когда моделировать, кому (роли!) моделировать — это в руководстве по системной инженерии, а потом и в руководствах по прикладным инженериям (например, если у вас система — организация, то в руководстве по системному менеджменту, если у вас система — мастерство, то в руководстве по инженерии личности).

Из нашего руководства по системному мышлению вы должны лишь понимать, что **концепция использования привязана к границам целевой системы и надсистемы соответственно, они не должны быть упущены вашим вниманием, они обязательно должны присутствовать в проекте! И ещё в концепции использования должны быть помянуты архитектурные характеристики (о них позаботятся не**

разработчики::«внутренняя/командная роль», которые готовят концепцию использования, а архитекторы::«внутренняя командная роль»). Все эти модели::описания системы как чёрного ящика документируют. Конечно, без разработки концепции использования систему можно сделать (строители египетских пирамид их построили, но вряд ли там была разработана и документирована концепция использования), но успешной она может стать только случайно (например, не надо спрашивать, сколько лет строились пирамиды, сколько они при этом стоили, и сколько недовольных проектом погибло. В государственных проектах до сих пор крайне не любят прописывать концепции использования, это не случайно).

Исполнители ролей, которые замышляют (концепция использования, постепенно детализируемая всё время проекта), проектируют (концепция системы, постепенно детализируемая всё время проекта), изготавливают (воплощение системы, при этом тоже всё время проекта — речь не только о создании, но и о развитии системы), ликвидируют целевую систему (до этого дело доходит редко, но доходит) или отдельные какие-то части уже имеющейся и работающей целевой системы, чтобы получить новую фичу/возможность — это «**организация проекта»::создатель** (люди, AI, их оборудование и материалы труда). Этих исполнителей внутренних проектных ролей часто называют командой проекта, рабочей группой проекта, а иногда и просто проектом. Будьте внимательны к типам: тут не проект/design, а проект/project, и не проект::поведение, а ещё и проект::организация. Путаница между объектом и его поведением, обычная в быту для ролей и функций, в случае проектов традиционна. Проверяйте: ваш «проект» — это «команда проекта» или «поведение команды проекта»?

Организация определяется её подчинённостью: как она сотрудничает и работает совместно, но в ней важны проявляемые в окружение роли, которые организация может играть как «коллективный актёр» с его реквизитом. Организация (команда, коллектив из нескольких команд, предприятие из нескольких коллективов, кооперация из нескольких предприятий, рабочая группа проекта, собранная из самых разных людей с возможностью использования ими самого разного оборудования, и т.д.) — это **оргзвено, коллективный актёр**. Обычно для механической целевой системы создатель — организация. Но в графах создателей одна организация может создавать другую (часто такое называют «оргпроект», «проект организационного развития»).

А внешние проектные роли? Это те, кто напрямую не подчиняются в выполнении своих ролей исполнителям ролей в команде проекта. Есть и промежуточные ситуации: команда подрядчиков из другой фирмы, занятая нашей целевой системой, считается исполнителями внутренних ролей, поскольку формально она работает над нашей целевой системой (например, делает какую-то подсистему для этой целевой системы, или оказывает какой-то сервис, помогающий сделать целевую систему), а не над нашей надсистемой. Но она же считается внешней, ибо подчинение исполнителей ролей в рамках контракта будет только частичное. Так что роли в команде клиента — внешние, роли в команде проекта по созданию целевой системы — внутренние, а роли в команде подрядчика — серая зона, разные люди считают их разными (хороший тон считать их внутренними, но жизнь часто заставляет просчитывать ролевые интересы примерно так же тщательно, как для внешних ролей — «на Аллаха надейся, а верблюда привязывай»).

В предыдущем абзаце явно связывались отношения подчинения исполнителей ролей/агентов друг другу (кто чей будет начальник, а кто не начальник, а просто исполнитель роли клиента или инспектора, или ещё кого) с тем, какие роли будут считаться внешними, а какие внутренними. Да, **современное системное мышление начиная со второго поколения занимается не только целевыми системами, но и системами создания, в том числе и из людей и AI, в том числе коллективными создателями.** Тем самым системное мышление не только применимо в инженерных рассуждениях по отношению к «железным» или программным системам, оно применимо и в менеджерских (инженерии предприятия: стратегических/визионерских, организационных, операционных, административных) рассуждениях по отношению к коллективным системам из агентов (в том числе из людей, AI, компьютеров и оборудования — предприятиям, организованным рынкам/маркетплейсам, саморегулируемым организациям, и т.д.).

Да и к любым другим системам системное мышление тоже можно применять, хотя чем выше **эволюционный/системный/организационный уровень, тем сложней будет договориться даже по поводу концепции использования, а уж о концепции системы как способе реализации концепции использования и подавно. Это происходит из-за неминуемых конфликтов между системами разных уровней и неустроенности в том, какова могла бы быть концепция системы.**

Физическая теория эволюции говорит, что нельзя достичь истинного/глобального оптимума. У вас каждый раз гипотеза о том, что вы достигнете нового уровня квазиоптимума/«локального оптимума», получите наименее худшее инженерное решение, можете его пытаться незначительно улучшать, очень редко получится резко улучшить (придётся поработать на многих системных уровнях — большой эволюционный переход/major evolutionary transition), но всегда будет куда стремиться, это всегда будет квазиоптимум. Развитие жизни бесконечно, развитие технических систем тоже бесконечно, это часть техно-эволюции — всегда есть что улучшить, но любое улучшение не окончательно.

Системное мышление фундаментально/трансдисциплинарно: оно позволяет в том числе договорить продуктивно конфликтующие (разные интересы!) роли инженеров целевой системы (визионеры, разработчики, архитекторы, инженеры производственной платформы/внутренней платформы разработки), инженеров системы создания/менеджеров (бизнесмены, организаторы, орг-архитекторы, администраторы), добавить участниками этих договорённостей самые разные роли из других затрагиваемых проектом сфер деятельности, договорить их всех друг с другом в рамках коллективного мышления, коллективной деятельности по созданию систем.

Нет «прикладных проектов» какой-то направленности, для которых системное мышление было бы особо пригодно, а для других — не особо пригодно. Системное мышление универсально пригодно: используется и в проектах::«поведение команд для получения результата», и для того, чтобы сформулировать цели проекта, прежде всего — сформулировать целевую систему проекта. Например, системное мышление отлично работает, когда надо принимать решения при основании предприятия, когда ещё непонятно, какую конкретно целевую систему будет делать команда (какова рыночная гипотеза), поэтому и команды обычно тоже ещё нет — и ставится задача инженерии предприятия с нуля («business engineering», включая предложение бизнес-модели), которая уже и займётся проектированием

«железной» или «программной» технической системы, развивая начальную приблизительную догадку о том, какова это могла бы быть система.

Примеры системной терминологии

Рассказ о целевой системе всегда начинается с её описания как чёрного ящика, при этом по факту приходится рассказывать не столько о самой целевой системе, сколько о надсистеме. И это не случайно: системный подход начинается с того, что система определяется по её роли в окружении (ролевой/функциональный объект среди систем окружения), по её ролевому поведению/функции (изменениям состояний систем в окружении, которые проводит описываемая система как «чёрный ящик»).

От границы целевой системы по системным уровням наверх, к окружению, к надсистеме — это первый мыслительный ход в системном мышлении, это главный императив. Никакого «деления на части», никакого «состоит из», пока не выяснили, для чего используется целое, в котором мы интересуемся частями — сначала само это целое «является частью в» и «входит в» и понимание, что там делает это целое в своём «надцелом» !

Дадим пример использования терминологии системного мышления в описании простой механической системы с электрическими элементами — напорного насоса.

Целевая система — напорный насос, использованный в насосном агрегате (надсистема), который в свою очередь использован в насосной станции (т.е. наднадсистема — насосная станция). Его роль — напорный (чтобы вода доходила до верхних этажей здания) насос, действие/функция — поднимать давление жидкости. Одна из его подсистем — роторная сборка (ротор с лопатками, это центробежный насос/centrifugal pump).

Одна из внешних проектных ролей — собственник-оператор (owner-operator) насосной станции. Потребность — бесперебойный напор воды, выходящей из насосной станции. Потребность говорит не про насос как целевую систему, а про насосную станцию как наднадсистему. Потребности внешних проектных ролей чаще всего описывают надсистемы каких-то уровней. Хотя они могут описывать и системы создания, например «потребность — чтобы вы изготовили насос за три дня», но и тут чаще всего можно легко проследить как «нужное для надсистемы»: «иначе наша насосная станция не сможет приступить к работе вовремя».

Потребность бесперебойного напора воды, выходящей из насосной станции,

отражается в архитектурной характеристике насоса «время наработки на отказ» и даёт повод задуматься об аварийном источнике энергии для мотора насоса — но помним, что аварийный источник энергии, как и сам мотор насоса будет тоже в надсистеме для этого насоса. Насосный агрегат состоит из насоса и мотора, на картинке показан насосный агрегат, насос в нём показан зелёненькой частью. Остальное — окружение насоса, входящее в состав надсистемы. Вот подробней:

- Насосная станция (наднадсистема)
- Насосный агрегат (надсистема)
- Насос (система)
- Роторная сборка как ротор с лопатками (подсистема)
- Ротор (подподсистема)

Концепция использования: напорный насос выдаёт в присоединённую к нему трубу напор до 40 метров, расход (проток воды) до 10000 литров/час. Архитектурная характеристика: наработка на отказ от 5000 часов. Никаких предпочтений в значениях важных характеристик («расход/проток воды как можно больше», «максимальная наработка на отказ»), ибо это уже договорённые конечные значения, предпочтения самых разных ролей в них уже учтены. Концепция использования и архитектурные характеристики описывают целевую систему (относясь к разным временам: время эксплуатации «в моменте», общее время эксплуатации), предпочтения описывают «хотелки» отдельных ролей. Они про разные объекты: концепция использования отражает успешность системы (описание «чёрного ящика», выполненное как отражение договорённостей ролей по поводу их предпочтений), а предпочтения отражают роли. Роли — это системы создания, предпочтения характеризуют роли, а не целевую систему. Роли будут продвигать гипотезы по поводу успешности системы в выгодную им сторону. Скажем, инженер-разработчик требует для насоса больший расход воды «для заведомой производительности с запасом», но совокупная стоимость владения будет тогда много больше — и инженер-разработчик с инженером-визионером (отвечающим за то, можно ли это будет продать, или весь проект становится невыгодным) договорились, что до 10000 литров/час будет достаточно, «наименее плохое решение».

Некоторые системы в ближнем окружении: мотор, рама (они входят в состав насосного агрегата как надсистемы, но они внешние по отношению к насосу), входящие в состав насосной станции, но не насосного агрегата электрическая проводка, трубы.

Некоторые создатели: конструкторское бюро (проектировавшее насос), завод (изготовитель насоса), проектировщик и строитель насосной станции (они принимали участие в создании: выбирали именно этот насос, закупали его, проводили монтаж на насосной станции), гаечный ключ (им закручивали гайки, которыми насос крепится к фундаменту). Неясно, насосный агрегат создавался строителями насосной станции, или поставка насоса шла сразу в составе насосного агрегата (то есть конструкторское бюро и завод выдавали насосный агрегат), это надо выяснять дополнительно.

Другой пример: электроника с островками софта — наручные смарт-часы.

Чтобы определить надсистему, в состав которой (в момент эксплуатации) входят смарт-часы, придётся рассмотреть западную христианскую традицию отношения людей к собственности. Личности-как-бессмертная-душа считаются владеющими собственным телом, которое при этом рассматривается не столько как сам человек, сколько как просто носитель::хард его личности::софт, принадлежащая личности вещь. Личность::софт владеет своим телом::хард/аппаратура, это его собственность, он принадлежит сам себе, это выражение «свободы воли». Человек в христианской традиции = личность-как-душа плюс тело. Это свойство людей называют обычно **самопринадлежностью**. Собственные вещи человека считаются просто продолжением его биологического тела. Никто не может взять тело человека или повредить его, точно так же никто не может взять или повредить его личную рубашку, личные часы — они считаются входящими в состав человека, буквально (отношение is_part_of/composition). Это и есть «священная частная собственность». Тело = биологическое тело плюс другие личные вещи и даже плюс средства производства (например, машиностроительный завод) в собственности личности.

Тем самым мы с некоторой условностью можем считать надсистемой для смарт-часов самого человека-владельца этих часов: часы будем считать буквально входящими в состав его тела. Для многих личных предметов и инструментов это подтверждается и нейрофизиологическими экспериментами: личности относятся к ним буквально как к продолжению их тела[300], «экзотелу».

Ещё одна интересная система этого класса, включающая людей — это **домашнее хозяйство** (household), которое может включать и самих владельцев, и дом, и домашнюю утварь.

В подобного сорта системах входящие в них люди-агенты одновременно и играют как актёры проектные роли, и они же представляют из себя «просто тело/организм», с которым мы работаем ровно как и с другими материалами, т.е. просто учитывая его физические свойства — размеры, влажность, прочность и т.п. Человек как агент физичен/биологичен/материален, так что он состоит из мышц, костей и других тканей. В системном мышлении учитывают и разумность человека-агента в части мастерства игры им какой-то трудовой роли, и его материальность как тела.

В примере со смарт-часами надсистема — человек-владелец этих часов, но не как личность::софт, а больше как принадлежащее личности тело/организм со всем на

[300] Brandon Keim, «Your computer really is a part of you», https://www.wired.com/2010/03/heidegger-tools. Мозг управляет компьютерной мышкой так, как будто это и впрямь продолжение руки.

него надетым: рубашкой, туфлями, часами. Биологическое тело человека будет находиться в системном окружении часов.

Потребности/интересы личности владельца как внешней роли (владельца тела-с-часами): он должен быть информирован о точном времени, но этот список его интересов/потребностей по большому счёту остаётся открытым. Роль часов весьма сложна в формулировании, ибо речь идёт о гаджете с разнообразным поведением, то есть многофункциональном гаджете.

Концепция использования будет относиться не столько к самим «часам» как «измерителям хода времени», сколько к самым разным предполагаемым вариантам их эксплуатации — это могут быть как собственно часы, так и радио, плеер, измеритель пульса, также от смарт-часов ожидается, что они не натирают руку (кожа руки как часть тела-надсистемы находится в окружении смарт-часов!), ожидается, что они будут успешны, если будут модной на момент продажи расцветки, работать без подзарядки не менее 20 часов, вес будет не более 80 грамм, с ними будет работать магазин приложений, у них будет связь с внешним компьютером (смартфоном, планшетом, десктопом).

Системы в окружении — рука, одежда (как минимум, рукав рубашки или пиджака), зарядное устройство. Подсистема — защитное стекло (например, из Gorilla glass — закупается отдельно у контрактора, занимающегося производством защитных стёкол на заказ).

Создатели смарт-часов: конструкторское бюро, завод в Китае, магазин по их продаже в Европе. И вот если взять магазин как одного из создателей в роли «продавец», то про смарт-часы можно узнать много нового. Ожидаемое поведение магазина («магазина с часами внутри», что хочет владелец магазина от имеющихся в магазине во временной собственности часов, это stakeholder needs, потребности для целевых часов) — это продажи на какую-то немаленькую сумму, что легко переводится в гипотезу удобной упаковки часов для складской обработки (гипотеза! Может быть, это и не нужно, или удобство определено будет неправильно), красочной упаковки и буклетов для выкладывания в торговом зале, а также хорошей рекламы (то есть услуга рекламы рассматривается магазином как часть продукта — без какого-то уровня рекламы хороший магазин может просто не взять товар в продажу). И **часы-как-товар/упаковка-с-часами** получают тем самым дополнительные сценарии использования часов-как-товара магазином с целью вытаскивания из посетителей магазина денег в обмен на часы. По факту мы здесь вводим новую систему «часы-как-товар», упаковку с часами, отличную от часов и новую роль «посетитель магазина» (который может стать покупателем, а может и не стать им, а покупатель может стать потом пользователем часов, но может и не стать им). Момент использования часов::товар, то есть операция использования типа «товар» — продажа, то есть «поставка против платежа», а момент использования часов — смотрим на них время, ставим будильник, принимаем звонки. Не путаем использование товара магазином и использование часов их пользователем! Эту ошибку часто делают «предприниматели из акселератора»: они настолько увлечены идеей «продать», что свой «продукт» считают использованным не как продукт, а как товар! Что будет после покупки — их не волнует, часы для них могут быть и принципиально не ходящими, хоть макетом, лишь бы заплатили. Это ошибка: часы::продукт/система и часы::товар — это по факту разные системы, у них разные пользователи, речь идёт о разном времени (время создания, время продажи, время использования часов), о разных

ролях создателей, разных ролях команд проекта.

Это очень важно — выявлять самые разные внешние проектные роли, выявлять их интересы, определять потребности (согласовывать приемлемые и желаемые внешними проектными ролями свойства надсистемы, иногда даже предлагать эти свойства — исполнители внешних проектных ролей часто и сами не знают, чего бы они хотели, или не считают то, что они хотят возможным, разработчики целевой системы могут что-то предложить им), а потом выводить из этих потребностей полную концепцию использования, описывающую поведение целевой системы в ходе использования/эксплуатации/operations/функционирования. Иногда приходится даже вводить новую систему (в нашем случае часы-как-товар) и не путать её с целевой системой (у них разные времена использования, разные функции по отношению к окружению, поэтому важные характеристики и устройство — разные).

Концепция использования с многочисленными сценариями использования самых разных возможностей/фич целевой (или любой другой) системы опишет систему, поможет с определением её границ: что будет система делать, а чего система делать не будет. Скажем, вот эти смарт-часы будут показывать время, а вот эти смарт-часы будут показывать время и артериальное давление (а дальше уже можно в концепции системы думать, как это всё устроить так, чтобы была приемлемая для продажи цена). В случае часов-как-товара понятно, что в состав целевой системы входят и упаковка, и рекламные буклеты, и даже транслируемая по каким-то каналам реклама. А вот в состав часов можно включать и какой-то дополнительный софт приложений для этих часов. Можно считать, что магазин приложений для этих часов будет системой создания каких-то фич/возможностей часов, но вот с упаковкой так считать уже не получится.

Этот пример показывает, что для разных ролей удобно по-разному описывать (уже имеющуюся или проектируемую) целевую систему, а иногда даже вводить в рассмотрение дополнительные системы. Для магазина эта самая упаковка-с-часами-и-реклама/часы-как-товар будет целевой: её нужно замыслить (товаровед!), изготовить (закупка! Логистика!), использовать (продать!). А сами часы? А они не нужны. Нужна упаковка-с-часами/часы-как-товар!

Современная инженерия часто имеет дело с **киберфизическими системами** (cyber-physical systems), которые имеют в своём составе датчики (sensors), воздействующие на внешний физический мир исполнительные устройства (actuators: чаще всего разные моторы, но это могут быть и осветительные приборы, электрические выключатели) и компьютер (cyber-, кибер-, управляющую часть), управляющий работой всей системы. Примером такой системы может быть дрон для аэрофотосъемки.

Надсистема — стройка (недостроенное здание в ходе его строительства плюс развёрнутый вокруг здания набор систем создания, «мероприятие создания здания»). Одна из внешних ролей в проекте — это заказчик-застройщик, потребность которого — подконтрольность хода строительства. Дрон-фотограф (определяется по его роли) — целевая система. Функция::поведение дрона — выдача потока фотографий высокого разрешения стройки с интересных заказчику-застройщику ракурсов. Фрагменты концепции использования: длительность полёта и фотографирования не менее 1 часа, изображение стройки получается с оптическим разрешением не менее 20 Мпикселей, зарядка между полётами происходит не более 1 часа, к зарядке дрон подлетает сам. Подсистема — фотокамера, подподсистемы дрона (подсистемы фотокамеры как подсистемы дрона) — объектив и матрица фотокамеры. Системы в рабочем/операционном/эксплуатационном окружении: зарядка, стройка с её зданиями, сооружениями и оборудованием (краны), в воздухе они могут рассматриваться как препятствия (например, трос от крана).

Создатели дрона-фотографа — конструкторское бюро, завод-изготовитель, магазин, ремонтная мастерская, эксплуатационная служба с оператором дрона-фотографа (эта служба как бы дорабатывает дрон перед каждым вылетом и после вылетов: проверяет, заправляет, настраивает, чистит, ремонтирует. Поэтому мы считаем, что эта служба не в системах окружения, а в системах создания — работает не в момент использования/«принесения пользы»/работы/функционирования дрона).

Ровно такой же рассказ будет и о менее классических системах, нежели традиционные для инженерии электромеханические (насос) и киберфизические (дрон, смарт-часы) системы.

Например, мастерство. Инженерии мастерства посвящено отдельное руководство по инженерии личности (личность — это совокупность мастерства агента). По факту это руководство, описывающее «мастерство одного создателя по созданию мастерства другого создателя». В реальных проектах такие графы создателей с длинными цепочками обучения (создания мастерства: один создатель создаёт мастерство создания мастерства у другого создателя, а тот продолжает эту цепочку, пока последний создатель в цепочке не делает нужное действие) довольно типичны.

по «мастерству создания мастерства», для системного мышления такие графы создателей типичны.

Возьмём, например, **мастерство системного мышления** — это такая целевая система, которая в момент её работы выполняет мышление::вычисление с

понятиями системного подхода, причём реализована эта система какими-то конструктивными частями мозга или компьютера с программой AI, играющими роль этого мастерства (но мы даже не знаем точно, какими именно частями мозга или частями компьютера). Подсистемы мастерства системного мышления: мастерство мышления о системных уровнях, мастерство мышления о графах создателей. Надсистема мастерства системного мышления: интеллект, как общее мастерство мышления по всему интеллект-стеку фундаментальных методов мышления (опять же, мы понимаем, что оно тоже реализовано какими-то частями мозга и/или компьютера агента, но мы не очень понимаем, какими — что не мешает нам об этом мастерстве рассуждать, как о материальном объекте.

Более того, это может быть в случае командного системного мышления и несколько мозгов и компьютеров плюс средства коммуникации между ними). Системы в ближнем окружении мастерства системного мышления: мастерство онтологического мышления, мастерство собранности. Создатель мастерства — онлайн-версия руководства по системному мышлению как программа на серверах дата-центра и опционально наставники стажировки — они могут быть, но их может и не быть. Создатели создателя мастерства, то есть создатели онлайн-версии руководства по системному мышлению — автор текста руководства и заданий (Анатолий Левенчук), сотрудники и добровольцы мастерской инженеров-менеджеров с их LXP (learning experience platform) Aisystant.

Моделирование: краткое описание системы

Опишите от трех выбранных вами из вашего рабочего проекта систем по примерам из предыдущего раздела. Не забываем глаголы (функции системы обычно выражаются отглагольными существительными, но эти существительные означают поведение, а не как-то ведущую себя роль).

Номер п/п	Система	Основная функция системы	Некоторые внешние проектные роли	Примеры потребностей (что хотят от надсистемы/окружения тамошние проектные роли, для проекта нашей системы — внешние роли)	Примеры сценариев использования (что делает система с её окружением — развёрнутое описание функции)	Некоторые системы в ближнем окружении	Примеры систем создания

Разделение труда и системные уровни

Человеческие роли, а сейчас и роли AI, а также роли коллективных агентов (организаций) в проектах по созданию и развитию самых разных систем специализируются на различных системных уровнях: рекурсивно на каждом системном уровне применяется мышление инженера, который собирает

приемлемую для надсистемы (уровень выше) целевую систему (текущий уровень), используя эмерджентность от взаимодействия собранных вместе подсистем (уровень ниже). Каждый агент, играющий роль такого инженера, специализируется на эмерджентности какого-то одного уровня, ибо его мастерство не может быть безгранично: два уровня закрыть мастерством одинаково высокого уровня много труднее, три ещё труднее. А ещё для разных систем (скажем, выращивание рапса и выплавка стали) мастерство в методах создания и развития этих систем содержательно очень разное, быстро его не познаешь и не освоишь до уровня беглого применения. Как решение этой проблемы появилось **разделение труда:** разные агенты специализируются на разных методах/способах работы, привязанных к каким-то довольно узким видам систем, привязанных к каким-то довольно узко выделяемым системным уровням. Труд работает с какими-то предметными областями, а предметы оказываются плотно привязаны к системным уровням, ибо при смене системного уровня меняются свойства предметов и тем самым меняется метод работы для изменения систем так, чтобы получить или убрать свойства этого системного уровня. Чем больше знаний о какой-то предметной области, тем труднее получить мастерство в этой предметной области у одного агента: одна предметная область тем самым определяется как более мелкая, выделяются какие-то виды систем, в них какие-то дополнительные системные уровни, в них какие-то отдельные эмерджентные свойства — и отдельные агенты начинают специализироваться на мастерстве достижения этих эмерджентных свойств. Это называется углублением разделения труда. Например, при углублении разделения труда вместо одной роли «инженер» вдруг возникает набор самых разных ролей — визионер, разработчик, архитектор, инженер внутренней платформы разработки. У каждой из этих ролей свои предпочтения в предметах интереса, поэтому агентам, играющим эти роли, надо будет договариваться, а если в очень маленьком проекте кто-то будет выполнять все эти роли вместе, то можно будет сомневаться в том, что у него достаточно мастерства для хорошего выполнения каждой из этих ролей, а также можно будет подозревать конфликт интересов (много ролей с противоречивыми интересами по одному предмету интересов у одного агента-исполнителя всех ролей. Например, в руководстве по системной инженерии будет подробней рассказано, какие противоречия между ролями визионера и разработчика, разработчика и архитектора, они находятся в продуктивном конфликте. И у одного агента-инженера, выполняющего все эти роли, будет неминуемо конфликт интересов).

Разделение труда вездесуще. В группе методов телесной работы медики лечат ткани тела, тренеры создают виды телесного/двигательного поведения (виды методов движения) и их стили (варианты методов) и тренируют адептов на уровне работы организма в целом, амбассадоры культуры заботятся о культурном разнообразии движений на уровне сообщества (множества организмов, которые выполняют какие-то движения), а «телесная терапия» — это вообще про попытки решить психические проблемы через приведение тела в какие-то состояния (обычно — расслабления и спонтанного движения). И это мы явно помянули не все системные уровни, в нашем примере социальных танцев их явно было больше!

То, что мы говорим о разных ролях, и разные агенты будут выполнять эти разные роли по отношению к движущимся телам людей — это и есть разделение труда. Такое разделение труда продолжается и на более низких системных уровнях, и на более высоких. Например, генетики позволяют выявить людей с особым клеточным метаболизмом, что даёт возможность заранее отобрать для участия в

соревнованиях людей, устойчивых к нехватке кислорода — и они не будут ограничены там, где другим спортсменам будет трудно из-за недостатка кислорода в клетках мышц во время нагрузки. Спорт высоких достижений всё больше и больше превращается сейчас в соревнование молекулярных биологов и генетиков, а не тренеров, но уже начинается и соревнование фармакологов — поднялось движение за спорт с разрешённым допингом, enhanced games[301], техническое решение проблем ограничений человеческого организма. Для создания спортсмена нужны будут и знания молекулярного биолога и генетика, и знания фармаколога, но и знания тренера.

Изменение свойств систем нижнего уровня приводит к изменениям свойств систем верхних уровней — как раз через эмерджентность. И каждый род эмерджентности требует мастерства достижения этих свойств: понимание, какие аффордансы могут быть применены в качестве конструктивов для достижения желаемого поведения, выполнения желаемой функции в надсистеме.

Системное мышление оказывается глубоко связанным с разделением труда: на каждом системном уровне есть специалисты, которые искусны в достижении системного эффекта этого уровня — они получают его, зная, как собрать систему (целевую или «нашу», речь идёт о самых разных системах) из её подсистем. Но мало этого, они ещё знают, где можно использовать собранную ими систему, чтобы такие же специалисты на уровне выше смогли использовать их систему как подсистему. **Каждый агент как квалифицированный мастер-создатель разбирается в своём системном эффекте и связывает своей работой три уровня системного разбиения — а все вместе специалисты могут создать и развивать очень и очень сложную целевую систему, а также все необходимые для её создания и развития системы-создатели из их графа.**

Это верно для инженерии, но это верно и для остальных сфер деятельности, которые по факту тоже «инженерии», хотя и не совсем привычные. Тем не менее, мы рекомендуем думать о них «инженерно», обычно это существенно упрощает понимание происходящего в проектах создания нетехнических систем.

Например, для сферы телесных практик (методы движения в спорте, йоге, восточных единоборствах, танцах, сценическом движении и т.д.) инженерный подход существенно ускоряет понимание, избавляя от многочисленных метафор и художественных отвлечений от сути дела, а также обеспечивая нормативное/культурное/инженерное разделение труда. Как сотни людей сегодня делают кинофильм «инженерным способом», так и множество людей (и даже более обще — агентов, ибо не только людей) могут помочь из кого-то сделать полноценного участника танцевальной субкультуры (в просторечье — сделать из кого-то хорошего танцора, но мы помним, что танцор — это ещё не артист, а артист — это ещё не участник сообщества, практикующего какую-то субкультуру). Танцевальный перформанс (изредка используется сленговая форма «перфоманс») при этом оказывается многоуровневым, его сложность можно преодолеть, если использовать по отношению к нему системное мышление.

Труд — это синоним для метода работы по созданию и развитию системы. Разделённый труд — это какой-то широкий метод, разделённый на более узкие методы/практики/«мелкие виды труда», при этом разные агенты проекта выполняют работы по разным узким методам одного широкого метода. Скажем,

[301] https://en.wikipedia.org/wiki/Enhanced_Games

инженерный труд (широкий метод работы) оказывается разделён по разным агентам, которые выполняют более узкие методы, вместе составляющие инженерный труд — визионерство, разработку, принятие архитектурных решений, инженерию внутренней платформы разработки (подробней об инженерии в целом как широком методе работы и более узких методах инженерии будет рассказано в руководстве по системной инженерии).

Разделение работ — это когда по одинаково умелым агентам, работающим по какому-то методу (или даже подметоду) работы распределяются работы. Скажем, нужно выполнить работы по окраске забора, 100 квадратных метров. Метод выбран — окраска валиком с краской, выполняющая метод роль — «маляр». Разделение труда — это когда роль «маляр» выделяется из общей роли «строитель» и поручается не любому строителю, а умеющему красить. Разделение работ — это когда берётся десять агентов, умеющих красить (с мастерством окраски валиком) и каждый агент красит не 100 квадратных метров, а только 10. Разделяется только работа, а метод этой работы остаётся тем же самым, каждый агент выполняет весь метод/практику/деятельность, а не его часть.

Разделение труда — эволюционный процесс. Разделённый по агентам труд обычно эффективней, чем труд одного агента, и можно просто наблюдать за тем, как работает техно-эволюция, как в культуре происходит углубление разделения труда — разделение труда оптимизирует коллективных создателей, лучшие решения по разделению труда запоминаются и потом реплицируются по агентам. Но при этом встаёт проблема: как в одном проекте объединить разделённый труд в общий труд создания и развития целевой системы так, чтобы система была успешной?

Системное мышление позволяет объединять разделённый труд: даёт понимание того, почему происходит разделение труда, как объяснить профессионализацию агентов в том или ином «виде труда»/«методе работы». Так, прежде всего разделение труда происходит для труда/методов работы, связанного с объектами разных системных уровней. Если в самолёте мы выделим «двигатель как чёрный ящик», то решениями прозрачного ящика для него будут заниматься, возможно, уже другие люди, на более низком системном уровне, «двигателисты» — на каждом системном уровне появляются какие-то предметные особенности, новые свойства, часто это требует смены специализации инженерных ролей.

Если это произойдёт на многих уровнях со всеми самыми разными системами каждого из этих уровней, то мы увеличим шансы успешности системы: получим не идеальную (это невозможно! Техно-эволюция позволяет каждый раз достигать только какого-то квазиоптимума, «локального оптимума»), но в каком-то приближении к идеалу. Обычно нам уже достаточно, чтобы система была не идеальной, но чуть оптимальней, чем системы конкурентов («чтобы убежать от льва, не надо бежать быстрей льва, достаточно бежать быстрей, чем самый медленный человек, убегающий от льва»).

Что важно, так это что после создания и начала эксплуатации первоначальной системы всё то же самое продолжается: непрерывная разработка новых и новых возможностей системы, новых вариантов технических решений, реализация новых фич, непрерывное введение их в эксплуатацию, «непрерывное всё/continuous everything». Раньше однократно «удовлетворяли требования», теперь «создают и далее непрерывно развивают систему», часто даже силами не одной команды, а

самых разных команд, специализирующихся на создании и развитии самых разных видов систем. Разделение труда при этом помогает справиться со сложностью: не надо все знания о системе и способах работы по её созданию упаковывать в мастерство одного агента (хоть индивидуального, хоть коллективного).

Выступления современных танцоров или фигуристов неизмеримо сложней и интересней, чем они были сто лет назад, а учиться танцам или фигурному катанию до сравнимого уровня сегодня можно много быстрее, ибо за сто лет неизмеримо улучшились методы создания и развития мастерства. Космические корабли (те так и вовсе не существовали сто лет назад), вычислительная техника, программные системы, медицинские роботы — создание всех их требует многих разных видов мастерства у агентов, но при разделении труда можно рассчитывать на то, что этому мастерству можно научиться за небольшой срок (при наличии достаточного интеллекта).

Разделение труда, связанное с получением компетентности в работе на трёх смежных системных уровнях (разбираться как и что делать на текущем уровне целевой системы, знать, для чего делать и где и как использовать — разбираться в надсистемах, знать как устроено то, из чего делать — разбираться в подсистемах) даёт возможность выстраивать длинные цепочки деятельности самых разных людей, разбивая общую сверхсложную деятельность на всех системных уровнях на множество не таких сложных на каждом системном уровне.

Но нужно отметить, что на более высоких эволюционных/системных/организационных уровнях сложности, чем организации/коллективы интеллектуальных агентов, с системными описаниями и инженерным отношением у людей проблемы и с разделением труда, и с описанием лучших методов социальной инженерии (в сравнении с «отдачей на самотёк», то есть полаганием на эволюцию). Фронтир системного мышления проходит по линии мастерства создания коллективных систем уровня сообщества и общества. Заниматься системным мышлением в таких проектах нужно с осторожностью, разобравшись с его применением на более простых системах. Уже уровень личности крайне сложен, организация из многих личностей ещё сложней, но инженерия с этим справляется. А вот говорить что-то о разделении труда (кроме того, что «действуют рыночные силы») на уровне сообщества, общества, человечества — очень сложно. Знаний о создании сообществ, обществ, человечества/«агентечества» в достаточных количествах, чтобы его разложить по разным агентам и специализировать их по разным видам мастерства — таких знаний пока нет. Хотя у политиков и есть «штабы», в которых можно наблюдать какое-то разделение труда. Но аккуратней с проектами по созданию сообществ, обществ и человечества («агентечества»: включаем в агентечество и уровни общества, и уровни сообщества, и как их составляющие AI-агентов вдобавок к людям, отсюда и «агентечество», а не «человечество»).

Вопросы. Концепция использования и концепция системы

Люсе поручили разработать новую игрушку. Она беседовала с разными людьми и записала, что они ей говорили. Что в этих записях относится к концепции использования?

+++ ожидается, что ребёнок будет улыбаться не менее 20% времени при игре с

этой игрушкой ### Да, это из концепции использования — речь идёт о части надсистемы, т.е. о говорится о ребёнке, находящемся в окружении куклы, целевая кукла своим поведением приводит его в состояние улыбки.

--- микропроцессор куклы включает встроенную поддержку нейронной сети для синтеза речи ### Нет, это часть концепции системы, игрушка рассматривается как «прозрачный ящик», оговариваются части игрушки.

+++ ребёнок таскает игрушку по квартире в ходе игры, не сидит на одном месте ### Да, это из концепции использования. На этот интерес нужно будет отвечать в концепции системы (например, в игрушке надо предусмотреть аккумулятор, а не питание от сети).

+++ игра с игрушкой по часу в день за три месяца даёт рост IQ ребёнка на 10 единиц ### Да, это из концепции использования — речь идёт о части в окружении целевой системы/куклы.

+++ игрушка не выделяет токсины в ходе игры ### Да, это концепция использования — речь идёт о поведении самой целевой системы (игрушки) на её границе с окружением.

Ивану Ивановичу поручили разработать новое покрытие для ротора вентилятора. Он беседовал с разными людьми и записал всё, что они ему говорили. Что в этих записях относится к концепции использования?

--- покрытие будет трёхслойным, этого должно хватить ### Нет, это часть концепции системы: речь идёт о «прозрачном ящике», внутренней структуре покрытия/целевой системы.

--- покрытие должно быть двухслойным ### Нет, это часть концепции системы, и очень смущает слово «должно» (кому должно? Почему должно? Как обосновано? Это остатки языка «требований», прошлое поколение системной инженерии. Сейчас избегают деонтических операторов, говорят о рабочих, предпринимательских, конструкторских и т.д. гипотезах): речь идёт о «прозрачном ящике», внутренней структуре покрытия/целевой системы.

+++ клиент ожидает, что вентилятор с ротором с покрытием работает втрое дольше до отказа вентилятора, чем без покрытия на роторе ### Да это высказывание про надсистему (вентилятор), в состав которой входит ротор, в состав которого входит покрытие как целевая система. Это из концепции использования.

+++ клиент ожидает, что ротор с покрытием проработает минимум втрое дольше до замены ротора, чем без покрытия ### Да, это про надсистему (ротор вентилятора), в состав которой входит целевая система (покрытие) — это про концепцию использования покрытия как целевой системы.

+++ расходы на антикоррозийные мероприятия для вентилятора с ротором с покрытием будут меньше на 10% по сравнению с обычным ротором ### Да, это про надсистему (вентилятор), в состав которой входит ротор с покрытием, в состав которого входит целевая система (покрытие), то есть это высказывание из концепции использования.

--- покрытие не будет включать в себя редкоземельные элементы ### Нет, это явно про концепцию покрытия (речь идёт о подсистемах покрытия, его внутреннем устройстве).

--- расчёты в обоснование работоспособности покрытия планируем выполнить с использованием языка акаузального моделирования Modelica ### Нет, это относится к концепции системы создания (вообще другой системы! Не вентилятора, не ротора, не покрытия! Не того, что есть вокруг покрытия в момент использования, а не в момент создания) т.е. функцию программы расчётов в системе создания предполагается реализовать с использованием конструктива «моделер на базе языка Modelica».

--- расчёты в обоснование работоспособности покрытия планируем выполнить на сертифицированном программном обеспечении ### Нет, это относится к концепции системы создания (вообще другой системы! Не вентилятора, не ротора, не покрытия!) т.е. функцию расчётов в системе создания предполагается реализовать с использованием сертифицированного программного обеспечения. Расчёты происходят не во время использования покрытия!

--- адгезивность покрытия сделаем нетребовательной к отсутствию плёнки окислов на роторе в момент нанесения покрытия ### Нет, это концепция системы: свойства частей покрытия на момент его изготовления.

--- сделаем нанопокрытие ### Нет, это концепция системы: указание на то, как должно быть устроено покрытие.

Из архива принесли всё, что нашли про проект тетрахопореза: письма с потребностями самых разных людей, папку с окончательными требованиями (это было ещё когда было принято разрабатывать требования, сейчас это уже в прошлом), флешку с 3D-моделью и программой для станка с ЧПУ для изготовления деталей, копию акта испытаний, фотографию готовой системы. Что из этого относится к описаниям «прозрачного ящика»?

--- письма самых разных организаций с описанием интереса к использованию тетрахопореза в их ситуациях ### Нет, устройство тетрахопореза в этих письмах не обсуждается, это не описание «прозрачного ящика».

--- концепция использования ### Нет, это описание тетрахопореза как «чёрного ящика».

+++ 3D-модель в САПР ### Да, 3D-модель в САПР — это описание внутреннего устройства тетрахопореза, описание «прозрачного ящика».

+++ программа для станка с ЧПУ ### Да, это описание «прозрачного ящика», в программе для станка с ЧПУ отражено внутреннее устройство тетрахопореза, форма его деталей (хотя тут можно вести философские споры: описывается поведение станка с ЧПУ или описывается форма деталей тетрахопореза «в конструктивной манере, через операции создания»).

--- акт приёмочных испытаний (acceptance test) ### Нет, это про поведение экземпляра тетрахопореза — и ничего про устройство тетрахопореза.

Йон Тихий хочет узнать функциональность сепулькария. На поиск какого из описаний системы ему стоит направить максимум усилий?

+++ концепция использования сепулькария ### Да, именно концепция использования описывает функциональность целевой системы: то, для чего она предназначена, что она делает.

--- архитектура сепулькария ### Нет, из архитектурных решений можно узнать,

как устроен сепулькарий (например, какие ограничения есть на связи между его конструктивными частями), но очень трудно или даже невозможно восстановить его функциональность.

--- принципиальная схема сепулькария ### Нет, по принципиальной схеме можно понять, как работает сепулькарий (это может быть частью концепции системы), но очень трудно или даже невозможно восстановить его функциональность на границе сепулькария и окружения.

--- компьютерную модель здания для сепулькария ### Нет, компьютерная модель расскажет только про устройство здания сепулькария, но ничего не скажет про сам сепулькарий.

--- исходные тексты компьютерных программ управления сепулькарием ### Нет, тексты компьютерных программ прояснят устройство компьютерных программ сепулькария, но ничего не скажут про его функциональность.

--- фотографии сепулькария ### Нет, по фотографиям будет понятно немного, как устроен сепулькарий (видны конструктивные части, аффордансы для функциональных частей), но вот что он делает — этого понятно не будет, функциональности на фотографиях не найдёшь.

Моделирование: концепция использования

Заполните табличку для трёх и более целевых систем известных вам проектов

Номер п/п	Целевая система	Примеры из концепции использования (что делает целевая система в надсистеме)	Надсистема	Примеры потребностей (что делает надсистема, что удовлетворит причастные к ней роли — создателей, владельцев, пользователей и т.д.)
Пример	Игровой сеанс	Игровой сеанс в игре даёт игроку непрерывное развлечение в тот момент, когда он этого захочет на столько, насколько он этого захочет. Игровой сеанс генерирует денежный	Игра как последовательность игровых сеансов	Игра удовлетворяет интерес игрока чередовать развлечения и другие деятельности Игра увязывает игровые сеансы между собой, уменьшая

		поток игродела		время создания игрового сеанса и облегчая учёт потраченного времени для облегчения биллинга в пользу игродела

Системы систем

Иногда систему, которая состоит из других систем, называют «системой систем». Это неправильно, это просто система, а другие системы в ней — это подсистемы. Вроде как не нужно специально подчёркивать тот факт, что любая система состоит из систем. Тем не менее, термин **система систем** (system of systems, SoS) есть, он закреплён за особым случаем выделения систем по социальным характеристикам систем их создания и особенностям времени создания, а не по чисто техническим особенностям вложенности систем друг в друга в окружении и работы в ходе использования.

Системой систем называют такую систему, которая имеет (критерии Maier[302]):

- независимые системы создания/enabling её подсистем (у ответственного за SoS нет полномочий предписать общее развитие-модернизацию для входящих в SoS систем)
- подсистемы, которые могут работать независимо от существования целевой системы (нет полномочий по указанию владельцам систем, как им работать)
- эмерджентность/системный эффект от объединения в систему (кто-то::создатель желает получить от целевой системы систем функцию/поведение, которое невозможно получить от работы с входящими в систему систем отдельными системами, и требуется совместная работа этих входящих в SoS независимых **составляющих/constituent систем**).
- эволюционное развитие (понимание того, что будет происходить в системе систем на каждом следующем шаге программы создания системы, требует исследований и дополнительных согласований, ибо нет роли, которая знает, как в каждый момент устроена система систем — подсистемы системы систем как автономные системы меняются независимо. Поэтому идёт череда модернизаций, а не разовое спроектированное проектное действие. Впрочем, это мало сегодня отличается от «обычной инженерии», ибо и с «просто системами» тоже ведут эволюционное развитие, хотя в этом случае про устройство системы организации инженеров всё известно, подсистемы не

[302] http://sebokwiki.org/wiki/Systems_of_Systems_(SoS)

автономны).

- географическое распределение входящих в SoS систем.

Эти критерии различаются, конечно, в разных инженерных и менеджерских школах, но общее остаётся:

- обычные «системы» подразумевают централизованное «владение» системой — наличие проектных ролей/stakeholders/«заинтересованных сторон», полномочных принимать решения по всем частям системы, полномочных распоряжаться всем, что в границах их системы. Это традиционный случай: автомобиль с двигателем и колёсами, железнодорожный мост и компьютер — это типичные «просто системы», у них есть свои системные инженеры, которые полностью определяют их функции, конструкцию, возможности по работе в том или ином окружении, составляют и выполняют планы по модернизации и выводу из эксплуатации. У каждой из этих «просто систем» есть один хозяин, один владелец.
- в системе систем каждая из входящих в неё систем имеет своего хозяина, и входящие системы могут работать/эксплуатироваться/функционировать автономно, без вхождения в систему систем. Это город, транспортная сеть, интернет, любая компания — ведь люди в ней самопринадлежны и не торопятся выполнять чьи-то приказы.

Тем самым разница между «просто системой» и «системой систем» определяется не через особую структуру или конструкцию системы (определяемую во время её использования), а через наличие независимых друг от друга систем создания (определяемых в момент создания), описывающих и воплощающих входящие в систему систем отдельные составляющие/constituent системы, а затем имеющие возможность независимо их использовать. Поэтому метафорически систему систем мы представим диаграммой, где показано, что система систем состоит из трёх подсистем («составляющие системы» системы систем), каждая из которых автономна и в свою очередь состоит из своих подсистем, показано два системных уровня), но главное — фигурки с портфелями указывают на агентов-собственников, которые должны договориться о том, чтобы из автономных систем получилась целая система.

В системе систем важны прежде всего владеющие частями-системами и полномочные в распоряжении ресурсами труда и капитала агенты (люди, предприятия), именно они делают систему систем/system of systems/SoS особым случаем.

В ISO/IEC/IEEE 21841:2019 выделено четыре типа систем систем, отличающихся

степенью их автономности:

- **управляемые** (directed), в которых есть назначенный архитектор (агент в роли архитектора, то есть отвечающего за деление системы на конструктивы и способы организации связи между конструктивами), который может выдавать свои архитектурные решения командам проектов составляющих систем и менеджер (агент в роли менеджера), который распоряжается общими ресурсами.
- **подтвержденные** (acknowledged), в которых признаваемый архитектор есть, но он может только уговаривать владельцев составляющих SoS систем изменить свои системы согласно разработанной им архитектуре (то есть у архитектора нет отношения надзора/governance, только менторинг по поводу принимаемых им архитектурных решений).
- **сотрудничающие** (collaborative), в которых владельцы всех составляющих систем договариваются друг с другом «по каждому чиху», но нет оргзвеньев (агенты, выполняющие роли) архитектора, менеджера проекта или аналогичных выделенных оргзвеньев, занятых созданием и развитием SoS на уровне целой системы.
- **виртуальные** (virtual), в которых владельцы систем, составляющих SoS, вообще не знают друг о друге ничего, и тем самым команды проектов создания и развития составляющих систем не влияют друг на друга явно.

Постепенно за пару десятков лет путём проб и ошибок был выведен основной способ работы с системами систем как «лучшая практика»: совместная постепенная асинхронная эволюция/«непрерывная модернизация» входящих в систему систем автономных систем — ибо согласованность и синхронность изменений в этих автономных составляющих системах «по единому проекту из единого центра» обеспечить крайне сложно.

Даты утверждения проектов модернизации отдельных составляющих систем будут различаться, время появления эмерджентной функциональности целевой системы систем тем самым плохо прогнозируется, получаемое на модернизацию систем-конституентов/«составляющих систем» финансирование будет выделяться в разные моменты и нельзя будет гарантировать его достаточность, некому будет вести общий проект реформирования как инженерно, так и менеджерски, не говоря уже об отсутствии общего для всех целеполагания/стратегирования, ибо нет общего для всех бизнесмена, который бы его координировал и затем был бы полномочен наказывать за отклонение от стратегии.

Хозяева автономных составляющих/constituent систем могут иметь разные мнения по поводу взаимодействия их систем с другими составляющими системами в составе системы систем, они могут сопротивляться переменам, ибо их вполне может удовлетворять и автономная работа их систем в их других надсистемах, а желание какого-то даже влиятельного исполнителя внешней проектной роли объединить автономные системы в ещё одну надсистему как общую систему систем они могут не поддерживать.

Для работы с системами систем тем самым необходимы работы методами лидерства. Лидерство понимается как метод катализирования сотрудничества создателей: это «режиссёрские» методы постановки спектакля (время создания спектакля!) в театральной метафоре, а именно методы уговора исполнителей ролей исполнять их

роли в выбранной режиссёром к постановке пьесе. В игровой метафоре — это методы организации игры «мастером игры», который озабочен тем, чтобы все «участники игры»::агенты согласованно и этично отыгрывали свои роли в заданной сценарием игры культуре/стиле. Лидерство — это «метод помощи в занятии позиции». Конечно, предполагается умение «убалтываемого» на выполнение роли агента вести себя дальше в соответствии с методом, наличие мастерства в том методе игры/работы, который будет выполняться по итогам работ лидерства.

Сегодня SoTA-метод лидерства — распределённое/shared лидерство[303] (нет одного «режиссёра», все выполняют роль лидера, каждый вокруг себя). Но в любом случае тут идёт опора на гуманитарные дисциплины/теории социологии, политологии, психологии, конфликтологии и т.п. Была надежда, что выделение «системы систем» как отдельного класса систем что-то даст для того, чтобы лучше организовать их создание. Но ничего не получилось. Все достижения методов «инженерии систем систем» (systems of systems engineering) на поверку оказываются достижениями других методов работы, главным образом менеджмента. Частично используются результаты развития методов системной архитектуры, которая пытается обеспечить максимальную автономность работы команд в «обычных проектах». Эти же результаты применимы и в проектах создания и развития систем систем.

Никаких особых прорывов в работе с системами систем не случилось, никаких особых методов работы с системами систем так и не было найдено. Само понятие системы систем отразило понимание, что в проектах создания и развития систем не обойтись без идей второго поколения системного мышления «агенты-создатели крайне важны в рассмотрении проектов создания каких-то целевых систем, нельзя системы рассматривать в отрыве от их создателей». Поэтому **мы не будем уделять в нашем руководстве специального внимания системам систем.**

Можно заметить, что сообщества (где все заняты своими делами, своими проектами, в отличие от организации), общества и человечество («агентечество») в целом — это формально тоже системы систем. Другое дело, что трудно представить себе целевую систему, которую они делают как создатели — кроме воспроизводства/репликации самих себя и удерживания какой-то автономности от других систем, то есть разговор тут примерно такой же, как про биологические системы, инженерный подход тут только-только начинает рассматриваться как возможный. Ну, и создатели таких систем вызывают вполне оправданное подозрение: история уже насмотрелась «отцов народа», «вождей народа» и прочих диктаторов, которые формально объявлялись создателями каких-то больших сообществ и обществ, но считать ли проекты социальной инженерии успешными проектами — это большой вопрос.

Основной аргумент против инженерного рассмотрения изменений сообществ и обществ: «люди в этих системах ничего не понимают, оставьте это эволюции, как и в случае существ-людей: евгеникой не занимайтесь, людскую породу улучшать не нужно, ибо даже если это и безопасно, то это породит неравенство и прочие неприятности по линии справедливости и этики». Основной контраргумент: «эволюционные пробы дарвиновской эволюции в большинстве своём приводят к ошибкам, результаты — вымирание. Может, не будем вымирать, перейдём к техно-эволюции со смарт-мутациями, а не обычными? Если больным и слабым людям не даём умереть и вылечиваем-выучиваем, то почему бы не делать того же самого с

[303] https://en.wikipedia.org/wiki/Shared_leadership

сообществами и обществами, а не пускать их существование на самотёк?».

Обратите внимание на безличные предложения в предыдущем абзаце: кто будет агентом, который сможет инженерно поменять общество, «вылечить-выучить»? Диктатор с инструментом изменений таким, как «небольшая армия»? Сладкоголосый политик? А если будет трое таких политиков, и все будут говорить разное, и у каждого политика вдобавок будет небольшая армия — оставить всех троих? Так это ж и есть та самая эволюция в итоге, и дальше только выбирать — она будет дарвиновской или техно-эволюцией. Увы, развитой теории коллективных агентов, которая давала бы возможность предсказывать результаты эволюции, нет. Поэтому возможности даже системного мышления в проектах социальной инженерии (инженерия систем уровня выше организаций, где понятна структура распоряжения трудом и капиталом) ограничены. Но они достаточны, чтобы не влезать в утопии (скажем, чтобы не строить социализм, опираясь на утверждение «все прошлые попытки были плохими, потому что не я делал, а у меня получится, я же хороший и люблю бедных»).

Интеллектуальные агенты в системах

Человек, равно как и AI-робот в составе систем учитывается сложно, ибо он **одновременно** может быть

- и целевой системой («продуктом»)
- и надсистемой, где чаще удобней рассматривать даже не всего человека, а его часть — часть тела::hardware или мастерство как часть личности::software,
- и материальным «оборудованием» типа станка в составе создателя (если он делает что-то руками/манипуляторами, а всякие молотки и стамески и даже моделеры тут просто продолжение его рук или мозга/вычислителя),
- и быть конструктивом/«материальным телом» с его весом и инерцией, влажностью и твёрдостью (как, например, в учёте инерционного движения в телесных практиках или робототехнике или в примере с наручными часами),
- проектной ролью, или даже целым набором проектных ролей: мы можем не учитывать мнений и интересов животных (хотя и это плохо), но мы принципиально не можем не учитывать мнений людей-в-ролях, вынуждены считаться с решениями AI и даже просто компьютерных программ (помним про «я бы помог, но компьютер не даёт»), и ещё решения организаций как «юридических лиц».

А ещё у людей и организаций есть отношения подчинения/руководства и отношения собственности по поводу вещей, и люди (и организации, а ещё и AI-агенты) занимают какие-то должности. И ещё они «принадлежат» каким-то государствам. С агентами сразу всё сложно: агенты сложны, их поведение плохо предсказуемо, они сложно описываются. Перед тем, как делить сложное мышление проекта между разными агентами (сегодня это не только люди!), мы рекомендуем отмоделировать и документировать то, что вы знаете про агентов, чтобы не запутаться и чего-то не забыть. Про людей, их AI-системы, команды и коллективы (команды команд), организации (предприятия и расширенные/extended предприятия/enterprises) вы будете думать и писать неожиданно много, мышление об интеллектуальных агентах и их группах не самое простое. Если подниматься ещё выше по уровням сложности, то это будут уровни сообществ (в том числе бизнес-эко-систем), обществ и человечества (а сегодня ввиду разнообразия агентов, правильнее бы называть

«агентечества») в целом.

Люди могут договариваться друг с другом по самым неожиданным поводам и предпринимать по линии этих договорённостей самые неожиданные для команды проекта действия, причём из лучших побуждений. Скажем, вы поручаете заводу-изготовителю сделать по вашей 3D инженерной модели медный теплообменник. Инженеры этого завода вдруг находят, что меди там многовато, и они из лучших побуждений убирают медь — меньше материалоёмкость, более тонкие стенки, лучше теплообмен! И изготавливают «улучшенный вариант, наши улучшения — это наш бонус вам». Вы включаете теплообменник, тонкие стенки под напором жидкости начинают гудеть громче, чем концертный орган! Теплообменник стал генератором вибраций, источником шума! Вот так бонус! «Хотели как лучше, получилось как всегда» — это весьма распространённое явление в проектах по созданию и развитию систем, поэтому-то любые изменения и имеют статус «гипотезы», всегда надо быть готовым к тому, что «гипотеза не подтвердилась». И ведь без «инициативы снизу» проекты тоже не живут!

Лучшие методы менеджмента и инженерии (например, культура архитектурной работы) как раз таковы, что неизбежные ошибки, неизбежные мутации не приводят к потере работоспособности системы, к необходимости переделывать систему с нуля. Подробнее эти методы/способы/практики разбираются в руководствах по системной инженерии (особенно тут полезен метод эволюционного принятия архитектурных решений в ходе инженерного проекта) и по системному менеджменту.

Агенты, в том числе и коллективные агенты обычно участвуют ещё и сразу в нескольких командах, которые заняты каждая в нескольких проектах по созданию и развитию систем. Помним, что в области интереса (набор предметов интереса в какой-то одной предметной области) у каждой выполняемой агентом роли есть не просто предпочтения в каких-то значениях интересных для них характеристик, но исполняющие эти роли агенты будут менять свои роли, менять себя (и развивать тело и экзотело, и развивать личность, осваивать новые виды мастерства), менять окружающий физический мир для реализации этих предпочтений. Агенты (люди, AI-агенты, их организации) дьявольски изобретательны в достижении предпочтительного для них состояния мира. Они будут влиять на ход проекта по созданию системы и на получившуюся систему самыми неожиданными для друг друга способами. Кто-то::агент упростит целевую систему инженерно без потери функциональности, кто-то договорится о дополнительном финансировании, кто-то договорится про такое изменение надсистемы, при котором целевая сможет быть дешевле. А кто-то провернёт мошенничество в надежде, что это или никто не заметит, или заметит — но ничего не сможет сделать без сильного ущерба для заметившего. Агенты изобретательны и деятельны. Посмотрите в зеркало: вы и есть такой агент, все остальные вряд ли хуже!

Пример системы с входящими в её состав людьми у нас в руководстве уже разбирался — это танцевальный перформанс/выход. **Разворачивающийся во времени процесс смены состояний перформанса задаётся на системный уровень ниже как перечисление смены состояний входящих в него взаимодействующих физических объектов, причём эти объекты выделяются вниманием в момент проведения перформанса.** Не следует останавливать взаимодействие частей-подсистем перформанса и разбирать перформанс::систему на конструктивы во время

эксплуатации/работы/функционирования (пытаться представить взрыв-диаграмму), надо не разбирать систему, а выделять в работающей системе части вниманием.

Танцевальный перформанс вполне можно представить как систему, только его нужно представлять в пространстве-времени, а не только в пространстве. Основные функциональные части-подсистемы тут будут танцующие::функция танцоры::роли/подсистемы. В перформансе люди-агенты выступают и как материал/конструктив (физические тела, меняющие форму в пространстве-времени), и как функциональные объекты — проектные роли танцоров. Танцевальный перформанс не обладает сложностью предприятия и тем самым не требует рассмотрения сложных вопросов корпоративного управления, стратегирования, операционного менеджмента/управления работами. Танцевальные перформансы бывают и сольные, и социальные (парные, друг для друга, а не для зрителей), и групповые/ансамблевые. Кроме танцоров там нужно учитывать место перформанса (помещение или даже улицу для уличных выступлений), зрителей, иногда судей (танцевальные конкурсы), музыку и музыкальную аппаратуру. Танцуют не всегда люди, иногда это киборги (скажем, танцор с дополнительными четырьмя робототехническими руками в рюкзаке[304]). Это отличный пример, на котором можно тренировать своё системное мышление[305].

Мероприятия так же являются классическим примером систем с людьми. Методы менеджмента мероприятий (event management[306]) даже стали университетской учебной дисциплиной — создать и развить (скажем, провести концертный тур) концерт или ежегодный рок-фестиваль трудно, но ничего необычного в этом сегодня уже нет.

Рассмотреть тысячи людей в составе системы-мероприятия вполне возможно, серию мероприятий/events (например, ежегодный фестиваль) можно рассматривать как изготовление непрерывно развивающихся версий системы в машиностроении или выпуск версий софта. Мы тут намеренно используем инженерную терминологию («изготовить концерт», «выпустить фестиваль»), чтобы показать общность мышления для самых разных систем. Конечно, в менеджменте мероприятий используется другая терминология, но сам выбор слов-терминов тут не так важен. Важно, что при мышлении о мероприятиях/events вы выбираете типы системного мышления и используете понимание того, как изготавливают успешные системы из системной инженерии. Сами способы/практики работы (культура работы, стиль работы, деятельность, метод — термины тут не так важны) в event management будут даны в прикладных курсах/стандартах/руководствах/регламентах, но увязать всё это мышление о самых разных подпроектах (скажем, договориться с инженерами по звуку, договориться по поводу вентиляции и т.д. — чисто инженерные подпроекты) можно именно на базе использования системного мышления. Важно обращать внимание на общий способ рассуждений во всех проектах, связанных с созданием систем с участием в них агентов (людей, AI-роботов и даже организаций) как подсистем.

Домашнее хозяйство, где есть дом с утварью и его жильцы (в том числе роботы), уже рассмотренный случай потребителя одежды и гаджетов — это всё примеры

---

[304] https://vk.com/wall-179019873_1747

[305] Тексты про танцы и системное мышление публикуется в https://vk.com/buffdance

[306] https://en.wikipedia.org/wiki/Event_management

систем с агентами-людьми.

Сложнейшими системами с людьми и AI-агентами являются такие групповые **создатели/enabling/constructor systems** как предприятия, рабочие группы проекта, расширенные предприятия (extended enterprise, предприятие с его контракторами, занимающееся какой-то большой целевой системой, например, атомной электростанцией, или серийным выпуском сложных транспортных систем — самолётов или автомобилей).

Системы с людьми (включая их организации) — это обязательно системы систем в силу того, что люди обладают свойством самопринадлежности. Поэтому с системами с людьми в их составе нельзя работать простыми инженерными методами, по которым можно сконструировать простую механическую или механическую с элементами электроники систему, изготовить её части и собрать их в работоспособное целое. Нет, **метафора часовщика** с изготовлением деталей и их сборкой не работает. С людьми (как и любыми другими живыми «выращиваемыми» системами, а также обучаемыми AI-системами) больше работают «сельскохозяйственные» **метафоры:**

- в небольших системах систем **садовника** (который имеет контроль над тем, что создаёт, успевает позаботиться о каждом росточке на каждой клумбе, знает положение каждого дерева и ухаживает за ним),
- больших системах систем **лесника** (который не имеет контроля над своим лесом — где какое дерево или кустик вырастет, но тем не менее лесник имеет достаточно влияния, чтобы предотвратить какие-то серьёзные негативные события: может предотвратить пожар, подкормить зимой животных, отогнать браконьеров).

В особо крупных системах (большое сообщество, общество в масштабах государства, всё человечество) иногда говорят уже не просто о **сложности** системы, а **сложностности** или сложносистемном мышлении[307], что не позволяет как-то строить действия с предсказуемым результатом — результаты проектов, имеющих дело с большим количеством людей, всегда в значительной мере неопределённы, они описываются вероятностно, хотя и детерминированы. Детерменированы — это означает, что в них нет места «случайности», всегда есть причины, но вот предсказать итоговый результат — нельзя[308], поэтому его и описываем вероятностно (случайность и вероятность — разные понятия). Нужно всегда помнить, что эволюция порождает сложность, порождает всё более и более высокие уровни организации, и действующая сила этой эволюции — конфликты между системными уровнями, ведущие к неустроенностям[309].

В особо крупных и сложных системах из людей, роботов, инструментов/оборудования, то есть сообществах (включая сообщества предприятий — эко-системы), обществах, человечестве (с учётом всех остальных агентов более подходит «агентечество», а не «человечество») в целом всё настолько сильно связано друг с другом и настолько неожиданным образом реализация намерений одних агентов сопротивляется или помогает реализации

---

[307] См. литературу в П.К.Гречко «Сложносистемное мышление: методологические перспективы. Парадигмальная эвристика complexity в современном социально-гуманитарном познании», https://journals.rudn.ru/philosophy/article/view/11283/10713

[308] Различение случайности и детерминистичности в физике: https://arxiv.org/abs/1507.03287

[309] https://elementy.ru/nauchno-populyarnaya_biblioteka/434505/Konflikty_kak_osnova_slozhnosti

намерений других агентов, что «неожиданности» являются нормой, отсутствия неожиданностей практически не бывает, «лучших решений» нет, есть только «наименее плохие из заведомо плохих».

Нужно помнить, что предприятие или даже команду нельзя проектировать, а потом изготавливать как компьютерный чип или водонапорную башню. Системы с агентами должны обсуждаться с использованием системного мышления, но вот замышлять их, создавать, развивать, эксплуатировать, ликвидировать их методами классической «железной» или программной инженерии нельзя, там нужно использовать методы обучения: инженерии личности для индивидуальных агентов, системного менеджмента для коллективных агентов (развитие предприятий — это как раз обучение новым методам работы коллективных агентов).

Трудно представить, что левый ускоритель в ракете будет подговаривать правый ускоритель лететь не к Марсу (куда они должны лететь), а к Луне — потому что так надёжней и быстрей, заведомо хватит топлива и менее хлопотно. А вот в системе из интеллектуальных агентов такое бывает сплошь и рядом. **Осторожней с применением технических инженерных методов работы с «железом» и «софтом» к системам с людьми и AI. Хотя так же осторожно нужно относиться и к обратной ситуации — неприменению общих системно-инженерных методов работы, которые являются обобщением как раз методов работы с «железом» и «софтом» к системам на всех эволюционных/системных/организационных уровнях (уровнях сложности), включая не только инженерию предприятия, но и социальную инженерию сообщества, общества и даже человечества в целом.** Обсуждение последствий этого неприменения см. в работах John Doyle[310].

Рассуждения на тему систем с искусственным интеллектом (AI), начиная с какого-то уровня агентности таких систем (возможности генерировать предсказания будущих состояний себя и мира, идей по улучшению таких состояний, выбора лучших из этих идей, а затем планирование и реализация планов по воплощению этих идей в жизнь), рассуждения о людях, людях в их коллективах, людях в их коллективах с AI (например, предприятиях) — это одни и те же рассуждения, системное мышление безмасштабно и неантропоцентрично.

Системная этика

Этика в системном мышлении рассматривается как метод/практика/культура минимизации конфликтов между интеллектуальными агентами по поводу их действий, затрагивающих интересы других агентов. Этика также легко распространяется с конфликтов интеллектуальных агентов на самые разные конфликты агентов более высоких системных уровней, состоящих из интеллектуальных агентов: организации, сообщества, общества и человечества/агентечества в целом. Иногда эти уровни называют «социальными», то есть относящимися ко многим людям.

Этика — это метод из интеллект-стека, а системное мышление рассматривает этику как ведущий способ рассуждений про успешность систем в условиях, когда существуют межуровневые конфликты. Рабочий создал пушку, для него это

[310] https://ailev.livejournal.com/1622346.html

хорошо — он получил зарплату. Предприятие получило за пушку оплату. Но это же орудие убийства! Сообщество пацифистов считает, что создание пушки — абсолютное зло. Сообщество любителей своего государства считает, что пушка может быть никогда не применена, но её наличие отвадит потенциальных захватчиков. Сообщество нелюбителей другого государства считает, что пушку надо немедленно употребить в дело, раз уж она есть. Большинству других людей из общества всё равно, что там с пушкой и её использованием. Где-нибудь в ООН пытаются уменьшить число пушек на планете, «сокращение вооружений, укрепление безопасности в масштабах мира». На уровне страны (её органов управления) наличие пушки — «рост вооружения, укрепление безопасности в масштабах страны». На уровне сообществ внутри стран (разных! Сообщества одних стран вполне имеют мнение как по поводу пушек своей страны, так и по поводу пушек других стран) — мнения резко отличаются. Что тут считать успехом проекта создания и развития пушки? Что успех с точки зрения одних ролей — это полная неудача с точки зрения других ролей!

В труде/деятельности/практике/культуре/стратегии/инженерии/методе работать без этики нельзя, и лучше бы при этом быть знакомым с этическими аргументами из современных (а не устаревших, например, где можно спокойно убивать «неверных») книжек, чем «голосовать сердцем»[311].

Так, в 2022 году решили проверить, как хорошо работает программная система искусственного интеллекта, которая была разработана для предсказания влияния молекул на нервную систему, что дальше должно было использоваться для позитивного влияния, то есть предлагалось с помощью этой системы разрабатывать лекарства (инженерный проект, где проектировалось вещество, которое должно было дальше работать с существом — обычная инженерия с несколькими уровнями). Эта система искусственного интеллекта так же хорошо, как предлагала лекарства, предложила после вопроса о максимизации негативного влияния на нервную систему новые нервно-паралитические боевые отравляющие вещества (включая переизобретение уже имеющихся на вооружении).[312]

Вопрос об этичности применения искусственного интеллекта, как и применения любой системы, относится и к применениям (включая самоприменение, по собственной инициативе) не только искусственного, но и «мясного» интеллекта, человеческого, а также любой смеси этих интеллектов, а также коллективного интеллекта групп людей и их компьютеров. Помним, что конфликты между системными уровнями есть всегда, их много. Но есть и конфликты между конкурирующими интеллектуальными системами одного уровня. То, что хочется членам одних сообществ, будет не нравиться людям из других сообществ, общество в целом может относиться «в среднем» как-то ещё третьим образом, а каждый отдельный член этих сообществ и обществ будет хотеть чего-то и для себя тоже, не захочет приносить в жертву свои интересы интересам сообщества, или общества, или даже человечества в целом — и оптимальных решений тут не будет.

Этика рассматривает распространение каких-то идей, которые являются вредными, как зло — некоторые мемо-вирусы приводят при бездумной передаче их на исполнение к исключительно неэтичным последствиям. И тут два варианта: вообще

---

[311] Материалы по этике на русском в каком-то количестве есть вот тут, но там не обсуждается многоуровневость этики: https://vk.com/reducing_suffering

[312] https://www.nature.com/articles/s42256-022-00465-9

не ограничивать выдачу каких-то идей (позволять рекомендациям быть любыми по содержанию, «полная свобода слова»), либо вводить рекомендательную цензуру, то есть запрещать распространение каких-то идей — рекламы финансовых пирамид, расовой ненависти, наркотиков и такой список можно продолжать и продолжать, далее мы быстро попадаем в запреты по политической линии. Так, в некоторых странах ругать действующего президента — это обычная политика, а в некоторых странах политики нет, и ругать действующего президента нельзя.

Если рекомендательную цензуру снять, то выдача будет сплошь из мемовирусов — они ведь будут отбираться рекомендательным алгоритмом просто по признаку их «заразности». СМИ и социальные сети выбирают новости, которые не полезны людям, не полезны обществу в целом, но которые полезны организациям-провайдерам этих СМИ и социальных сетей, то есть которые увеличивают потребление новостей. Новость «в Африке родился ребёнок с тремя головами» точно будет рекомендована большинством «бесцензурных» алгоритмов на основе того, сколько человек ей заинтересуются по сравнению с новостью «присуждена очередная Нобелевская премия по физике». Но если вдруг очень хорошо написать призыв убить всех рыжих и он станет вирусным мемом? Надо ли отфильтровывать такой мем, или оставить «свободу слова»?

Если запрещать вредные (для кого-то вредные, а для кого-то самая что ни наесть полезная пропаганда) идеи и ограничивать бесполезные, то нужно ли наоборот — «насаждать добро» в рекомендациях? Скажем, давать новостную ленту, которая будет отфильтровывать все фейки и подозрительные «бесполезные новости», зато выдавать интереснейшую ленту, например, научно-популярных новостей, за счёт переформулирования их средствами AI в интереснейшие рассказы? Да, скажут просветители. Психологи скажут: это манипуляция, вы подсадите людей на чтение, и без разницы содержание этого чтения, вы будете стимулировать «залипание» в любом случае — а работать когда?! Чью рекомендацию выполнять — просветителей или психологов?

Конфликты, везде конфликты! Теория эволюции, обсуждающая неустроенности (множественность одинаково квазиоптимальных конфигураций систем с чрезвычайно лёгкими переходами между этими конфигурациями) из-за этих конфликтов, говорит, что стабильных решений тут не будет, будет вечный поиск не устраивающих никого компромиссов, и будет огромное число разных компромиссов примерно одинакового качества с очень редкими улучшениями, чаще всего связанными с появлением ещё одного системного уровня — будет происходить рост сложности.

Этичность точно так же рассматриваться должна без какой-либо индивидуальной, расовой/религиозной и даже видовой пристрастности. Деантропоморфизация этического мышления будет полезна в том числе и для людей. Этична ли смерть? Ведь программируемая смерть[313] — это свойство всей жизни! Один ответ — нет, ни разу, ни в коем случае, а другой ответ — да, ибо эволюция в ней нуждается!

А может быть тогда не смерть, а «эволюционная модернизация» по линии трансгуманизма? Выход из биологической эволюции в техно-эволюцию (геном, хранимый за пределами каждого организма в лаборатории генной инженерии)?

Для такого инженерного решения, выходящего за рамки

[313] https://arxiv.org/abs/2207.04886

узковидовых/парохиальных этических приоритетов, мы сталкиваемся с пересечением инженерного деятельного (изменяющего себя и мир) мышления и этики. **Этика в инженерном мышлении есть всегда, её нельзя выкидывать — и системной инженерии без этического рассмотрения быть не может.**

Если вы создаёте простейшие столовую вилку, ложку и нож, то ими тоже можно убить — использовать как оружие. Дальше вопрос, который любят задавать юристы: а намерение какое? Столовое, «двойного назначения», или таки «чтобы убить было легче, чем кушать» (маскируем «боевой клинок» и «боевую вилку» под столовые нож и вилку)? С ножом пример не случаен, ибо в самолёте у вас и маникюрные ножницы могут иногда отобрать как «возможное оружие» (но производителю их такое использование и в голову не придёт!).

Ещё пример: дети могут сунуть руку или железный гвоздь в розетку — запретим 220 вольт в домашней сети? Что, не все дети это будут делать, а только очень некоторые, а электричество очень полезно для освещения и стиральной машины? Ну, «мы» (кто «мы»? где проходят границы этого «мы», кто входит в состав «мы»? кто выражает мнение этого «мы»? почему выразители противоположного мнения не должны включаться в «мы», а мнение этого «мы» после этого пересматриваться?) за ценой не постоим, детские жизни дороже! Переходим на 12 вольт, хотя понимаем, что и потери на таком напряжении будут выше, и провода заведомо толще! Но дети, дети с их гвоздём в розетке!

Этих примеров множество. В социальной сети с миллиардами постов взрослые могут написать неправильный с точки зрения каких-то людей (но вполне ОК с точки зрения многих других людей) пост, а потом какие-то люди случайно (ну вот как дети, которые случайно суют руку в розетку) могут там этот неправильный пост прочесть. Решение: запретим все миллиарды постов соцсети в страновом интернете, чтобы нельзя было случайно прочесть эти единичные посты! Что, социальные сети для чего-то там полезны? «Мы» (те же вопросы про «мы»! всегда их задавайте, когда вам говорят от имени «мы»!) за ценой не постоим, неправильных 100 постов в сутки оцениваем как больше вреда, чем может оправдать вся польза от остальных 10млн. постов в сутки — а выкинуть миллиарды уже написанных постов, так миллионы людей, которые писали эти посты, спокойно это переживут!

Мышление в части задействования этики (равно как и в плане системного мышления, которое отправляет мышление во время ситуации использования целевой системы прежде всего, даже если рассмотрение идёт в ходе создания системы) выключать нельзя, любой ведь инженерный объект можно представить как «двойного назначения», а кроме этики сравнивать пользу и вред — нет такого другого метода мышления (включая обсуждение вопроса, можно ли сравнивать пользу и вред — вопрос об утилитаризме[314], и вопрос о том, должны ли мы судить о делах по этим делам — деонтика[315], или таки включать обсуждение последствий по линии причин-следствий, консеквенциализм[316]). Тут главный вопрос: бытовое ли у вас рассуждение про методы этики, или вы понимаете, какие могут быть аргументы в пользу того или иного решения в силу того, что хоть что-то изучали из

[314] https://plato.stanford.edu/entries/utilitarianism-history/
[315] https://plato.stanford.edu/entries/ethics-deontological/
[316] https://plato.stanford.edu/entries/consequentialism/

теорий/объяснений/знаний/дисциплин по этой линии, а не просто «голосуете сердцем», решаете интуитивно. И понимаете ли вы, что «абсолютного решения», «верного ответа» в этике не будет, ибо неустроенности (аналоги geometrical frustrations) в системах неустранимы, сложность систем будет только расти в ходе эволюции, разные варианты этических решений будут расти в разнообразии, но существенные улучшения будут не так часты, как этого хотелось бы (все закономерности эволюции и техно-эволюции работают и в случае этики).

В любом случае: если не рассматриваются интересы разных системных уровней, то это не современная этика. Системная этика учитывает, что интересы одного человека конфликтуют с интересами команды (он хочет сходить в туалет, но вот команда ведёт совещание, и приходится воздерживаться от такого похода в туалет — или наоборот, не воздерживаться, и команда оказывается недовольна), само заседание команды может не нравиться какому-то сообществу, ибо команда будет обсуждать «контркультуру», а сообщество хранит культуру. Члены сообщества знают, что каждая культура в какой-то момент была контркультурой, это написано во всех книжках по культуре! Так что конфликт между сообществом и командой есть. Общество же и эту культуру сообщества может считать неприемлемой — но тут стоп: а кто именно представляет «общество»? У членов общества могут быть крайне разные мнения насчёт сообщества и поддерживаемой им культуры (это сообщество как раз и выделяется изо всех членов общества тем, что каждый его член практикует какой-то метод работы, какую-то деятельность/культуру/практику/стиль), а ещё значительной части общества может быть вообще всё равно. Повторим: **на вопрос этики «что такое хорошо и что такое плохо» ответ не засчитывается, если нет системного рассуждения, то есть рассмотрения происходящего на разных системных уровнях, вскрытие межуровневых конфликтов.**

Подробней место метода/культуры этики в интеллект-стеке фундаментальных методов мышления рассматривается в руководстве по интеллект-стеку.

Вот несколько примеров конфликтов, ведущих к этическому разбирательству между объектами разных эволюционных/организационных/системных уровней:

- Человечество — мы тут одни на маленьком глобусе (человечество со всеми его людьми и компьютерами, но не все агенты будут человеками, тогда трансгуманистически назовём «агентечество» или ещё как), но планируем распространяться по космосу. Этично ли это? Или мы будем «загрязнять космос» и нужно самоубиться согласно слоганам «зелёных» (в случае безжизненного Марса — «красных»)?
- Общество — каталлактика[317]/экономика, политии[318] (в том числе государство). Желающих рассказать, что этично или что неэтично делать с обществом (в том числе кому делать — кто или что агент таких изменений, и каким способом этот агент будет работать) по части его нарезки на государства, ограничения торговли и т.д. можно найти в любой рюмочной, в любой пивной, что уж говорить о редакциях СМИ, где журналисты с их профильным чтением художественной литературы имеют своё просвещённое мнение по всем этим вопросам, а уж блогеры так и вообще тут профи! Вопрос

[317] https://ru.wikipedia.org/wiki/Каталлактика
[318] https://ru.wikipedia.org/wiki/Полития

привязки к территории тоже обсуждается, и ответ на него не так очевиден[319]. Или такие вопросы как здоровье (санитарное государство: локдауны и карантины при любых подозрениях на эпидемию, принудительные вакцинации и другие медицинские процедуры, назначаемые без особых на то оснований — а кто и как будет определять основания? Суды? Эксперты? Они врут более часто, чем можно ожидать. То есть они не врут, но два эксперта — три мнения, это называется «шум» в решениях[320]).

- Сообщество — споры между разными культурами и субкультурами как раз здесь. Когда-то рейверы (сообщество, члены которого практиковали посещение рейвов и ночных клубов для коллективных танцев) и брейкеры (сообщество, члены которого практиковали уличные танцы и мероприятия с конкурсами-баттлами) Москвы не любили друг друга и ходили драться друг ко другу на мероприятия. Это очень не нравилось организаторам мероприятий (другой системный уровень, чем у участников мероприятий, другие цели, другие представления об этичности тех или иных поступков). И они придумали ход: лидеры и рейверов, и брейкеров в одностороннем порядке выпустили для своих сообществ декларации о том, что «мы считаем ниже своего достоинства задирать этих не совсем людей, мы их вообще замечать не будем!». Тексты были идентичны, хорошо восприняты каждым из сообществ, драки прекратились — сообщества демонстративно друг друга не замечали. Это и есть социальная инженерия на уровне сообществ.
- Коллектив/организация, которая трудится — это проекты (организованные коллективы, то есть с понятными ролями и полномочиями каждого). Организация заинтересована платить меньше налогов, платить работникам меньше зарплаты, брать с потребителей за свой товар или сервис побольше. И это только часть противоречий. Один работник «подставил» другого (дал возможность реализоваться заведомо провальному проекту этого другого, но провал таки случился, виновного в провале наказали). Как к этому относиться? А если бы этот один работник не дал реализоваться провальному проекту — как бы этот другой к нему относился? Ещё пример: работники делают заведомо неконкурентоспособную систему, все об этом знают, но никто не отказывается работать. Система сделана, не работает (как и ожидалось), денег за неё не получено (зарплаты платились из кредита), компания обанкротилась. Правы ли работники? За что менеджеры наказали собственников этого предприятия? Или работники просто украли свою зарплату? Но они же работали! Тогда это мошенничество?!
- Человек, который сегодня рациональный/разумный человек-киборг. Тут возникает огромное число проблем, начиная от проблемы взаимоотношений полов (никакой любви преподавателей со студентами! Ибо студенты зависимы! Это несправедливая любовь, даже по согласию! И разновозрастной любви тоже не должно быть — подростки могут заниматься сексом друг с другом, но не со взрослыми!) до самообмана (часто полностью осознаваемого, вроде «захочу бросить курить/есть на ночь/ложиться спать под утро — легко брошу, просто не хочу бросать»). Да и сами вопросы: кого

---

[319] текст А.Левенчук 1999 года про контрактные юрисдикции, https://old.computerra.ru/1999/320/196008/, более свежий обзор панархизма/контрактных юрисдикций/частных юрисдикций — https://medium.com/libertarian-state/контрактные-юрисдикции-краткое-введение-8153668ed268

[320] Есть книга на эту тему: https://www.litres.ru/daniel-kaneman/shum-nesovershenstvo-chelovecheskih-suzhdeniy/, но можно сомневаться, что предлагаемые в ней меры как-то решат проблему.

считать личностью, дееспособным. Детей — начиная с какого возраста или приобретённых умений? Инвалидов — кого считать инвалидами? Проблема психиатрических больных (нет «объективных» тестов в психиатрии, это более-менее произвольная оценка), да и просто назначений. Проблема осуждения преступников (или не преступников? В разных сообществах, в разных обществах могут быть разные ответы на этот вопрос для одного и того же поступка. Впрочем и в одном и том же обществе можно было в один год получить штраф за то, что идёшь в маске и не позволяешь тем самым тебя легко узнать, а в другой год получить штраф за то, что идёшь наоборот, не в маске — а в эпидемию маску носить как раз надо).

- Существо — тут разведение существ (типа как коров на свете, выращиваемых для забоя, много больше, чем можно было бы их ожидать в дикой природе), сохранение разных генотипов (борьба с эволюционными вымираниями!), генная инженерия («мы не можем ждать милостей от эволюции»), и тут же «благо всех чувствующих существ», включая интеллект компьютерных агентов уровня муравья: этично ли выключить такой интеллект, или это будет «убить агента»? А наступить на муравья — этично? А вымыть руки бактерицидным мылом? Это ж геноцид, то есть устранение живых/чувствующих существ по принципу различий в геноме! Ладно, а аборты: рациональности/разума-то в эмбрионах человека ещё нет! И даже в младенцах особо нет, чем тогда они отличаются особо от кошек или даже тех же коров?
- Вещество — его явно можно использовать во вред! Поэтому надо запретить работать с некоторыми веществами. Уран-235 без обсуждений, но и золото тоже, это ж неподконтрольная властям валюта, конкурирует с бумажными деньгами! И наркотики, конечно, но не алкоголь, хотя и алкоголь тоже, или всё-таки нет для алкоголя. И сталь, если она в форме пистолета, и даже ножа, или у него сомнительная цена в больших количествах, это ж «демпинг».

Конечно, это крошечная выборка примеров. По сути дела, вся социальная (личность-организация-сообщество-общество) жизнь как конкуренция инженерных проектов крутится вокруг возможных ответов на эти вопросы, попытки оптимизировать конфликты[321]. Но мы помним, что в силу многоуровневой неустроенности огромное число возможных решений имеет вполне сравнимую оптимальность, улучшения этой оптимальности (удачные решения, обычно как-то согласованные на нескольких системных уровнях) редки, разнообразие решений огромно, а «правильного» единственного решения не может быть. Плюс непрерывно растёт уровень сложности, человечество тут не должно быть последним, и вряд ли прилетят инопланетяне, чтобы усложнить ситуацию, но трансгуманисты с этим усложнением вполне могут справиться (собственно, даже наличие двух полов, или нескольких основных человеческих рас уже вызывает вопросы на тему сколько разных человечеств живут на планете).

Скажем, клетка в организме хочет безудержно размножаться. Для клетки это «хорошо и этично», а для органа и тем более организма это рак, «плохо и неэтично» — и тут вроде всё понятно, поскольку клетке и даже организму мы отказываем в разуме. Человек в проекте хочет выспаться и не ходить на работу, а для команды и дальше коллектива это провал проекта и неполучение зарплаты всеми. Но должна ли клетка подчиняться требованиям органа и дальше организма (кому должна? кто

[321] https://trv-science.ru/2022/03/vsya-nasha-zhizn-zadachi-po-optimizacii/

определяет «долженствование»? как организм решил, что эта его гипотеза, что вот так делать надо обязательно реализуется? Можно ли вообще говорить таким языком «долженствования», если речь идёт об отношениях не слишком разумных сущностей?), и должен ли аналогично человек подчиняться требованиям команды и дальше коллектива (вот тут всё то же самое, но слово «должен» вдруг приобретает совершенно определённый смысл)? А если неразумные клетки или разумные люди не выбирали присоединиться к организму или коллективу, если их не спрашивали об этом — могут ли они начать священную борьбу с неблагоприятным окружением, этично ли это? А если коллектив (отказываем ему в разуме! но тоже проблема, ибо вычислители вполне нормально объединяются для увеличения вычислительной мощности) борется с другими коллективами, а люди в них перебегают из одного такого коллектива в другой, или сплошь «двойные агенты»? Чью сторону возьмём в споре? А если эту борьбу обозвать патриотизмом, а перебежчиков со всех сторон обозвать «предателями» — поменяется ли оценка событий от приклеивания эмоционально окрашенных ярлыков, взятых из каких-то неявных этических теорий?

Неустроенности из-за межуровневых конфликтов неизбежны, эволюция при этом рассматривается как многоуровневая оптимизация[322]: неустроенности всегда будут, но они при наличии развитой этики не приведут к смерти всего стека индивидуальных систем разных уровней — жизнь при наличии этики не прекратится, а при наличии развитой этики оптимизация будет высокой, создаваемые системы будут всё более и более успешны, жизнь будет процветать.

Этические задачи из-за межуровневых социальных конфликтов будут всегда, недаром в конфликтологии замечают, что в информационной войне выигрывает тот, кто берёт цели максимально высокого уровня, если он сможет показать это в пропаганде: тот, кто защищает семью в квартале проиграет в информационной войне защищающему интересы квартала в целом, кто защищает мир во всём мире/цивилизацию выиграет у того, кто защищает интересы одной страны — но горе, если будет показано, что вроде как борец за счастье всего мира прежде всего борется за счастье себя и своих приближённых в ущерб этому самому миру. При этом помним, что системных уровней много — и можно играть на неустроенностях (квазиоптимальных конфигурациях систем, переходы между которыми крайне легки, geometrical frustrations[323]) «через уровень» («да, я борюсь против всех моих ужасных соседей по кварталу, ибо я воюю за интересы города в целом, за объединение всех кварталов города, а мои соседи по моему кварталу этому мешают!»), что окончательно всё запутывает.

Этика тем самым — это метод многоуровневой оптимизации для разумных (и даже «разумность» тут обсуждаема) системных уровней: чтобы неразумные квазиорганизмы не поубивали свои разумные квазиклетки, а разумные квазиклетки не поубивали свои квазиорганизмы (с учётом того, что всё это происходит не на двух системных уровнях, а на множестве и в разумности коллективам, сообществам, обществам и даже цивилизации в целом можно отказывать, с чем отнюдь не все будут согласны).

Экономисты любят приводить примеры с продавливанием законодательства, где кто выигрывает 100 тыс. рублей от какого-то решения, может продавить его, если от

[322] https://www.pnas.org/doi/10.1073/pnas.1807890115
[323] https://en.wikipedia.org/wiki/Geometrical_frustration

100 тыс. человек убудет всего по одному рублю от каждого. Это всё происходит на разных системных уровнях: кому-то «избранному» на более высоком системном уровне (скажем, чиновнику, продавливающему какие-то политические интересы) становится существенно лучше, когда от каждого гражданина будет взято чуть больше налогов — с сотни миллионов налогоплательщиков взятый дополнительно лишь один рубль или доллар даст сотню миллионов долларов на уровне бюджета, и даже если просто из этих денег заплатить чиновникам зарплату, то это существенное перераспределение средств — вроде как небольшое бремя для каждого на уровень ниже (отдельных граждан), но выгоды на уровне выше («органы управления» чем-то «большим» — обществом, например) огромны.

Бывает и наоборот: за счёт того, что более высокому уровню (многим!) становится чуть хуже — клетки печени, скажем, растут и радуются в опухоли, и ещё дают метастазы, а все остальные самые разные клетки организма получают проблемы, организму в целом от этого плохо. В коллективных системах такое тоже может быть: скажем, резко начинает расти число солдат, полицейских и прочих силовиков (которых очень много разных видов) которые просто обирают более мирное население — обществу в целом от этого плохо, солдатам вполне хорошо.

Ещё один этический вопрос — это введение центрального планирующего органа на уровне общества, принуждающего к перераспределению богатства или убытков: социализм — это же как раз «госплан», плановое хозяйство. Ведёт это к игнорированию сигналов техно-эволюции, стагнации в развитии, очередям за самым необходимым. С другой стороны, никакого расслоения на бедных и богатых: все одинаково бедны, даже самое большое начальство живёт в бедном обществе относительно бедно, им неоткуда взять роскошь, её просто нет! В советское время показывали бедную жизнь пролетариата Англии (бастующих шахтёров) — но главное, что замечали люди у этих бедных шахтёров, так это цветные телевизоры, которых не было в СССР в то время даже у относительно богатой «номенклатуры». Равно как было очень мало автомашин в личном владении. А потом социализм рухнул — и автомашина перестала быть роскошью, стала средством передвижения. И так практически везде в мире. Но что этичней: распределение равной бедности, или появление кроме бедных людей ещё и богатых людей?

Может быть принудительное перераспределение даже «духовных благ». Во всём мире «свобода совести» относительна: некоторым «совестям», то есть религиям, явно посвободней, чем другим, причём это будет называться «исторической справедливостью». Этично ли это? С точки зрения титульной религии — вполне этично, больше душ именно этой религии «спасётся», хотя и не в этой жизни. С точки зрения других религий это прямое нарушение «свободы совести», с точки зрения атеистов — это прямой вред, преимущество верящих в особый вариант дедов морозов в массовых масштабах, уважение к детским заблуждениям.

В межстрановых (уровень общества) событиях (миграция, войны, существование «исторической родины» для диаспоры, внешние займы, торговые войны) распространено мнение, что многим непричастным к организации этих событий людям этично или страдать, или нести ответственность. Если ты случайно оказался на «оккупированной территории», будь готов, что тебя будут не поддерживать в твоей оккупации, а наоборот — будут тебя лишать всех возможных благ, чтобы ты своим существованием не усиливал оккупантов (а, например, организовал восстание, или уехал куда-нибудь из оккупированной местности)! Этично ли «способствовать освобождению от оккупации», усиливая беды населения от этой

оккупации?

Если ты не участвовал в каком-то событии (например, не был на фронте в какой-то войне), но попал в общество, где какие-то соседи таки участвовали, то тебя будут дискриминировать по принципу «коллективной ответственности». Если один рыжий или гражданин рыжей страны кого-то убил, то будут ограничивать в правах всех рыжих или граждан какой-то страны (большие группы людей, выделенные по принципу гражданства, вероисповедания, пола, цвету кожи или волос, любым другим произвольным признакам) — принцип «коллективной ответственности». Если убило кого-то предыдущее поколение, совсем другие люди — ограничивать в правах будут даже незнакомых с предыдущим поколением людей, ибо «коллективная ответственность». Этичен ли принцип «коллективной ответственности»? А этично ли прямо отказываться от «коллективной ответственности» и наоборот, бороться с назначением тебя «коллективно ответственным»?

Этично ли проводить опрос о целостности страны/сепаратизме/«освободительном движении»/сецессии[324] только среди жителей, которые не живут в спорной местности? Или только среди жителей всей страны вместе со спорной местностью, их обычно вдесятеро больше живёт не в спорной местности, результаты примерно те же будут, что в предыдущем случае? Или только среди жителей самой спорной местности, но на них предъявляют права те, кто в этой местности не живёт! Что делать с их интересами? Там же кипят страсти! И как доказать, что опрос был не под дулом пистолета? Не фальсифицирован? Пример фальсификации опроса тут выборы президента США: вроде как развитая демократическая страна, а фальсификация вторых выборов Трампа вдруг оказалась важной плохо решаемой уже много лет проблемой.

А что при вопросах о целостности страны с участием «страдающего меньшинства» — его страдания куда деть, если большинство проголосовало за какой-то один вариант, а меньшинство проигрывает? А ежели результаты 50-50, «примерно поровну», и есть понимание, что это «поровну» не случайно[325]? Системное мышление про многомасштабность/многоуровневость и этика существенно связаны. Талеб приводил в «Антихрупкости»[326] пример ренормализации, чтобы показать, как активное меньшинство подчиняет себе более-менее пассивное большинство. Wokeism как идеология[327] основан на тех же механизмах, поэтому «власть большинства» оказывается не всегда работающей. А этично что: уважение прав меньшинств (в том числе компенсация ныне живущим представителям за беды предыдущих поколений — есть же и «коллективная ответственность» и симметричная «коллективная выгода») или уважение прав большинства? Или как в инженерии — вопросы голосованием не решаются, а нужны содержательные обоснования?

Является ли голосование нормальным способом решения инженерных содержательных проблем, проблем изменения состояния мира с одного на другое? Этично ли решение проблем именно голосованием, «демократия»? Карл Поппер

---

[324] https://ru.wikipedia.org/wiki/Сецессия

[325] Карл Фристон: по важным вопросам разделение всегда будет 50-50, дело в самих обсуждениях/коммуникациях, в «Теория демократии и "всё новое приходит сбоку"», https://ailev.livejournal.com/1616855.html

[326] https://www.litres.ru/nassim-nikolas-taleb/antihrupkost-kak-izvlech-vygodu-iz-haosa/

[327] https://ivanov-petrov.livejournal.com/2287616.html

считал, что демократия — это даже не механизм бескровной смены власти, а механизм исправления ошибок в выборе власти. Но кто считает действия власти ошибкой? Большинство ведь отличненько может заблуждаться! Этично ли вести решение проблем голосованием? Ход на репликацию безошибочных мемов в ходе эволюции вроде намечает какую-то линию рассуждений на эту тему, но аргументация по этой линии не выглядит убедительной.

Дело даже не в том, что если голосованием решать вопрос о толщине металла для корпуса железнодорожной цистерны не рекомендуется, а рекомендуется нормальная работа инженерной команды, то почему в социальной инженерии вдруг должно помочь голосование. Вопрос в том, что в голосовании не может быть нормально решён вопрос представительства, нет математически корректного решения задачи голосования в парламенте, **волю «народа» голосованием точно отразить нельзя в принципе** (это обсуждает Дэвид Дойч в главе «13. Альтернативы» книги «Начало бесконечности. Объяснения, которые меняют мир»).

И это вроде вопросы системного мышления про отношения «часть-целое» на уровне территорий и находящихся на них людей с их имуществом, и **всегда нужно поэтому смотреть на уровень ниже «двух сторон международного конфликта» (их там много больше) и уровень выше — что там в мире в целом, где такого происходит много и сложилось определённое мнение, хорошо это, или плохо (например, когда и как границы менять хорошо, когда и как плохо — когда это «восстановленная справедливость», а когда «границы должны быть незыблемы»).**

Вопросы применения силы для продавливания на более и более высоких уровнях (драка двоих, драка стенка на стенку между «проектами», набеги сообществ друг на друга, войны и военные спецоперации, санкции торговые и военные как-то организованных обществ, включая, впрочем, и вполне не организованную, а спонтанную партизанскую войну) в защиту тех ответов на задаваемые тут вопросы, которые стороны считают этически для них обоснованными, мы даже тут пока не рассматриваем. Вроде у цивилизации/человечества SoTA в том, что применение силы недопустимо и даже если дошло до войны, то и она должна идти «по правилам», и «кто начал первым/превентивно, тот и виноват», но недаром министерство нападения любой страны называет себя исключительно «министерством обороны», с меметической инженерией политтехнологи во всех странах разобрались, и пропаганда в современном мире не хуже геббельсовской и сталинской, а много, много лучше, эффективней и использует современные достижения когнитивной нейронауки. В СМИ и соцсетях все вроде за высшую справедливость, а в реальности мы живём среди не слишком этично ведущих себя людей (и уже не слишком этично ведущих себя AI-агентов, и не слишком этично ведущих себя организаций — причём эта неэтичность организаций, включая правительства, может быть эмерджентна из вполне этичного поведения сотрудников и служащих). И при этом что такое «этично себя ведущие» на каждом системном уровне узнать тяжело, ибо придётся решать задачу многоуровневой оптимизации, которая заведомо не имеет абсолютного решения, только субоптимальные.

В этих бездушно-инженерных рассмотрениях (линия мультимасштабной социальной инженерии как изменения систем сразу на многих системных уровнях, то есть изменение и людей, и организаций, и сообществ, и обществ, и даже попытки изменить человечество) этика всплывает на каждом шагу, начиная с вопроса

этичности принудительного («органы управления») изменения решениями одного какого-то «уполномоченного агента» жизни множества других агентов. Анархисты[328] считают это в принципе неэтичным, у минархистов[329] — этична только крошечная часть, у этатистов[330] — это вполне этично.

Скажем, работа «социальных аналитиков», которые отказываются признавать себя социальными инженерами. Так можно рассматривать работу консультантов, и не только по части «аналитики для проектов создания успешных киберфизических систем», но и более высоких системных уровней стека создателей — сообществ, обществ и человечества/агентечества в целом. Автор нашего руководства когда-то спросил Игоря Минтусова[331]: вот тебя Гитлер попросил помочь сделать газенваген поэффективней — какие ты дашь советы? Он отвечал, что будет как врач, не отказывающий в лечении: даст самый лучший совет, как сделать поэффективней, но ответственность за использование совета лежит не на нём, а на Гитлере. Идея о том, что консультант по норме (а где эта норма прописана? Откуда она вообще известна?) предпринимает усилия по тому, чтобы его советы таки были реализованы (если не реализованы, то зачем советует?!), или что клиент может оказываться лишь усилителем для претворения его советов в жизнь (при наличии доверия к советам) ему в голову намеренно не приходила и отвергалась. Ибо он же был главным образом «консультант по победе в выборных кампаниях разных уровней власти», помогать сволочам, которым выборные органы нужны были исключительно для доступа к бюджетам, входило в его обязанности, но он очень хотел, чтобы совесть его не беспокоила. Вот отсюда и трактовки: себя как врача, не имеющего права отказать злодею (но ведь это не вопрос жизни-смерти злодея как человека-живого-существа!), и себя как «чисто обработчика информации», который не несёт ответственности за изменения мира.

Конечно, «аналитик» как инженер несёт ответственность за то, что «насоветовал». Нет анализа без последующего синтеза! Роли аналитиков берут на себя агенты в замаскированных ролях инженеров. Эти агенты просто не хотят нести ответственности и брать на себя принятие решений. Выбор быть «аналитиком», а не «инженером» — это в том числе и вопрос этики в их отношении к делу!

Но идеи социального (уровня сообщества и выше) устройства даже инженеров всегда рисковые, «изобретения-гипотезы» с непонятными часто следствиями, и гарантий успешности предлагаемых инженерных решений дать никто не может.

В бизнесе есть поблажки как «ограниченная суммой вложений ответственность», иначе в собственники бизнесов вообще никто бы не шёл, а в социальных проектах госчиновник или политик не могут разориться от своих решений, не могут понести ответственность за свои плохие решения — разве их могут поругать, «поставить на вид».

Консультант как инженер должен предпринимать меры, чтобы его инженерные/содержательные решения в виде «советов» таки вели к изменениям в окружающем мире, иначе зачем вообще советовать?! И ещё нужно учитывать, что работа эта обычно коллективна, и тут «роль инженера в проекте» примерно так же может обсуждаться, как «роль личности в истории» (да, от одного инженера многое

---

[328] https://ru.wikipedia.org/wiki/Анархизм
[329] https://ru.wikipedia.org/wiki/Минархизм
[330] https://ru.wikipedia.org/wiki/Этатизм
[331] https://en.wikipedia.org/wiki/Igor_Mintusov

зависит, но отнюдь не всё).

Ещё стоит вопрос с самой этикой: откуда она пришла в таком виде? Она меняется со временем? Да, меняется. С этикой всё то же самое, что и с любой наукой — человеческое знание про этику растёт. Универсальной модели/объяснительной теории мира (включая модель/объяснительную теорию этики) нет, есть много конкурирующих неуниверсальных. А ещё есть вечная невязка/несоответствие этих моделей/объяснений с актуально происходящим в реальности. С этикой всё то же самое: к ней нужно относиться как к физике, постоянно развивающейся теории и регулярной смене парадигм (основополагающих объяснений, радикально новых догадок, как это всё нужно описывать).

Опять же, этика тут нормативная наука, которая задаёт некоторую цель, но эта цель тоже предмет обсуждения, как в любой науке[332].

Этика вполне развивающийся метод, а в составе метода активно развивающаяся теория/дисциплина. И тесно пересекающаяся с системным мышлением через вот эти самые конфликты между уровнями и неустроенности из-за них. И при системных рассуждениях приёмы этического мышления выкидывать из фундаментального мышления в целом нельзя ровно так же, как нельзя выкидывать рациональность/логику (тоже имеющую статус нормативного метода с нормативной теорией, то есть предписывающей «как надо», а не описывающей «как вообще бывает»[333]). Подробней об этике рассказывается в руководстве по интеллект-стеку, там этике посвящён целый раздел. Этот раздел находится как раз перед риторикой, ибо риторика (убеждающая речь) должна быть этичной, убеждать надо только в том, что будет успешным — нельзя убеждать в том, что заведомо неуспешно, не отвечает известному на сегодня минимуму конфликтов между системными уровнями (максимальная, но недостижимая успешность: когда все вокруг довольны!).

Следование только логике/рациональности без этики, или только этике без какого-то сценирования будущего, или только системному мышлению без обращения к прикладным методам, или только быть собранным и осознанным — делать что-то это одно в проектах по созданию и развитию систем не поможет в части их успешности, максимизации удовлетворения интересов всех затрагиваемых проектом ролей.

Поможет только задействование всего интеллекта по всем методам мышления интеллект-стека, плюс ещё и владение прикладными методами работы в разных предметных областях, хотя бы на уровне кругозора, а в предметной области вашего проекта обязательно не на уровне кругозора, а на уровне профи.

Если у вас есть этические проблемы в вашем конкретном проекте, то вам с этикой придётся разбираться на уровне более глубоком, чем кругозорный. И лучше бы вы разобрались. Если этических проблем нет, то они могут в любой момент возникнуть, или они таки есть, просто вы их не замечаете. И поэтому лучше бы вам разбираться с этикой на уровне

---

[332] https://plato.stanford.edu/entries/metaethics/ и там дальше https://plato.stanford.edu/entries/constructivism-metaethics/, и в целом про возникновении знания по морали https://plato.stanford.edu/entries/moral-epistemology/, но ещё это развитие этики как объяснительной теории (хотя и нормативная, но всё-таки это объяснительная теория, объясняющая причины и следствия принятия этических решений) у Дойча популярно и по-русски раскрывается в «Начале бесконечности».

[333] https://plato.stanford.edu/entries/logic-normative/

более глубоком, чем «по верхам, из постов друзей в социальных сетях». Впрочем, это верно для всего интеллект-стека, не только верно для одной этики. Образование лучше иметь хорошее, чем плохое. Образование — это как раз изучение универсальных/фундаментальных/трансдисциплинарных по отношению к прикладным областям методов мышления, а по отношению к прикладным областям мы говорим другое слово — «обучение».

Скажем, вы привыкли считать, что организация или сообщество — это «такой человечек, лицо». И дать организации распуститься или дать сообществу распасться — это как убить человека, и даже больше одного человека, «жизнь организации» приравнивается к жизни многих людей. Чтобы какая-то организация или даже общество не распалось, некоторые люди готовы принести в жертву несколько человек, но часто не готовы принести в жертву какую-то такую «общность людей», чтобы предотвратить убийство нескольких человек.

Уподобление людей и объектов социальных системных уровней в принципе неверно. Организация, сообщество, общество созданы из людей-индивидов, которые при распаде одних своих коллективных форм социальной организации просто перейдут в другие — другие организации/предприятия, другие сообщества, другие общества. Сохранять организации (и проекты в них), сообщества, общества «любой ценой, даже ценой смерти каких-то людей» не нужно.

При исчезновении организации, сообщества, общества как целости из входящих в них людей, сами люди не умрут и быстро встроятся в другие организации (и проекты в них), сообщества, общества. Кому-то будет в новых организациях, сообществах, обществах чуть лучше, чем в прежних, кому-то чуть хуже, но в среднем всё будет так же субоптимально — определяться общим знанием цивилизации о том, как противостоять природным силам. Но вот если распадётся на органы человек, то эти органы уже не встроятся в других людей, не найдут себе место, это чистые эволюционные потери. И вот тут уже нужно быть жёстким: минимизировать людские потери, они невосполнимы.

Государственное строительство и госпроекты

Если ребёнку в руки попадает молоток, то все предметы в доме превращаются в гвозди — и это означает, что не ребёнок владеет инструментом, а инструмент владеет ребёнком. Если агенты, выполняющие роли инженера киберфизических систем, учителя (инженер личности), менеджера (инженер организации), политика (инженера общества) получают возможность получить бюджетные деньги, то они непременно этим воспользуются, ибо это чрезвычайно прибыльно: по факту деньги ведь на проект создания какой-то кому-то нужной или даже не нужной, или даже для многих вредной системы собирают со многих людей под угрозой наказания за неуплату налогов, а отдают одному этому агенту (индивидуальному или коллективному, неважно). Почему бы и не заняться выгодным проектом?

А ещё есть проекты по строительству государства, поддержанию той или иной организации власти и сопутствующего власти хозяйствования (то есть силовой отбор налогов и дальше их перераспределение на различные цели).

Если агент-госстроитель (надо чётко понимать, что тем самым госстроитель поддерживает какую-то политику/стратегию для государства, так что он

политик::роль) знакомится с системным подходом, системной инженерией, инженерией личности, менеджментом, то он непременно захочет их использовать, чтобы поднять вероятность достижения своих интересов в ущерб всем интересам политиков-конкурентов (помним, что политики борются друг с другом за воплощение каких-то разделяемых ими идей об устройстве общества, конкурируют за ресурсы).

Системную инженерию в её классическом виде «железной» и «программной» инженерии, инженерии киберфизической системы для целей госстроительства использовать нельзя, она предназначена прежде всего для проектов создания и/или модернизации киберфизико-человеческих систем небольших масштабов (аэрокосмические проекты по сравнению с госсистемами — это очень небольшой масштаб). Метафора часовщика (проектирование — изготовление деталей — сборка) неприменима, тут система систем, её надо как-то выращивать/обучать. Хотя можно «обучить» и очень быстро: сказать, например, что «заплати налоги — и спи спокойно» и пригрозить тюрьмой за несоблюдение. Научатся все, и быстро!

В проектах классической системной инженерии даже для больших киберфизических систем (авиалайнер, подводная лодка, ракета) чётко определены проектные роли по отношению к «традиционным» аппаратным или даже программно-аппаратным системам. Социальной инженерией этими методами заниматься нельзя, они не работают с сообществами и обществами — а это основные системы, которыми занимаются в государстве (как бы ни определялось значение этого «государства»: как страна; как все чиновники страны, но без граждан; как все граждане страны вместе с чиновниками; как объект международного права и т.д.). Методы системного менеджмента как инженерии предприятий тоже будут работать плохо: они хорошо будут работать там, где понятны отношения распоряжения трудом и капиталом, где начальники дают распоряжения подчинённым. В госстроительстве это означает, что менеджмент внутри какой-то госорганизации наладить можно, а вот «управление гражданами» наладить по линии менеджмента нельзя, то есть можно делать только «госстроительство-без-граждан», что может быть не просто бессмысленно, но и вредно. Это сразу отсылка к этике: могут ли одни люди распоряжаться другими людьми, если нет какого-нибудь добровольного явно подписанного двумя сторонами контракта, а не «подразумеваемого общественного договора» или ещё какой удобной фикции, принуждающей свободных людей подчиняться каким-то другим организованным в госорганы людям просто на основании того, что они родились где-то и когда-то, причём ничего вроде контракта не подписывали.

В госстроительстве имеют дело главным образом с большими системами из заранее неопределённого числа людей, которые малознакомы или вообще не знакомы создателям системы — сообществами и обществами. Обратим внимание, что создание таких инженерных проектов, как вполне программно-аппаратные системы слежки за гражданами, системы эффективного убийства (обычно называемые оружием, и при этом помним, что все в мире министерства нападения называют себя министерствами обороны), систем принуждения к уплате налогов и т.д. — это ведь впрямую связанные с госстроительством проекты, и они не столько про создание и развитие «железа» и «софта», сколько про создание и развитие средств непосредственного (чаще всего силового) воздействия на людей, это сходу вопросы этики::метода принятия решений, а по методу этому должна работать совесть::роль. Как и в случае любого другого метода, роль действует не по наитию,

а по какому-то методу (даже если нет понимания, как этот метод называется — это не значит, что метода нет). Совесть задействует метод/способ/культуру рассуждений по той этике, мастерство выполнения которой известно агенту.

Часто ли мы слышим учёт интересов совести::роль в госпроектах, обсуждение передовых этических методов принятия решений по поводу возможности того или иного действия? С учётом того, что в сверхбольших системах, в которых взаимодействует очень много людей, хуже можно предсказывать последствия тех или иных решений — часто ли это обсуждается в явном виде в ходе госстроительства?

Создание эффективных газенвагенов в фашистской Германии — это задача была чисто инженерная, «ничего личного, это бизнес», или всё-таки связанная с людьми? Понятно, что все эти проекты сразу вызывают этические вопросы, ибо опираются на силовые (армия, полиция, инспекции без возможности от этих инспекций отказаться) способы их реализации. Но внутри исполнителей это «просто деятельность», вопросы этики даже в голову не приходят[334].

Кто такие создатели систем из людей, почему они считают возможным указывать другим людям, как им жить (или даже принуждают не жить, как в случае газенвагенов), используя при этом насилие государственной машины — диктуя свои правила и угрожая насилием за их неисполнение?

Краткий тут совет — если уж нужно заняться госстроительством, то используйте знания по этике, политологии, конфликтологии, праву, экономике, социологии и системное мышление в его «мягких» (soft systems) вариантах, но не используйте системное мышление в вариантах системной инженерии киберфизических систем, системного менеджмента как инженерии предприятия, как вы привыкли его использовать для мышления о системах на уровнях инертного вещества, существа, личности, организации/фирмы. Эти проекты обычно даже не уровня небольшого сообщества, а уровня общества, когда вы лично не знаете людей, которых это затрагивает (включая тех людей, у кого на это дело отняли налоги и предложили потом спать спокойно).

Государство и люди в нём — это не отсеки подводной лодки, это не детали медицинской аппаратуры, это даже не атомная электростанция вместе с её персоналом. Лётчик вряд ли кого сможет убить при личной встрече, совесть не позволит. А вот сбросить атомную бомбу на город — запросто, город ведь далеко, смертей не видно. **Люди, занимающиеся работой с государством, должны помнить, что им придётся играть роли, которые будут уменьшать свободы граждан — государство не даёт свободы, оно их главным образом ограничивает. Это этический выбор людей, помогающих государству.**

У проекта по созданию и развитию систем всегда есть вполне определённые внешние проектные роли, которые платят за этот проект: заказчики проекта (их может быть много разных!). Кто заказчик в госстроительстве? Политики? Чиновники? «Народ» (например, описанный опросом общественного мнения или результатами фокус-группы)? «Элита» (и кто её определяет)? Группы экспертов (вариант «экспертократии» — но как выбрать из этих многочисленных групп экспертов «правильную», все эксперты ведь говорят разное и часто прямо противоположное, и очень хорошо это противоположное обосновывают)? У кого

[334] https://ru.wikipedia.org/wiki/Банальность_зла:_Эйхман_в_Иерусалиме

вообще спрашивать: путаница ведь страшная. Эпистемологи смеялись во время пандемии, что их путают с эпидемиологами — журналисты, а иногда и чиновники просили комментарии по поводу происходящего. Но это ведь общая ситуация, при любой встрече с неизвестным нужно найти чьё-то мнение. Например, можно спросить знаменитость — и вот уже по вопросам политики выступают знаменитые актёры и журналисты (а не знаменитые социологи).

Нужно чётко понимать, что **в случае госстроительства и госзаказов речь идёт о политике**, а не о классических проектных ролях при создании систем. Не нужно себя обманывать, говоря, что «есть заказ, оплачивается он из бюджета, следовательно исполнителем роли заказчика является тот чиновник, который будет подписывать мне акт приёмки работ». Чиновники просто узурпировали право формулировать заказ и тратить деньги от имени «народа», и их действия могут быть прямо противоположны тому, что считали бы приемлемым люди в том же «народе». Это как если бы в пьесу «Гамлет» Шекспира пришёл актёр из театра Петрушки и изобразил вместо Гамлета свою любимую роль Петрушки, ибо «меня назначили играть Гамлета, вот и играю как знаю».

Агенты-чиновники, например, могут реализовывать свои личные запросы по обогащению, и только изображать выполнение ролей представителей «народа» в заказах, а не играть роль представителей «народа»: их и их семьи может волновать «откат», проект интересовать как удобный для «распила», а результаты для собственно госстроительства могут не волновать вообще, они не будут выполнять проектных ролей заказчика госпроекта, у них совсем другие роли — и эти роли в их личных и семейных проектах, или даже личных проектах обогащения их начальников (а там дальше будет личное оправдание «я только исполнял приказ» и «всё равно разворуют, так пусть лучше это буду я, чем кто другой»).

Но то, что чиновники могут не играть роль «народа», а преследовать личный шкурный интерес, это не главное. Важно, что **принципиально нельзя сыграть роль представителя «народа»**! Госстроительство — это прежде всего политика, а что для одного человека в роли политика успех, для другого будет полным провалом ввиду других идеологических установок. Кто-то считает, что аборты — это плохо по этическим соображениям, а кто-то по тем же соображениям считает, что аборты — это спасение для молодёжи. Кто-то считает, что бесплатное (то есть за счёт налогоплательщиков) образование должно быть и начальное, и среднее, и высшее, а кто-то считает, что образование должно оплачиваться из кармана и не регулироваться в части содержания — иначе не будет никакого разнообразия в образовании, а это плохо для развития человечества.

Таких различий в интересах (предпочтениях для важных характеристик систем на каких-то системных уровнях) столько, сколько в «народе» разных политических групп. Поэтому государственные проекты будут неуспешными по определению: в них никогда не будут учтены предпочтения всех внешних ролей, и даже эти предпочтения нельзя будет выявить честно и справедливо. Напомним, что по определению системной инженерии успешная система — это которая учитывает предпочтения в значениях важных характеристик/интересы для всех своих внешних проектных ролей.

Если вы считаете, что что-то в политике честно и справедливо, будьте готовы, что другие люди вас обвинят в полной некомпетентности, нечестности и несправедливости — они честным и справедливым будут считать что-то полностью

противоположное. Отношение к абортам, мигрантам, эвтаназии, прогрессивности налогообложения, поддержке тех или иных отраслей хозяйства, акцизам — нет таких вопросов, по которым не были бы прямо противоположные мнения.

Обвинения в неэтичности, некомпетентности, нечестности, несправедливости будут всегда взаимны. С чего вы взяли, что правы вы, а не те другие? Почему реализацией идей из госпроекта должны заниматься вы, реализуя какие-то одни цели, а не те другие, которые хотят реализовать что-то прямо противоположное? Из общих соображений «те другие» ведь получили такое же образование, в целом не глупей, и явно не преступники. Почему деньгами налогоплательщиков должны распорядиться вы, или ваши чиновники-заказчики, а не политические ваши и ваших чиновников противники? Тем более что противников много разных, в политике ведь существует много вариантов для каждого возможного проектного решения для устройства общества, там явно не два варианта «правильный» и «неправильный».

Конечно, формально для госпроекта может быть и подписанный какими-то чиновниками (людьми в ролях чиновников) «паспорт проекта», «коллективно рассмотренное техзадание», и даже может быть отчёт о его «успешной» реализации. Но это будет «успех» с точки зрения именно этих подписавших документы чиновников, но никаких других внешних проектных ролей. Одни чиновники будут что-то подписывать в бумагах по паспорту проекта как «успех», а другие политики будут делать пикеты ровно на эту же тему, писать статьи в газетах про явный вред этого же проекта и разными другими способами пытаться выразить своё недовольство — в системной инженерии и системном менеджменте игнорировать все эти роли считается не лучшим методом/практикой/культурой/стилем, такой проект нельзя будет считать успешным.

Когда у кого-то берут деньги в виде налогов, а потом тратят на чей-то проект, с самим фактом наличия которого налогоплательщик не согласен — это оно и есть, неучёт мнения главнейшей проектной роли, владельца своих денег, а хоть и уже отобранных в бюджет. Отобранных, кстати, силой — никто не отдаёт налоги добровольно, отдают только под угрозой тюрьмы, так что неуспешность начинается с этой точки, нельзя сказать, что «деньги не пахнут» — нет, бюджетные деньги пахнут, их не заплатили за услуги, а отобрали, чтобы потом заплатить.

Главная проектная роль в госстроительстве, в госзаказах, в связанных с государственными органами проектах — это «народ», поэтому желающих что-то сказать от имени «народа» в строчке бюджета и в паспорте проекта, а также от имени «народа» расписаться в отчёте очень много. Для этого политиками и чиновниками много говорится о субсидиарности, партисипативности, сдержках и противовесах, но факт остаётся фактом: по гамбургскому счёту успешных проектов в государстве нет.

Мнение об успешности всегда является частным мнением какой-то группы людей, имеющих какие-то вполне определённые ролевые предпочтения. Другие люди, у которых ролевые интересы другие, будут сильно недовольны, и такие другие люди всегда есть в «народе». Но поскольку на стороне властей сила полиции и деньги налогов этих же недовольных (на эти деньги и полиция тоже оплачивается), они ничего сделать не смогут. **Это всё политика, это не системный менеджмент, не проектное управление, не системная инженерия технической системы, не медицина или образование.**

Строить государство и организовывать госпроекты вы будете не из своего бездушного материала, а всегда из чужого с собственными их мозгами: из других людей. Люди — это не железо, и не компьютеры. Не считайте, что именно вы из них что-то построите успешное/удачное/хорошее для вас или этих других людей. Не считайте, что вы как системный мыслитель (системный менеджер, или инженер, или даже просто программист, или сапожник, или деятель культуры) квалифицированы что-то строить из людей, реализуя какие-то идеи госстроительства. Эти люди как самопринадлежные системы, из которых вы будете пытаться строить системы систем, они не ваши материалы для строительства, и они не материалы ваших заказчиков-чиновников, заказчиков-политиков. Они самопринадлежны, они все свои собственные. И они так же точно могут хотеть что-то построить из вас — в том числе и то, что вам не понравится. Золотое правило работает и тут: не делайте с людьми того, чего не хотели бы, чтобы люди делали с вами.

Инженер по безопасности может защищать систему от врагов (антиклиентов). Инженер-госстроитель не может быть инженером по безопасности, разве что он строит тюрьму. Разработчики государственного регулирования все строят тюрьму, им платят именно за это, никто никогда не платит за дерегулирование, снятие ограничений свободной деятельности[335]. Задумайтесь над этим перед тем, как построить очередной блок государства, который дальше получит властные полномочия и употребит их для того, чтобы разрастись и получить ещё больше власти. Если заказ на строительство газенвагенов выгодный, то это не значит, что нужно брать этот заказ. И нет, системное мышление не может «с этикой помягче», этика присутствует во всех рассуждениях системного мышления.

Нельзя также считать, что можно получить помощь от государства в развитии системного мышления, системного менеджмента, системной инженерии, системной биологии и других системных методов (часто об этом говорят, как о «промышленной политике», субсидирование работы по каким-то методам, чтобы увеличилось число агентов, которые имеют какую-то степень мастерства в этих методах). Чиновники ничего не понимают в этих сферах деятельности. Даже если они выделят деньги на развитие каких-либо методов из этих сфер, или создание руководств и учебных программ — вовсе не факт, что это будут конкурентоспособные методы, конкурентоспособные учебные программы. Это будут какие-нибудь (не лучшие! Именно «какие-нибудь») методы и учебные программы, которые были выбраны людьми, конкурентноспособными в получении господрядов, а не конкурентоспособными в предоставлении услуг на SoTA уровне.

Пусть проблемы инженеров и менеджеров решает не «промышленная политика», а рынок, где каждый рискует своими деньгами, а не чужими, не деньгами налогоплательщиков. Дело чиновников — не вмешиваться в решения рынка, не поддерживать рыночно слабых, спокойно давать им разоряться, не тормозить рыночно сильных и давать им заработать, и **не путать рыночно сильных и слабых с административно сильными и слабыми — то есть не путать умеющих расположить к себе инвесторов и клиентов участников рынка с**

---

[335] Формально у госорганов есть многочисленные инициативы по дерегулированию, например, деятельность по сокращению перечня лицензируемых видов деятельности, реформа контроля и надзора, но это только для отвода глаз, просто упоминания в идеологических документах: результаты этой деятельности практически нулевые много лет, нет реального дерегулирования.

умеющими расположить к себе чиновников. Рынок эффективно реализует техно-эволюцию, деньги переносят сигналы о живучести тех или иных мемов/идей.

Нужно также очень осторожно относиться к примерам системной инженерии из военных проектов и проектов из других областей, которые полностью зарегулированы государством, например, проектов атомной энергетики. Поскольку в этих отраслях принят принцип оплаты из бюджета «затраты плюс» (затраты, реально понесённые в ходе проекта, плюс оговорённый небольшой процент прибыли), то только полные идиоты не будут потихоньку год от года повышать стоимость проектов и сроки их выполнения, повышая тем самым и размер прибыли. Посмотрите на гражданскую технику, её стоимость, рост технических характеристик, сроки разработки за последние двадцать лет (возьмите хоть те же смартфоны: двадцать лет назад даже сотовых телефонов толком не было, не говоря уже о смартфонах) и сравните с военной техникой — сроками и стоимостями разработки. Разница будет разительна. Поэтому нужно признавать, что в военной системной инженерии есть множество интересных методологических находок, но **слепо копировать этот опыт нельзя: вполне возможно, что вы откопируете заодно и прилично выглядящие способы повышения стоимости и удлинения разработки.** На свободном рынке фирмы с такими методологиями бы не выжили, но на военных якобы рынках действуют совсем другие закономерности. И конечно, каждый системный инженер решает для себя сам этический вопрос: хочет ли он проектировать и строить машины для убийства (именно этим занимается военная системная инженерия). В 2007 году был доклад SpaceX на конференции Международного совета по системной инженерии (INCOSE)[336], а в 2020 году на этой конференции не было SpaceX, но присутствовало много инженеров из военных и правительственных проектов. Фронтир уже не рассматривается в INCOSE, системное мышление ушло в компании, ориентированные на рынок, а не на военные и государственные применения. С государством и войной сегодня ассоциируется не прогресс и технические достижения, а застой и неэффективная трата денег на инструментарий.

Всё то же самое верно и для системного менеджмента: военный менеджмент явно не является образцом того, как должен быть устроен менеджмент в гражданском мире, хотя многие начальники и хотели бы, чтобы их подчинённые ходили строем. Поэтому аккуратней читайте книги военных начальников, не принимайте примеры из военной культуры за чистую монету, не считайте, что найденные военными приёмы обращения с людьми будут работать вне казармы.

Будущее и проактивность деятельности

Системное мышление даёт возможность понять, как думать о будущем. Будущее физично — это просто весь физический мир, только интересует его временной срез не прямо сейчас, а через некоторое время. Конечно, в будущем каждому::агент интересно что-то своё и этот интерес будет из самых разных ролей. Ещё не факт, что речь идёт о нынешних ролях, в будущем состав ролей будет другим. Например, роли «инженер по требованиям» в современной инженерии больше нет, осталась только в устаревших инженерных методах, а области интереса системных архитекторов десятилетней давности и области интереса нынешних архитекторов

[336] https://www.incose.org/events-and-news/incose-and-se-news/2007/12/17/space-systems-engineering-risk-management-symposium---registration-open

довольно существенно отличаются.

Возможное будущее определяется не «объективно», а субъективно самыми разными агентами в их взаимодействии. Нынешние агенты это будущее порождают/«воображают» (строят рендерингом, то есть добавлением множества «натуралистичных подробностей» из порождающих моделей мира, в которых можно найти только какие-то важные закономерности, без подробностей).

Мнения агентов по поводу того, что им интересно, какими бы методами работы в каких ролях они бы занялись, ещё и быстро меняются: ведь

- и мир меняется, и сами агенты меняются физически,
- а ещё меняются у агентов модели мира и модели себя, а также
- степень уверенности в точности этих моделей мира и себя,
- культура работы тоже меняется, содержание методов эволюционирует — и лучший метод на какой-то момент требует интереса к объектам, которые были неинтересны предыдущим версиям метода.

Поэтому для каждого интеллектуального агента (то есть необязательно человека) неинтересно всё будущее в целом, а интересно только то, что обычно интересно тем ролям, которые агенты играют в своей жизни, да ещё и в будущем сами эти роли меняются и интересы меняются вслед за изменением ролей и SoTA методов этих ролей.

Агентам, играющим роль домохозяек неинтересны примусы и керогазы сегодня, и ещё нужно найти людей, которые знают, что это такое — а ведь ещё лет пятьдесят назад это было обычной кухонной утварью. Автор нашего руководства отлично помнит и приготовление обеда на керогазе, и приготовление обеда на примусе — и это в ходе обычной городской жизни. «Накачать примус» — это регулярно поручалось даже детям, там ведь был встроенный насос!

Но потом в дома пришёл газ, SoTA методов нагрева пищи для домашней готовки разительно поменялись. А уж как поменялись за эти пятьдесят лет методы добычи и распределения газа!

Будущее представляется агентам как окружение для их целевых систем, которые эти агенты (например, люди) будут создавать и развивать в каких-то своих будущих проектных ролях. Окружение — это самые разные надсистемы. Какие? Это описывается будущими потребностями. Какие именно целевые системы будут люди делать в будущем? Это описывается будущими концепциями использования. Мы узнаём будущее по описаниям его целевых систем и их надсистем — по концепциям использования каких-то будущих (то есть пока не существующих, пока неизвестных нам систем) по удовлетворяемым потребностям, которых пока нет.

«Будущеведение» тем самым — проектирование систем. Проектирование — это как раз и есть «заглядывание в будущее». Концепцию использования разрабатывает (разработка сейчас, а само использование будет в будущем) подроль инженера-разработчика: инженер-проектировщик, а ещё там есть подроль инженера-технолога производства, который рассчитывает на внутреннюю систему разработки будущего, а ещё инженера-оператора, который сможет или не сможет работать, ибо будет работать в этом самом будущем, а общие характеристики системы определяет инженер-архитектор, его тоже волнует будущее, и ещё там визионер должен согласиться, что проект будет выгоден, а не пойдёт в убыток для команды разработчиков. Это не меняется ни для ближайшего (дни и месяцы), ни для среднего (год-три), ни для далёкого (от трёх лет — сегодня и на три года ничего загадать нельзя!) будущего. Инженеры всегда одним глазом смотрят на будущее, другим — на настоящее. Все проекты/design и проекты/project — они про будущее.

Деятели/практики/инженеры должны уметь строить **порождающие модели/generative models**, чтобы предвидеть детальное будущее, исходя из своих (иногда неосознаваемых) абстрактных знаний. Порождающая модель — это модель, из которой можно выполнить обратное моделированию действие: породить какие-то представления об исходном объекте моделирования, «вообразить» варианты мира, которые дали бы при моделировании модель. Если это порождающая мета-модель, то вообразить/отрендерить/породить/сгенерировать модель. Если это модель, то вообразить/отрендерить/породить/сгенерировать мир. А какие ещё модели есть? **Дискриминативные/discriminative models**, они дают нам классификацию, отвечают на вопрос «вот описание — что это за тип объекта?». Порождающие модели отвечают на запрос: «дай пример объекта, отвечающий описанию его типа». Порождающая модель по тексту (описание) сгенерирует картинку (добавит подробностей, уберёт абстракцию). А дискриминативная модель по тексту скажет тип объекта, который описан текстом (уберёт подробности, поднимет абстракцию)[337].

Когда вы попадаете в какой-то проект по созданию и развитию систем, после освоения мышления и работы по руководствам рациональной работы, системного мышления, методологии, системной инженерии, инженерии личности, системного менеджмента, вы уже не можете утверждать, что ничего не знаете об этом проекте! Вы знаете самые общие черты всех проектов — и это означает, что вы как-то сможете сориентироваться в этом новом для вас проекте. Если вы попадёте в будущее, то вы тоже увидите какие-то проекты — и сможете в них

[337] https://www.youtube.com/watch?v=hjsZSmL67Ck

сориентироваться! У вас есть знание мета-мета-модели, это означает, что вы быстро разберётесь с мета-моделью (онтологией предметной области вашего нового проекта), моделью (важным про реальные объекты) и далее самим моделируемым миром. Это неважно, происходить это будет в далёком плохо прогнозируемом будущем (например, аж через три года), или вот прямо сегодня, но аж через пару часов.

Основатели фирм/founders как инженеры предприятия (которых называют часто «предприниматели», но мы избегаем этого термина: каждый под ним понимает свой набор ролей, а некоторые вообще считают, что речь идёт о «психологическом складе агента») зарабатывают на том, что иногда удачно оказываются в нужное время в нужном месте: предугадывают будущее, занимаются удачливым (удача тут важна!) визионерством::«метод/практика работы» (определяют в роли визионера, эквивалентной Шумпетеровскому экономическому предпринимателю, а не бытовому пониманию предпринимателя, какую целевую систему будет выгодно делать, а какую невыгодно) и стратегированием (определяют, какой из методов/способов работы позволит достичь желаемых результатов). Подробней это разбирается в руководстве по системному менеджменту (и там же даются материалы по рыночной экономике, ибо в плановой экономике всё это происходит существенно по-другому).

Тем самым основатели фирм корректно в случае удачи (или некорректно в случае неудачи — и такого много, много больше, просто о неудачах не рассказывают нигде) используют воображение, порождающее корректную картину мира в будущем.

Так, Элон Маск основывал фирму Tesla, предвидя, что электромобили будут хорошо продаваться. В его модели мира он учитывал, что стоимость аккумуляторных батарей для электромобилей будет понемногу опускаться каждый год, и в какой-то момент электромобиль будет выгодней автомобиля с двигателем внутреннего сгорания, и он оказался прав — где-то в 2022 году совокупная стоимость владения электромобилем и автомобилем с двигателем внутреннего сгорания стали примерно одинаковы[338]. Дженсен Хуанг, когда основывал NVIDIA, посчитал, что на компьютерах можно будет играть в графические игры — и если это будет развиваться, то потребуются аппаратные ускорители для видео. Так и получилось.

Стратегирование как выбор коллективного метода/стратегии работы фирмы — это ведущий метод

- серийного предпринимательства (основание фирм, эти люди называют себя founders),
- корпоративного предпринимательства (спонсорство проектов в проектном управлении, этим заняты отнюдь не только менеджеры или основатели фирм как «предприниматели», но все сотрудники фирмы, особенно на высоких должностях, то есть и люди на инженерных должностях),
- творческого проактивного принятия решений в любой другой деятельности (предпринимательство как проявление инициативы, исследования и творческая пытливость, деятельное визионерство).

Стратегирование происходит в ходе согласования интересов инженеров, которые могут создать какую-то целевую систему, продвиженцев, которые могут её продать,

[338] https://www.leaseplan.com/corporate/~/media/Files/L/Leaseplan/2022-CCI-Report_final.pdf

менеджеров, которые могут организовать предприятие, фандрайзеров, которые могут привлечь на всё это инвестиции. Это всё разные роли. Слово «предприниматель» может принимать множество значений (включая совсем уж бытовое «богатый человек» или даже просто «человек с рисковым характером»), в том числе речь может идти и о выполнении каких-то из перечисленных ролей, или даже любых необходимых для будущей фирмы ролей, то есть предпринимателем тогда называют не роль, а вот прямо агента. Мы табуируем слово «предприниматель» в силу многозначности его значений. Для **шумпетеровского предпринимателя** оставим в случае целевой системы термин **визионер**, а для системы создания — **бизнесмен**. Подробней обо всём этом будет рассказано в руководстве по системному менеджменту, и помним, что в случае фирмы минимально будет множество создаваемых систем:

- целевая система со своей ценой и создания, и продажи, это предмет заботы команды инженеров::роль фирмы. Выгодность продажи целевой системы тут.
- клиентура (все клиенты) этой системы со своей ценой создания и продажи команде инженеров и бизнесмену, предмет заботы команды продвиженцев::роль (продвижение — это маркетинг, реклама, продажи). Юнит-экономика как раз тут, выгодность этого продукта.
- сама фирма со всеми её людьми и оборудованием. Её сколько-то стоит создать (инвестиции), за сколько-то можно потом продать — и вот максимизация цены продажи тут — забота бизнесмена::роль. Выгодность продажи фирмы тут.
- а ещё надо создать инвестуру, которая даст инвестиции на создание фирмы, иначе нельзя будет создать ни саму фирму, ни её целевую систему, ни клиентуру.

Всё это, конечно, увязывается между собой в ходе стратегирования — а потом в ходе выполнения стратегии как метода достижения успешности всех этих создаваемых систем. Всё это стратегирование (определение стратегии как метода достижения успешности всех создаваемых систем), а потом планирование выполнения стратегии (планирование работ с учётом ресурсов, в том числе оно может приводить к понимаю, что выполнение стратегии с текущими ресурсами невозможно и нужно корректировать стратегию) — это работа с будущим, для этого нужны порождающие/generative модели и жёсткая критика со стороны дискриминативных/discriminative моделей

В любом случае, в ходе любого проекта по созданию и развитию систем придётся разрабатывать самые разные модели, описывающие целевую систему (концепцию использования и описания поведения системы в тестах для обоснования успешности системы, концепцию системы, архитектуру и проект системы, который доводит концепцию системы до детальности, достаточной для изготовления, и т.д.), и систему создания (например, модели стратегирования — бизнес-модель, модель оргразвития, модель согласования интересов).

Системность мышления предполагает, что придётся моделировать (мыслить моделированием) сразу с учётом нескольких системных уровней и сразу с учётом множества систем на этих уровнях. Скажем, если вы замысливаете выпуск электромобилей, то придётся продумать целевую систему в её окружении (электромобиль с окружающей его сетью дорог и сетью электрозаправок, а также внутри электромобиля самую дорогую подсистему: аккумуляторные батареи) и систему создания (автомобильный завод, который во много раз сложнее

автомобиля).

Если вы просто как инженер придумали, как вместо пятнадцати отдельных деталей, которые потом соберутся в изделие сделать это изделие из одной детали, напечатанной на 3D-принтере и сэкономить тем самым на логистике, сборке и проверке качества сборки 15 исходных деталей, то это ровно то же самое предпринимательство-в-инженерии, проактивность/активная жизненная позиция, творческий акт. Инженер-проектировщик замысливает систему (концепция использования), изобретает концепцию системы, согласованную с возможностями методов изготовления (то есть участвует и инженер-технолог), визионер оценивает рыночные перспективы такого решения — будет ли это достаточно дешёво изготавливать даже не по сравнению с конкурентами, а для достаточного объёма продаж, чтобы это окупилось. Бизнесмен принимает решение, что фирма может инвестировать в создание такой системы.

Не факт, что создание какой-то системы проходит как отдельный явно оформленный проект внутри предприятия, тем более как отдельное предприятие фирма. Хотя если встанет вопрос о масштабе такой работы, то и отдельный проект может быть, а какого-то человека, который будет играть одновременно многие из этих ролей будут называть «корпоративный предприниматель», и даже отдельная фирма может быть сделана, а агент, который выполняет эту инженерную роль, будет заниматься ещё и менеджерскими задачами разворачивания с нуля фирмы, то есть будет ещё и инженером предприятия/менеджером, но в статусе основателя фирмы, «предпринимателя» в бытовом значении этого слова.

Мы не будем говорить «предприниматель», мы будем продолжать говорить про инженерные (целевой системы) и менеджерские (инженерии системы создания) роли, в том числе особые роли «шумпетеровского предпринимателя» для целевой системы (визионер) и фирмы (бизнесмен), подробней об этом говорится в руководстве по системному менеджменту.

Для любых ролей, занимающихся изменением мира (деятельностных/проактивных/практических/культурных/инженерных), нужно принимать решения о том, как именно нужно действовать, исходя из видения успешного будущего. Эти решения принимаются на основе порождённого

- из модели (детального описания)
- и/или мета-модели (предметного абстрактного описания),
- и/или даже мета-мета-модели (абстрактного трансдисциплинарного описания, на уровне общих представлений о мире)

такого реалистичного образа/видения будущего, который агент сочтёт приемлемым/выгодным для реализации, будет вкладывать в его реализацию свои (а иногда и чужие тоже: инвесторы — это тоже внешние проектные роли) ресурсы.

В ходе стратегирования вырабатываются в виде стратегии как набора моделей гипотезы об оптимальном поведении системы-создателя (стратегия — это как раз ожидаемо «оптимальный» способ/метод действий/работы, описание поведения), создающей успешную систему. Далее идёт планирование ресурсов, на тему выполнимости плана тоже делаются гипотезы.

Выработкой стратегий, порождением альтернатив в действиях, последующим принятием решений по поводу этих альтернатив занимается фундаментальный метод мышления — рациональность, на которой базируется системная инженерия:

как порождать альтернативы/опции для решений, а затем выбирать одну из этих опций для реализации. Подробней об этом будет ещё в руководствах по системной инженерии (рациональность в принятии инженерных решений) и интеллект-стеку (рациональность как один из фундаментальных методов мышления в составе интеллект-стека, роль «разум»).

Главное тут запомнить: можно как-то разделить труд в принятии таких крупных решений (например, решения о строительстве атомной станции или запуске космического корабля на Плутон), но в принципе методы рационального поведения тут для любых агентов остаются одни и те же, только мышление::вычисления становится сложнее, самого мышления по объёму вычислений становится больше, это мышление становится коллективным.

Но ровно то же самое по методу рациональное мышление::вычисления происходит, когда вы решаете предпринять проект «оторваться от работы и попить кофе». Никакого разделения труда, объём вычислений ничтожный, они занимают мало времени и отвлекают мало агентов: вы сами генерируете/порождаете альтернативы — отрываться или не отрываться от работы, пить кофе или чай или вообще пообедать, пить кофе без отрыва от работы, оторваться от работы и просто отдохнуть-полежать без кофе и т.д. Затем вы собираете информацию для принятия решения (оцениваете объём оставшейся работы, смотрите на часы и вспоминаете, когда вы пили кофе перед этим, идёте на кухню и интересуетесь наличными сортами кофе и чая и т.д.), затем принимаете решение (тут работает теория принятия решений).

Мышление об ожидаемом успехе труда/деятельности/практики/инженерии/методов/способов работы собственной и других агентов (как сотрудничающих, так и мешающих), устроено одинаково по методу: порождение гипотез по успешным методам/способам действий, выбор наиболее вероятного в плане успешности способа/метода действий в условиях неопределённости, планирование ресурсов, выполнение действия, нахождение расхождения с моделями мира, себя, затем уточнение этих моделей и выполнение следующего прохода — порождение смарт мутаций для гипотез и т.д., причём всё это происходит более-менее одновременно. Одинаковость будет для ситуаций:

- инженерии себя (как тела и как личности) как серии шагов по инвестиции себя в какую-то деятельность, обучения чему-то, тренировке тела, получения умений для занятия какими-то видами проектов,
- «корпоративного предпринимательства», когда инвестиции делаются в создание какой-то системы, нужной компании/организации (инженеры и менеджеры в больших компаниях),
- «технологического предпринимательства» (стартапов, выстраиваемых вокруг каких-то методов с их инструментарием), когда инвестиции делаются в создание систем, которые будут пользоваться рыночным успехом,
- «социального предпринимательства», когда инвестиции делаются в создание систем уровня сообщества, общества и даже человечества.

Всё это опирается на действия, ориентированные на будущее: проактивность/инициативность/деятельность/предприимчивость как необходимая часть труда/практики/деятельности/инженерии/культуры/метода работы, заключающаяся в рациональном принятии решений по поводу попадания

в будущем в улучшенное какими-то действиями состояние мира: генерировании разных вариантов будущего, исходя из лучших наших порождающих моделей, сборе информации об этих вариантах, принятии решений о реализации одного из них, а затем вложении сил, денег, ресурсов, нервов (если речь идёт о людях-агентах), времени в реализацию этого оптимального/рационального по соотношению потенциальных выгод и рисков варианта.

Рациональный агент готов на временное ухудшение своего состояния, уменьшение своих ресурсов на время какого-то действия — чтобы оказаться после этого действия в мире, где его состояние лучше, чем ожидалось бы в момент планирования на тот же момент в будущем, где этого действия не было. Рациональный агент предприимчив, инициативен, проактивен (впрочем, это уже говорили в предыдущем абзаце).

Самые успешные деятели — это те, кто творчески генерируют больше вариантов, а потом ухитряются рационально выбрать оптимальные из них на базе самых современных точных моделей мира (включая модели себя!). **Определяющая характеристика в продуктивной деятельности тем самым — качественное порождающее моделирование самых разных вариантов моделей, затем качественная различающая/discriminative критика, что ведёт к качественной модели мира. Забота системного мышления — прежде всего качественная модель мира (в том числе «модель себя» для агента как части мира).**

А занимаются этим «предпринимательством» (помним, слово табуировано, потому что по факту за ним скрываются самые разные методы, которых придерживаются самые разные роли) практически все агенты (люди, AI, организации), хотя не все выбирают быть основателями компаний, быть собственниками бизнеса и тем самым получать не только все возможные прибыли, но и все возможные убытки.

Умных людей удивительно много, миллиардеров удивительно мало: умные люди часто трезво оценивают свои шансы и не играют в рулетку с непрерывно меняющимся рынком, непрерывно идущей техно-эволюцией, это очень нервно и хлопотно и не всем нравится, а даже если и нравится, не у всех получается. Всё это будет более подробно рассмотрено в руководстве по системному менеджменту.

Сложность и меры сложности

Понятие сложности интенсивно разрабатывалось не только в рамках **системных исследований** (т.е. исследований по развитию системного подхода), но и во многих других научных и даже инженерных дисциплинах/теориях. Окончательного согласия по тому, что такое сложность нет, и не предвидится: для разных проектов и разных ролей в проектах интересны разные виды сложности. Seth Lloyd собрал некоторые примеры определений для мер сложности[339]. Все эти определения он отнёс к попыткам ответа на три вопроса:

1. Насколько трудно описать систему? Обычно это измеряется в битах, затрачиваемых на представление описания. Мерами сложности тут будут информация, энтропия, алгоритмическая сложность или алгоритмическое

[339] http://web.mit.edu/esd.83/www/notebook/Complexity.PDF

содержание информации, максимальная длина описания, информация Фишера (Fisher), энтропия Рени (Rényi), длина кода (беспрефиксного, Хаффмана, Шэннона-Фано, корректирующего ошибки, Хамминга), информация Чернова, размерность, фрактальная размерность, сложность Lempel-Ziv.

2. Насколько трудно создать систему? Сложность как трудность создания измеряется во времени, энергии, долларах и т.д. Меры сложности тут вычислительная сложность, временна́я вычислительная сложность, пространственная вычислительная сложность, основанная на информации сложность, логическая глубина (depth), термодинамическая глубина, цена, шифрованность (crypticity).

3. Какая степень организованности? Тут может быть два варианта:

а) «результирующая сложность» (effective complexity), трудность описания организационной структуры, неважно корпоративной, химической, клеточной. Приведём их по-английски: Metric Entropy; Fractal Dimension; Excess Entropy; Stochastic Complexity; Sophistication; Effective Measure Complexity; True Measure Complexity; Topological epsilon-machine size; Conditional Information; Conditional Algorithmic Information Content; Schema length; Ideal Complexity; Hierarchical Complexity; Tree subgraph diversity; Homogeneous Complexity; Grammatical Complexity.

б) количество информации, которой нужно обмениваться между частями системы из-за такой организационной структуры: Algorithmic Mutual Information; Channel Capacity; Correlation; Stored Information; Organization.

Есть и понятия, которые не являются понятиями сложности, но очень близки: Long-Range Order; Self-Organization; Complex Adaptive Systems; Edge of Chaos. Есть и совсем альтернативные меры сложности (например, основанные на скорости описания оцениваемых объектов, а не объёме этого описания[340]).

Ситуация с понятием «сложность» очень характерна для системного подхода: употребляемые в нём слова кажутся вполне «бытовыми» и имеющими ясные и интуитивно понятные с детства значения. Но нет, слова эти вдруг оказываются терминами, за которыми скрываются очень разные понятия, с этими понятиями работают самые разные логические или количественные модели самых разных школ мысли, проводятся количественные измерения самых разных их характеристик для самых различных целей.

В рамках нашего руководства мы будем считать сложной систему из достаточно большого количества элементов, настолько большого, чтобы одному агенту-индивиду не получалось оценить все связи и взаимодействия с точностью, достаточной для рациональных проектных действий.

Если речь идёт, например, об агентах-людях, то в проекте создания и развития организации::система в этой организации легко может оказаться пятьсот человек. Этого более чем достаточно, чтобы в одной голове не получилось оценить все связи и взаимодействия этих людей, даже если одна голова будет вести записи в своём компьютере-экзокортексе. Такая организационная система заведомо сложна.

Одному агенту (даже коллективному) так же трудно удержать все связи и

[340] http://people.idsia.ch/~juergen/speedprior.html

взаимодействия в системе смартфона, в системе атомной электростанции, многих других инженерных системах — расширенные предприятия в таких проектах объединяют работу самых разных отдельных автономных предприятий. Кто-то проектирует чип процессора для смартфона, кто-то изготавливает чип процессора, кто-то проектирует смартфон, кто-то проектирует мероприятия по маркетингу — задействуется множество самых разных предприятий, в них множество умных людей, все преследуют какие-то свои интересы, о некоторых интересах договариваются. Это всё сложно, чрезвычайно сложно.

Для наших целей обучения системному мышлению такого неформального понимания сложности вполне достаточно, но вы должны помнить, что есть и формальные понимания сложности, есть множество теорий сложности, из которых пришли эти формальные понимания.

В физических теориях эволюционного роста сложности биологических систем основное определение идёт по другой линии: количество операций, которые нужно сделать, чтобы создать тот или иной биологический объект[341].

Системное мышление не даёт никаких «объективных ответов» на вопросы о системах. Эти ответы всегда зависят от того, какая роль из какого интереса спрашивает, и какая роль из какого интереса даёт ответ, какие конфликты каких системных уровней пытаемся преодолеть, какой этики придерживаемся.

В системном мышлении нет никакого пошагового алгоритма, приводящего к правильному ответу (и это немудрено: помним о неустроенности: огромное количество самых разных ответов дают огромное множество примерно одинакового субоптимального качества решений), нет последовательности шагов, гарантирующих какой-то приемлемый результат этого мышления, нет «типичных случаев». Всегда найдутся нюансы, которые «типичное» делают абсолютно нетипичным.

Системное мышление сложно, его долго осваивать — этим оно напоминает какую-то «необъективную высшую математику», более-менее типовые рассуждения над набором объектов с типами, предписываемыми системным подходом. Системное мышление даже не трудится давать точные определения своим понятиям: разные агенты норовят приспособить эти понятия для своих самых разных текущих интересов текущих отыгрываемых ими ролей.

При всей этой неопределённости **понятия системного мышления позволяют компактно и просто описывать сложный мир, чтобы управлять коллективным вниманием в сложных проектных ситуациях!** Альтернативные варианты (например, редукционизм или холизм — когда вы явно не используете системные уровни для организации внимания людей в проекте)

---

[341] Complexity is notoriously difficult to define precisely, and no single, universal definition has been agreed upon (24–26), although it seems to be commonly held that “when we see it, we know it.” The definitions of complexity that appear to be meaningful in biology involve, depending on the level of analysis, the number of evolutionarily conserved nucleotide sites in genomes and the number of genes or functional components in organisms or suborganismal functional systems, as well as the hierarchical organization of biological entities, be it functional networks and pathways, cells, organisms, or communities (20–23, 27). Perhaps the most general definition, pathway complexity, deriving from the concept of algorithmic complexity in mathematics, includes the number of steps required to create a given object. It has been proposed that entities with a pathway complexity above a certain threshold can only originate from biological processes (23).И далее по ссылкам в работе «Physical foundations of biological complexity», откуда взята эта цитата, https://www.pnas.org/doi/full/10.1073/pnas.1807890115

оказываются много, много хуже.

Задания: системность мышления окружающих вас людей

Поставьте отметку о выполнении:

1. Вы попробовали определять в разговоре, есть ли системное мышление у окружающих вас людей (те, кто первым ходом обсуждения идут к надсистеме — тех считайте с системным мышлением, а кто сразу идёт без выяснения окружения к подсистемам или системам создания — тех считайте без системного мышления). Написан пост с вашей оценкой распространённости системного мышления в вашем окружении (среди друзей, коллег, знакомых по хобби, близких людей).
2. Вы отслеживали примеры редукционизма и холизма в предлагаемых разными людьми вам объяснениях, а также в собственных объяснениях. Например, попытки высшие психические феномены объяснить в терминах простейших явлений на уровне работы каких-то отдельных частей мозга. После чего обращали внимание на пропущенную часть рассуждений (необходимость учёта эмерджентности в редукционистских рассуждениях и добавки высокоуровневых рассуждений, а в холизме наоборот — учёт того, что происходит на более низких системных уровнях). Написан пост о том, как вам удаётся это делать, и как вы пытаетесь исправить ситуацию с несистемностью как мышления окружающих вас людей, так и вашего личного мышления.

Проблема наборов/collections как систем: чаще всего нет, но иногда — да.

В нашем руководстве приводится некоторое количество примеров групповых систем (например, клиентура и лот). В целом мышление о групповых системах не так просто. Скажем, система ли сообщество, или не система, если организатор сообщества считает его системой, но люди в сообществе занимаются каждый своим делом и не общаются между собой — где там эмерджентность? Система ли выпущенная серия одинаковых деталей, или просто «куча» — детали-то не взаимодействуют, эмерджентность не проявляют? Какая онтологическая природа «обрабатываемой партии заготовок»/batch в операционном менеджменте?

Можно уйти от каких-то ответов на эти вопросы, если объявить все эти объекты «предметами метода», представлять их как «объекты внимания», тем самым сдвинуть рассмотрение вопроса на вопросы методологии, разбираемые в соответствующем руководстве (предметами метода могут быть и системы, и описания, и поведения и что угодно, что меняется методом), но не в нашем руководстве по системному мышлению. Но мы тогда для этих сложных случаев «групповых систем» теряем мощь самого системного мышления. Мы потеряем гарантированное заземление (системы физичны), системное описание (понятно, как его делать), возможность объявления целевой системой с границей к её окружению, рекурсивное применение системного мышления, граф создателей, сложные цепочки отношений для концептов типа stakeholder/customer needs (это к свойствам надсистемы и ожиданиям внешних проектных ролей, а не свойствам самой системы) и т.д. Так что отвечать на вопрос «система ли это» для каких-то наборов/collections систем — надо.

Какие групповые системы нам интересны? Конечно, все, так что мы только дадим несколько примеров — важно тут понять принцип. И, конечно, однозначного ответа «из учебника» на вопрос «а у меня тут система или нет» не существует, ибо всё

зависит от конкретной ситуации и того, как вы затем будете использовать ответ. Поэтому примеры примерами, а думать в своём проекте придётся каждый раз.

Чаще всего затруднения по классифицированию систем происходят со следующими объектами:

- Партии/batches, лоты — единицы обработки.
- Серии (мелкосерийное или крупносерийное производство — это как раз об этом, все партии одной модели) как единицы выпуска, в том числе партии и серии с учётом mass customization (скажем, все боинги 737 выпускаются в уникальной конфигурации, в соответствии с пожеланиями клиентов).
- Парки/fleet (на русском — это по месту стоянки транспорта, а по-английски — указание на то, что движется).
- Запасы/резервы/pools — в том числе пул проектных менеджеров, пул вагонов для парка электровоза.
- Указание на материал (неисчисляемое существительное, типа «молоко — наша система», или «мы делаем полиэтилен» без указания на форму выпуска и количество)
- сообщества каких-то ролей (скажем, вокруг организации можно выделить сообщества сотрудников, инвесторов, заказчиков, поставщиков/suppliers, конкурентов, иногда выделяют сообщества пользователей как не совпадающее с сообществом заказчиков, и часть ролей может играться не физическими "лицами", а организациями)

Ключевое затруднение тут — это указание на то, что эмерджентные свойства возникают в системе в ходе взаимодействия её частей. Взаимодействуют ли детали в их партии/batch? Вроде нет. Случай, когда урановое сырьё приказали размещать кучками вдали друг от друга, чтобы кучки не перегревались, ибо между ними таки было взаимодействие — они экзотические. С другой стороны, когда Элон Маск говорит, что выпустить одну огромную ракету — это просто, а вот наладить серийный выпуск так, что эти ракеты выходили бы из ворот ракетного завода раз в 72 часа, сложно. Вроде бы у него речь идёт о ракетном заводе, и привычно говорить, что целевая система завода — ракета, это решает огромное число вопросов. Но если учесть, что речь о серийном выпуске, появляются дополнительные вопросы. Мы обсуждаем уже не одну систему, а их набор/группу/серию/collection/batch:

- **выпуск** (группу выпускаемых систем, или в случае сервиса — группу систем, для которых был оказан сервис),
- **скорость выпуска** (первая производная выпуска по времени)
- **ускорение выпуска** (вторая производная выпуска по времени, «рост скорости выпуска»)
- **рывок выпуска** (третья производная выпуска по времени[342], «рост ускорения выпуска» или «рост темпа роста скорости выпуска», канцелярит тут обычно побеждает физические термины. Физические — потому что производные вполне математичны, а вот производные по времени — уже физичны, время уже рассматривается физикой, а не математикой).

Мы можем обнаружить эмерджентные свойства у наборов объектов, когда заметим, что ключевое в свойствах системы — это проявление таких свойств в окружение. Взаимодействие может быть не прямое друг с другом, а через

[342] https://en.wikipedia.org/wiki/Jerk_(physics), https://ru.wikipedia.org/wiki/Рывок_(кинематика)

среду/окружение/environment, и проявление появляющихся/эмерджентных/системных свойств будет как раз тем, что среда будет испытывать совокупное воздействие отдельных систем в их наборе/серии:

- **Транспортный парк, флот, пул**. Транспортные средства, вагоны или тележки индивидуально не проявляют свойства транспортной системы, но вместе они могут составить транспортную систему: например, удовлетворять расписанию, выдавать архитектурные характеристики надёжности и доступности. Возможно, для транспортной системы нужно будет что-то добавить к тамошним наборам: службы ремонта и обслуживания, управляющие системы, наличие дорог, наличие топлива, наличие водителей, наличие организаторов движения и т.д. Но даже если рассмотреть полноценную транспортную систему, то составной её частью (подсистемой) можно считать транспортный парк (для судов — флот) и вагонный/тележечный пул. Это свойства, проявляемые во время эксплуатации.
- **Партия/серия/batch/series/round** деталей проявляет эмерджентные свойства, но они относятся обычно ко времени изготовления, а не ко времени использования детали. Скажем, доля дефектов тут --свойство/качество, которое не существует на уровне одной детали. Если речь идёт о каком-то товаре, то какой-то объём партии может обвалить цены на рынке при продаже, а при покупке — наоборот, задрать цены. Если речь идёт о страховке, то актуарные расчёты (размеры страховой выплаты) тоже зависят от вероятностей страховых случаев, эти вероятности относят ко всей совокупности объектов. В биологии речь может идти о свойстве выживаемости для генов, ибо только ничтожный процент каких-то "партий" спермы или яйцеклеток будет доходить до стадии взрослых организмов, дающих потомство.
- если речь идёт о **материалах** (железо, молоко, полиэтилен — всё «неисчисляемые существительные»), то заведомо мы не говорим о системе. Но для перехода к системе тут проще всего спросить о форме и упаковке. А затем перейти к рассуждениям о партии/лоте продукта (например, полиэтилен — «гранулы полиэтилена» — «баночка гранул полиэтилена» или «эшелон гранул полиэтилена», «молоко» — «бутылка молока» — «партия молока»).
- **Системы с избыточностью подсистем** делаются для получения приемлемой архитектурной характеристики надёжности: включение в состав системы нескольких одинаковых подсистем, которые между собой не взаимодействуют, но они дают инженерную избыточность, их совокупная надёжность выше, поэтому осмысленно их рассмотрение как отдельной системы. Конечно, функция там будет одна и та же у одной подсистемы системы и у системы с несколькими такими подсистемами системами (с дублированием функций, если информационные системы — то с проверкой одинаковости ответа от всех). Там тоже эмерджентность получается при включении отдельных объектов-систем в состав надсистемы, но при этом появляются и общие новые свойства, которых нет у одной системы — и поэтому может быть целесообразно выделять группу как подсистему какой-то системы, чтобы иметь возможность обсудить эти свойства, а не прямо включать системы из группы как подсистемы.
- Группа невзаимодействующих между собой однотипных модулей (и их

невзаимодействие будет обеспечиваться архитектурно! архитекторы намеренно будут убирать зависимости!) может быть включена в состав какой-то системы для того, чтобы добавить масштабируемость, **резерв по мощности**. Система содержит слоты, предусматривающие использование переменного числа однотипных модулей в зависимости от потребности. И вот общая производительность может быть отнесена на систему из подсистемы-платформы со слотами и далее трассирована к отдельным модулям-подсистемам, но может быть отнесена и к «группе модулей» как отдельной подсистеме, ибо там могут возникнуть и другие характеристики (скажем, помехи модулей друг другу, тепловыделение и т.д.).

- Сама **«однотипность систем в группе» это тоже эмерджентное свойство, и оно влияет на окружение**! Так, для однотипных модулей можно организовать оптовую закупку самих модулей, оптовую закупку обслуживания самих модулей, закупку этих модулей как "запчастей", оптовое тестирование, а не обращаться к десяткам поставщиков. С учётом скидки на опт это может оказаться крайне важным. Рассуждения тут примерно такие же, как про парк и пул, только речь идёт о времени создания и развития, а не эксплуатации. А во время эксплуатации/использования мы рассматриваем отдельные системы, они не видны как набор/пул. Скажем, вместо уникальных насосов для судна мы закупаем заведомо более дорогие и надёжные насосы, но оптом — и оказывается, что совокупная стоимость владения для такой группы насосов дешевле, чем если их не объединять в группу.
- **Зоны покрытия** территории каким-то сервисом с какой-то плотностью как их эмерджентным свойством тоже такой класс систем, где вроде бы отдельные системы группы не взаимодействуют, но вся группа может выделяться как система с дополнительными свойствами, относимыми не просто к числу систем в группе, но к каким-то пространственным характеристикам. Скажем, вы делаете отдельные площадки для базовых станций сотовой связи. Но покупать у вас будут не отдельные площадки «в розницу», а покупать будут площадки, размещённые на какой-то территории, причём с заданной плотностью, не ниже. Поэтому торгуете вы вроде «площадками», но по факту — группами площадок, размещённых на какой-то территории. Территория зоны покрытия не ваша, ваши там только площадки. Но продавать будете группу площадок как «инфраструктуру площадок в зоне покрытия группы базовых станций». «Инфраструктуры» характеризуются как раз этим: повсеместностью чего-то одинакового, поэтому при обсуждении инфраструктур будут часто возникать такие «группы одинаковых систем» с эмерджентными свойствами группы, привязанными и к числу входящих систем группы, и к их размещению на какой-то территории.
- **клиентура (client base, clientele)** тоже обладает характеристиками целого для тех, кто её «эксплуатирует»: там проявляются разные свойства вроде «трендов» роста, ухода к конкурентам, обновляемости состава и т.д. С этими трендами работают маркетологи и менеджеры сообществ. Тренды характеризуются метриками по тем свойствам клиентуры как целого, которых нет у отдельных клиентов. То же самое с другими ролевыми группами, и даже необязательно «физическими лицами», это могут быть и компании. Скажем, поставщики/контракторы/suppliers будут представлять собой один объект внимания как совокупность (неважно, юрлица или физлица) — и дальше

могут приниматься решения о минимизации или увеличении числа поставщиков, известные как инсорсинг/insourcing и аутсорсинг/outsourcing. Для этого их надо обсуждать как один объект даже в ситуации, когда они вроде не взаимодействуют между собой. Но они взаимодействуют через изменение работы компании, они ведь входят как подсистемы в расширенное предприятие. Скажем, один из ключевых пунктов стратегии SpaceX — это для скорости согласований минимизировать число поставщиков (это было прямо противоположно государственным космическим программам, которые явно дробили заказы для "обеспечения конкуренции, облегчения распространения знаний", и это оказалось очень дорого из-за изобилия ошибок конфигурации и вело к очень большим срокам разработки).

Сюда же к **сообществам-как-системам** можно было бы отнести «учебные потоки» из нескольких групп, набранных в один и тот же год в университете и сами «учебные группы», ибо по образовательной программе в деканате объектами управления идут эти самые «потоки» (группы, набранные в один год и идущие по типовому курсу этого года), а студенты в них отслеживаются на уровень ниже: важно знать число студентов на потоке, чтобы «управлять движением групп по курсам». Но рассуждение тут пограничное между «не взаимодействующими системами в группе систем» и «взаимодействующими подсистемами в системе», ибо эти «потоки» и «группы» всячески стремятся преобразовать в сообщества, где один из видов вовлечения/engagement, дающих чувство принадлежности/belonging к сообществу — это как раз общение в группе, взаимодействие. Это может работать, даже если эта группа создана по совершенно формальным признакам. Вспомните, как учитель физкультуры делил учеников на команды по абсолютно произвольному признаку, скажем, «по росту на первый-второй рассчитайсь!», и через пару минут члены команд дружили::взаимодействие между собой и враждовали с пару минут назад образованной другой группой. То же происходит с сотрудниками предприятий в целом и подразделениями: они наполовину сообщества, наполовину «человеческий материал, из которого сделана компания, одинаковые для кадровой системы «кадры»».

По этой же линии рассуждения про сообщества, в которых не предусмотрено общение между членами сообществ, но которые рассматриваются их организаторами, а также и остальными людьми как принадлежащими к такому сообществу (часто примером такого «воображаемого сообщества» приводят нации, но таких типов сообществ сейчас находят довольно много[343]).

Если вы любите рисовать, то маркетологи кружков и курсов рисования вас найдут: вы для них члены сообщества любителей рисования, их «целевая аудитория». С вами будут не только работать «индивидуально», но ещё и всячески прививать чувство принадлежности к сообществу. Это оказывается взаимовыгодно: у вас будет чуть лучше с ощущением благополучия и смысла жизни, а организаторы сообщества получат лояльного клиента, которого можно «доить» какими-нибудь абонентскими платежами долго, а не просто «разово использовать». Часть проблем, связанных с рассмотрением сообществ как систем, поднята в тексте «Системность и эмерджентность сообществ»[344].

Отдельно нужно упомянуть разнообразные сети, но они не вызывают подобных

---

[343] https://www.sciencedirect.com/topics/social-sciences/imagined-community

[344] https://ailev.livejournal.com/1736164.html

вопросов: в сетях обычно «что-то течёт/flow» и поэтому очевидно взаимодействие, так легко обнаружить эмерджентные свойства, например, «пропускная способность сети», «полезность сети для клиентов» (это отсылка к закону Metcalfe[345]).

Мы по-прежнему отличаем синергию (резкий рост уже имеющегося свойства одной системы от взаимодействия между двумя системами) от эмерджентности (появление вообще нового свойства, которое раньше не рассматривалось). Скажем, оптовые партии, крупные лоты имеют совокупную для всех отдельных систем в их составе цену (неважно, покупки или продажи: если цены покупки и продажи отличаются, то сделки не будет). Это важное свойство: от него зависит принятие решения о производстве или покупке и это свойство зависит от количества объектов в партии или лоте. Оптовые цены, экономия на масштабе — все они связывают цену даже на одно изделие с размером партии. Но цена есть и у одного изделия, и у многих провзаимодействовавших — а нового свойства нет. Поэтому мы говорим о синергии как резком изменении свойства при попадании в какое-то подходящее окружение. Так что лучше всё-таки смотреть на окружение и задаваться вопросом: мы просто улучшаем какое-то свойство (берём 3кг кофе вместо 200г — это просто "много кофе", вес больше), или кому-то что-то надо такое, что не получишь от отдельных входящих в группу продуктов/изделий/единиц? Для примера с кофе тут трудно что-то "придумать", но наверняка встретится где-то проект, где "больше кофе" приведёт к каким-то проявляемым новым свойствам (не будем даже гадать, каким: тем и отличается системное мышление в плане своего применения, что требует конкретного проекта — где нельзя подкручивать ситуацию под предлагаемое решение, а надо подкручивать предлагаемое решение, чтобы оно помогло решить проблемы в конкретной ситуации).

Дополнительно понятие эмерджентности и исследования на эту тему приводились в подразделе «Онтология и отношение композиции: системность» руководства по интеллект-стеку. Какие объекты не рекомендуется называть системами, обсуждалось в подразделе «Не всё системы, что ими называют» нашего руководства, ещё много дополнений было в подразделе «Эмерджентность и мета-системный переход». Эмерджентность — одно из центральных понятий в системном мышлении.

При этом бытует два разных словарных значения слова «эмерджентность»:

- Эмерджентность в понимании математиков: неожиданный скачок свойств в связи с ростом сложности системы — так, например, обсуждают эмерджентность в нейросетях как скачок на графике одной переменной «умения сети» по мере роста переменной «число нейронов в сети».
- Эмерджентность в понимании физиков как появление вообще нового типа графиков, появление новых переменных. Это можно учитывать как противопоставление «одноуровневой» (математической) и «многоуровневой/системной» эмерджентности, подробней можно посмотреть в тексте «Как думать об эмерджентности больших языковых моделей»[346].

В системном мышлении исследование каких-то систем, составленных из вроде как не взаимодействующих непосредственно или минимально взаимодействующих (например, через окружающую среду, через какие-то причинные связи с задействованием конфаундеров — то есть «через мышление агента-создателя»,

[345] https://en.wikipedia.org/wiki/Metcalfe%27s_law
[346] https://ailev.livejournal.com/1725090.html

моделирующего какой-то кусок реальности) довольно распространены. Так, это типовой предмет изучения в предметной области complex systems[347] (изучает системы, где «всё от всего зависит», в том числе и то, каким образом появляется поведение в системах из «одинаковых элементов», которые вроде как работают независимо — но при этом получается внешний видимый макроэффет), а ещё network science (, тут ещё определяются с точным предметом, но сюда в том числе попадает изучение распределённых представлений — ибо сети как раз среда, где возможны такие представления). Там такие ключевые слова как sparse networks, epidemics, synchronization and oscillator networks — но чаще всего это относится к "математической" эмерджентности, одномерной: есть какое-то свойство, растёт сложность сети, и "вдруг" появляется explosive phenomena (скажем, возникает цепная ядерная реакция). То есть взаимодействия есть, они ограничены, но их хватает для того, чтобы резко выросла какая-то характеристика — обычно при росте сети или пороговом воздействии (скажем, пропускная способность транспортной сети от "никаких препятствий для потока" до "стоячая пробка" меняется при добавке буквально 5% единиц транспорта к какому-то предельному значению). Но всё-таки тут довольно легко разобраться с системой (например, мозгом) и подсистемами на несколько уровней (отделами мозга и так далее до отдельных нейронов).

Отдельно можно разбираться с квантовофизическими феноменами (квантовая статистика тут много чего может рассказать) и дальше с квантовоподобными/quantum-like моделями (их разрабатывают в лаборатории Андрея Хренникова[348]), используя разработанный для систем микромира математический аппарат для систем макромира ровно так же, как когда-то изобретённый для описания механики математический аппарат дифференциального и интегрального исчисления оказался удобным для описания самых разных областей физики — и оптики, и электромагнитных явлений, и много ещё чего, что уже не так прямо касалось механического движения, для которого и был разработан этот математический аппарат (подробней это рассказано в руководстве по интеллект-стеку фундаментальных методов мышления).

Математика квантовых полей, использованная для описания не квантовофизических явлений, помогает разобраться в том числе и с эмерджентными феноменами в системах с минимальными взаимодействиями или даже вовсе без взаимодействий. Вообще, квантовоподобные модели очень интересны, ибо они (через ренормализацию[349]) позволяют исследовать системы с памятью (например, спиновые стёкла, нейронные вычислители и даже эволюцию в целом). Такие системы проявляют эффекты макроповедения, получающиеся от микроповедения как поведения/взаимодействия индивидуальных систем.

Скажем, квантоподобные модели могут показывать эффекты макроповедения сообществ на базе микроэффектов автономного индивидуального поведения агентов в этих сообществах. Этими агентами могут быть и люди (неэргодические системы!), а сообществами — сообщества людей (сообщества — тоже неэргодические системы!). Опять сошлёмся тут на работы Андрея Хренникова и сотрудников его лаборатории, опять сошлёмся на то, что подробности можно

---

[347] https://en.wikipedia.org/wiki/Complex_system
[348] https://lnu.se/en/staff/andrei.khrennikov/
[349] https://en.wikipedia.org/wiki/Renormalization_group

посмотреть в руководстве по интеллект-стеку.

Ещё эмерджентные свойства в физических неравновесных системах могут появляться благодаря внешним ограничениям или воздействиям (скажем, частицы могут собираться в какие-то сгустки без прямого взаимодействия между собой).

Интересное направление — это рои роботов с относительно простым поведением каждого робота. В целом такой рой демонстрирует сложное эмерджентное поведение, хотя каждый робот следует очень простым правилам. Роботы формально не взаимодействуют при этом, но они могут ориентироваться по каким-то внешним сигналам, в том числе замерять положения относительно друг друга. Стая хищников на охоте как раз демонстрирует возможность поймать добычу, которая недоступна одиночным животным. По этой линии «макро-микро» исследуются самые разные коллективные феномены, например, как возникает коллективное внимание в социальных сетях[350], при этом предупреждается, что «коллективное внимание» определяется иным образом, чем внимание в когнитивной психологии.

Кто хочет копнуть ещё глубже — смотрите про эмерджентность в обзорах философской энциклопедии[351].

В любом случае, смотрите на ситуацию своего проекта: для одного проекта будет плохим выбором обозвать системой кучку картошки, это не будет системой, но для другого проекта это может оказаться очень удачным ходом. Что для вашей роли в проекте хорошо, для другой роли может оказаться плохо. Придётся думать, придётся договариваться, и даже придётся договаривать всех.

Прежде всего ищите в таких случаях взаимодействие, возможно не непосредственно между отдельными объектами группы, а через среду (хотя бы и в форме передачи сигнала). Ищите совместное действие со средой. Если же какой-то вариант взаимодействие находится (возможно, не в момент использования, а в момент создания — скажем, для «партии товара», особенно если там находится что-то общее, типа общей упаковки), то вероятность того, что вам удобно будет назвать набор однородных предметов системой резко вырастает. Если взаимодействия и макроэффектов совместного поведения на окружение нет — то, возможно, это «просто куча», набор однородных систем.

На картинке то ли фрагмент какой-то системы, то ли просто много людей.

---

[350] https://www.nature.com/articles/s41467-019-09311-w

[351] https://plato.stanford.edu/entries/properties-emergent/ и https://plato.stanford.edu/entries/physics-interrelate/

9. Целевая система и её надсистема

Сначала найти целевую систему

Первое, что нужно делать в системном мышлении — это не думать о всех системах сразу и глубокомысленно заявлять, что «всё системы, и всё со всем связано». Это правда, но ресурсов не хватит думать о всех системах сразу, а связи имеют разную важность. Поэтому системное мышление наоборот: помогает думать не обо всех системах сразу, а также выявлять самые важные связи между системами.

Увы, «менеджерские варианты» системного мышления как раз говорят, что важно много думать, но не говорят, как именно думать: не дают метода мышления. А ведь «менеджерские варианты» изложены в самых издающихся и популярных книжках зарубежных авторов в области системного подхода в менеджменте, выпущенных в конце прошлого века (при чтении литературы всегда смотрите на даты её издания!).

Этот «менеджерский вариант» условно можно отнести к «полуторному» поколению системного мышления, потому как вроде бы говорится о системах создания (предприятиях), но задействуется первое поколение системного мышления: «всё очень сложно, система-предприятие состоит из частей, нетривиально связанных, и само это предприятие находится в очень сложном мире». Увы, дальше предлагается просто «думать больше, думать более тщательно», но не рассказывается, как именно — кроме ясного посыла, что к редукционизму неплохо бы добавлять холизм и думать всегда об окружении сначала, а собственно системе потом. Посыл верный, но пользы от такого мало.

Чтобы получить быструю и непосредственную пользу, в коллективном проекте надо понимать, что системное мышление должно быстро договорить всех вокруг — и для этого неплохо бы иметь понимание, зачем это все тут собрались в проект. Для этого надо выделить ту главную систему, которую вы в проекте хотите создать и

развивать с нуля (greenfield), или развивать/изменять (brownfield). То есть первый ход — это сначала в графе создателей определить ту систему, для которой самые разные создатели объединяются мышлением в граф по отношению создания. Эта система называется целевой, без договорённостей по поводу неё договориться обо всём остальном не удастся.

Сначала в проекте нужно договориться о том, какова целевая система (system-of-interest): понять её функцию в надсистеме и её границы, назвать её, чтобы можно было обсуждать дальше. Это очень непросто, речь идёт о творческом/предпринимательском/проактивном решении, высокорисковом моделировании неопределённого будущего. В целевую систему будет вложено много самых разных ресурсов, хорошо бы, чтобы она была успешна — чтобы она принесла выгоду, а не убыток, причём выгоду мы тут понимаем достаточно широко, речь необязательно идёт о денежной выгоде.

Чаще всего целевая система на момент достижения договорённостей о ней ещё не создана, проект только-только начинается, «ничего ещё не известно» — но системное мышление говорит, что это и есть повод задуматься: а что же создаём, что будем развивать? Но если ошибиться и не договориться, то все участники проекта будут создавать разные системы — и проект не получится, не будет успешным, реализуется басня про лебедя, рака и щуку.

Особенно важно не промахнуться в том, что если вы создаёте инструменты для какого-то метода (скажем, Software-as-a-Service, то есть какое-то серверное приложение, или инструменты какого-то сервиса), то целевыми тут являются системы, к которым вы эти инструменты применяете, а не сами создаваемые инструменты. Поговорка, что «никому не нужны свёрла, нужны дырки в стенах» — как раз про это, акцент всегда должен быть на системы, которые обрабатываются вашими инструментами, хотя самое сложное — это создать инструмент. Создатели обычно много сложней, чем изготавливаемые ими системы. Построить завод по массовому выпуску автомобилей много сложней, чем построить один прототип электромобиля в мастерской, причём не обращая внимания на стоимость постройки этого прототипа.

Чтобы создать инструмент, надо понимать, что вы этим инструментом собираетесь обрабатывать, что именно вы создаёте инструментом в момент эксплуатации инструмента. Вот это «создаваемое инструментом», «создаваемое предприятием» чаще всего и будет целевой системой. Но «чаще всего» — это не всегда, и с этим надо разбираться подробней.

Большинство затеянных проектов по созданию систем оказываются неудачными, и мы о них не знаем только потому, что о неудачах не принято трубить, трубят только об успехах! Систему ведь нужно создавать сейчас, а для этого предвидеть её успех в будущем. Но будущее предвидеть гарантированно в общем случае нельзя, можно только угадать — и это часто не удаётся, как минимум это не удаётся сделать с первой догадки, нужно несколько попыток. У вас должна быть какая-то «порождающая/generative модель мира» и вы должны будете сделать демоделирование/рендеринг/вообразить на основе этой порождающей модели мира с непонятной точностью и подробностями вашу целевую систему — это всё-таки «угадывание» в конечном итоге.

Целевую систему вы должны всё-таки угадать, а не «вычислить по формуле»,

«вывести по алгоритму». Дальше эту догадку надо прокритиковать, попытаться улучшить, если она оказалась хуже других догадок, и в итоге выбрать лучшую из имеющихся (поэтому обычно делают много догадок, и только выжившую критику принимают всерьёз, то есть кладут в основу проекта создания системы).

Про целевую систему говорят, что её «выявляют»/discover в ходе стратегирования. Стратегирование — это когда вы придумываете/выявляете метод улучшения вашей конкретной ситуации. Этот метод работы, который должен вам помочь, называется стратегией. Часто стратегирование заключается в том, что вы говорите, что ваша стратегия улучшит ситуацию тем, что вы будете делать такой-то класс систем, например, что вы будете создавать::«метод работы» автомобили::«результат работы по методу», будем «торговать электроэнергией»::метод с результатом работ — «сделка», будем «учить»::метод «мастеров игры в гольф»::результат метода. Выполнение работ по придуманному методу (реализация стратегии, получение результата) и должно привести к улучшению ситуации.

Успех — это что никто не отберёт от вас результат, не будет недовольных, вы получите прирост ресурсов, а не убыток от реализации стратегии. Вот целевая система — это и есть то, что создаётся в ходе выполнения какой-то стратегии::метод (обычно стратегии для какого-то коллективного агента — организации, где все договорились, как будут пользоваться ресурсами достижения результата в работах по стратегии. Конечно, в основе стратегии лежит догадка/гипотеза. Иногда эту догадку/гипотезу называют «предпринимательская ставка»/«entrepreneurial bet», как в азартной игре, ибо это только гипотеза, что команда выиграет при реализации стратегии — догадка о целевой системе и методах работы по её созданию может оказаться неверной, ресурсы будут инвестированы/вложены, а «улучшения ситуации»/выгоды/«превышения доходов над расходами с учётом дисконтирования» не будет. Иногда говорят и по-научному, «предпринимательская гипотеза» (и оценкой её верности при старте проекта занимается роль шумпетеровского предпринимателя, в наших руководствах она называется короче — «визионер»).

Если нет беглости в системном мышлении, то вы будете определять целевую систему неправильно не только потому, что плохо предвидите будущее, но и просто из-за ошибок непродуманности. Вы будете терять в мышлении требующие вашего внимания объекты, и это привнесёт в проект риски. Скажем, вы определили целевую систему как поведение — всё, вас мало кто поймёт, вы мало с кем договоритесь. Типичная фраза: «моя целевая система — это разработка софта», и никакого звоночка в голове что «разработка::метод» это не система, отглагольным существительным называют обычно функции/методы::поведение, а неявно помянут софт в качестве системы и создатель софта в качестве ещё одной системы, но мы помним, что с софтом надо ещё разобраться, ограничивается ли дело только программой, или надо ещё поглядеть, кто создаёт организацию, которая использует софт для получения искомого результата. И дополнительно ещё «моя целевая система» — точно ли «моя»? Целевая система нужна для того, чтобы договориться всем, договорить всех, а не вам со всеми договориться. Хотя если инвестор вы лично — нет проблем, «моя целевая система» пойдёт, всех остальных наймёте, но всё равно «поведение» системой называть не стоит, нанятые вообразят в качестве целевой системы каждый что угодно — и проект развалится, коллективного мышления не случится.

Если вы определите целевую систему неправильно, то вы просто будете заниматься

не тем, чем надо, будете зря тратить время: фокус вашего внимания будет не на важном, а «рядом с важным» или даже «далеко от важного». **Системное мышление — это про осознанное управление вниманием, удерживание этого внимания на важном, определяемом типами объектов системного подхода — даже когда лень, «не хочется», когда думать трудно.**

Системное мышление требует собранности (ещё и поддержанной компьютером: результаты системного моделирования должны быть записаны, чтобы не было шанса их забыть). Например, если вы высказали догадку о целевой системе, то системное мышление тут же говорит: «смотри от границы системы наружу, в надсистему. И не думай пока над прозрачным ящиком, тем более во время создания. Разберись сначала со временем использования, с чёрным ящиком). Это трудно — выдерживать это замечание, но без этого не поймёшь, что же тебе надо делать, каковы будут создатели и какие методы работы эти создатели будут применять для создания системы. Это трудно, поскольку речь обычно идёт о самом старте проекта, когда про целевую систему почти ничего нельзя сказать, но системное мышление требует рассказать о том, что она делает в каком именно окружении, зачем она это делает и почему это кому-то может быть нужным. Это буквальное следование мантре системного мышления.

Успешность (в том числе коммерческая успешность) создаваемой системы не может быть только «спроектирована» и тем самым гарантирована (неопределённое будущее всегда внесёт свои коррективы, исследования показывают, что доля удачи в успехе больших проектов больше, чем доля того же интеллекта, это подробней рассказывается в руководстве по методологии в разделе по стратегированию). Тем не менее, удача может присутствовать — но всегда есть шанс, что она не будет использована из-за ошибок мышления. Системное мышление поможет избежать ошибок мышления в их наиболее вероятных местах, поможет договориться с самыми разными людьми, достигшими мастерства в исполнении самых разных профессиональных проектных ролей и поэтому заранее предупреждающих о самых разных рисках. Эти ошибки системное мышление поможет отслеживать в ходе всего проекта по созданию и развитию системы. Вам неоднократно придётся замысливать, проектировать и изготавливать, тестировать и эксплуатировать, модифицировать систему снова и снова, выкидывать/ликвидировать ненужные экземпляры системы, закрывать проект создания и развития системы за исчерпанием интереса к системе — и системное мышление требует, чтобы вы и все остальные участники проекта чётко понимали, о какой именно системе идёт речь, чтобы все могли договориться.

Вовремя замеченные признаки возможной неуспешности проекта создания целевой системы дадут команде проекта шанс как-то на них отреагировать, в том числе и отреагировать путём закрытия проекта — тратить ресурсы на заведомо неуспешный проект ни в коем случае нельзя, а **системное мышление поможет обнаружить необходимость закрыть проект раньше. Системное мышление как санитар леса: заведомо дохлый проект оно убивает сразу, экономя время, нервы, деньги всех участников такого проекта.** Если вы обнаружили, что создаётся софт ERP, а нужно создавать подразделение, которое будет задействовать софт для планирования работы предприятия, но этим подразделением никто не озабочен, все озабочены только софтом — всё, это неудача, бегите из такого проекта (если, конечно, вы не считаете, что можно скушать деньги на бесполезный проект, а отвечать за скушанные вами зря деньги будете не вы, а кто-то другой — это для вас

этически приемлемо. Если неприемлемо — или реорганизуйте проект, или убегайте).

Системное мышление поможет и найти способы продолжить двигающийся к неуспешности проект, увеличить вероятность его успешности — оно подскажет, над чем нужно будет задуматься в первую очередь, на что потратить время размышлений о создаваемой и развиваемой системе. Системное мышление не столько помогает преодолеть проблемы (придумать, как решить проблемы), сколько вовремя выявляет проблемы, обращает на них внимание (буквально: оно же про управление вниманием!). Системное мышление больше связано с критикой, чем с выдвижением догадок. Системное творчество связано с выдвижением догадок и последующей критикой, так вот системное мышление говорит только, на какую тему нужны догадки (но не даёт метода выдвижения догадок), а потом помогает эти догадки прокритиковать. Изобрести системное мышление само по себе ничего не может, но может подсказать, что изобретать — и затем прокритикует, если окажется, что вы изобрели что-то не то.

Что надо делать, если вы определились с целевой системой, но догадка оказалась неверной? Дальше системное творчество: при найденных конфликтах и противоречиях нужно или подправить текущую догадку, или выдвинуть новую догадку. И тут нужно помнить, что единственного верного решения нет, решений огромное количество, но все они субоптимальны, и лучшие из доступных решений примерно одинаковы по их качеству (неустроенность). Кардинальные улучшения, «гениальные догадки» — редки!

Предприимчивость/проактивность/деятельность (тут слово «деятельность» как противоположное слову «бездеятельность», пассивность) понимается в фундаментальных методах мышления совсем необязательно в классическом экономическом смысле слова (то есть системы создают и развивают совершенно необязательно из-за желания заработать деньги, вложив/инвестировав сегодня в ресурсы, имеющие небольшую стоимость, а получив побольше завтра за обработанные ресурсы, которые имеют стоимость больше). Но в любом случае **деятель/практик/трудящийся/инженер, что-то предпринимая/делая, выполняя какое-то дело/проект, прикидывает его успешность: пытается представить будущий мир, в котором результат проекта (система) существует — и пытается понять/познать/learn (в том числе деятельно, «активным зондированием», проводя эксперименты в физическом мире, задавая вопросы и т.д. — это не «умозрительно понять», это провести полноценное исследование/познание с экспериментами!). Участники проекта должны совместно представлять, что самые разные проектные роли будут ощущать по итогам этого проекта — то есть оценить, будет ли проект успешным.**

Проекты могут быть самыми разными. Кто-то может танцевать («делать из себя артиста-танцора» — не забываем, что поведение всегда относим к каким-то системам, которые надо будет создать), но и в этом случае нужно думать о том, чтобы перформанс был успешен: успех перформанса артиста-танцора определяется не успехом в глазах только его самого. В некоторых странах на похоронах не принято танцевать, а в некоторых странах наоборот, принято — с гробом на плечах. Нужно смотреть на обстоятельства танцевания, а не только на сам танцевальный перформанс как связанную только с танцеванием артистов-танцоров систему. И только ли артист-танцор тут будет целевой системой в момент перформанса, или

артист в момент его функционирования только подсистема в чём-то большем. Артист-танцор — это часть перформанса как микро-мероприятия, но и тут можно не останавливаться — частью чего является перформанс? Фестиваль, вечеринка, концерт? Может быть, коллективная договорённость достигается по этому поводу, а не по поводу подготовки танцора и доучивания его до уровня артиста и целевого выхода на перформанс?

Системное мышление не даст ответа на подобные вопросы, но оно даёт типы для того, чтобы вы могли задаться такими вопросами в вашем проекте — и в результате быстро коллективно договориться, что же вы-команда создаёте и развиваете. И ещё обратить внимание, что составы агентов этого «вы» могут быть существенно разные в зависимости от того, о каком проекте вы думаете, какого проекта вы считаете себя участником. А если вы участник нескольких проектов, то они ещё могут и конфликтовать, например, два проекта каждый могут срочно требовать от вас отдать 100% рабочего времени именно ему, и надо будет как-то объяснить и себе, и окружающим, почему вы тратите свой ресурс так, а не иначе.

Проекты существенно различаются по их масштабу. Так, кто-то может замысливать культурную революцию, воображая сообщество или даже общество, которое ведёт себя по-другому, чем сейчас — и это будет для него целевой системой. Осталось только найти участников проекта, для которых это же общество будет целевой системой с теми же характеристиками (о которых ещё надо договориться).

В любом случае, в проекте нужно реалистично вообразить: что (какая система) будет в итоге выполнения проекта. Более того, надо выполнить все предписания системного подхода: оценить не только, какой окажется целевая система и каково будет её поведение в момент эксплуатации, но и окружение этой целевой системы и влияние поведения целевой системы на это окружение, и что на эту тему влияния целевой системы на окружение будут думать внешние проектные роли, и что будут делать агенты, которые играют эти роли, если им проект понравится, и если не понравится. И это мы не касаемся систем в графе создателей! Но эти системы должны договориться, эта оценка результата проекта (целевая система с её функцией/методом работы в её окружении, удовлетворение интересов внешних проектных ролей) должна быть общей для создателей, а поскольку речь идёт не только о создании, но и о развитии системы, эта оценка результата проекта должна проводиться на протяжении всего проекта, а не только в его начале.

Предпринимательский/проактивный/творческий/исследовательский характер любой деятельности в проекте происходит из-за того, что все стратегии и реализующие их планы — это только «прогнозы»/гипотезы/догадки, а также они являются не «объективными» (о них не догадываются как о «законах природы»), а предметом договорённостей (о них догадываются как о том, по поводу чего проще будет договориться), плюс договорённости могут «передоговариваться», будущее невозможно предугадать точно, а предсказание будет верным, но вряд ли будет верным долго.

Эта неопределённость заставляет и инженеров-технарей, и менеджеров как инженеров организации с этими инженерами-технарями, и инвесторов в эти организации, и политиков, которые заняты системами более высоких уровней, снова и снова думать о том, как в конечном итоге будет использована целевая система в её окружении и какие ещё могут быть эффекты от этого использования, удерживать внимание на ситуации использования целевой системы и отслеживать

происходящее вокруг этого использования, хотя это использование будет только в будущем.

Все инженеры, когда создают и развивают свои системы, воображают их работающими ещё при замысливании и проектировании — без этого не было бы инженерии! Не было бы ни классической инженерии, ни инженерии предприятия, ни социальной инженерии! Неизвестность будущего не отменяет необходимости иметь порождающую модель (то есть модель, по которой вы можете породить образ будущего как варианта воплощения в нём целевой системы с какими-то параметрами), из которой вы обязательно должны вообразить/отрендерить/породить образ этого будущего и представить себе все возможные приятные и неприятные сюрпризы в этом будущем!

В системном мышлении нужно всегда думать о системе в её системном окружении (время использования!) и о мнении внешних проектных ролей по всему графу создателей (внешние проектные роли должны быть готовы заплатить за создание и/или модернизацию целевой системы и систем её создания, но помним и о тех внешних ролях, которые будут считать проект неэтичным, сколько бы вам ни сулили за его реализацию. За изготовление газовых камер могут платить, и неплохо, но возьмётесь ли вы за такой проект?). Если система никому не нужна (то есть никому не нужно её поведение в надсистеме, ни для чьей надсистемы не нужна функция этой системы в достаточной мере, чтобы окупить затраты на создание системы), если система заведомо неуспешна по коммерческим или этическим соображениям — её обычно не делают. Ну, или делают, когда кто-то готов за это заплатить ради морального удовлетворения. Случаи, когда «готов заплатить, но не свои — например, своровавные деньги, или деньги налогоплательщиков», мы не рассматриваем.

Перед тем, как пытаться оценить успешность системы, нужно её найти/выявить/discover. Это стратегирование, предпринимательская/проактивная/enactive часть инженерии/деятельности/труда, и это должно выполняться коллективно.

Целевая система в системном мышлении — это как начало координат в ментальных «полярных координатах». Все остальные решения самых разных ролей по поводу выделения других систем — надсистемы, систем в окружении, подсистем, систем создания — делаются по отношению к целевой системе. Чтобы можно было понять, по поводу чего принимаются эти решения, сначала договариваются о целевой системе. И успешность тоже тем самым определяется по отношению к целевой системе и проекту её создания. Это в разговор о том, что будет целевой системой проекта, вовлекает не только роли в команде проекта и всём графе создателей, но и внешние проектные роли.

Как использовать знание целевой системы? При любых попытках договориться или кого-то договорить:

- Вы объявляете, что вы с контрагентом (или два контрагента и вы, если вы кого-то договариваете) в общем проекте по созданию целевой системы — и называете эту систему. Контрагенты должны согласиться с предлагаемой целевой системой. Если нет общей целевой системы — то что вы вообще делаете вместе? Общность интересов устанавливается именно на основе того, что вы и ваши контрагенты находятся где-то в графе создателей одной целевой системы.
- Теперь вы предъявляете связь того, что делает контрагент в отношении целевой системы — путь по отношениям начального создания и последующего развития в графе создателей.
- Далее предъявляете свой интерес к целевой системе (вы же в одном проекте!).

И только после этого говорите, что у вас за дело к контрагенту. Без того, что вы и контрагенты в графе создателей одной целевой системы все вы там — «агенты с улицы» (речь ведь не обязательно о людях, агентами могут быть и организации — вы и ваши собеседники могут представлять, интересы организаций, и помним о

конфликтах между системными уровнями, интересы представляемых контрагентами организаций и интересы ваших контрагентов в переговорах могут и отличаться, свои организации они могут и легко продать за тридцать сребренников). Без предъявления рассказа, что вы в одном проекте (создаёте одну целевую систему, только на разных местах в графе создателей) вы все — случайные собеседники, кредит доверия минимален. Но если вы и контрагенты в одном графе создателей какой-то целевой системы, и это вам обоим известно — ситуация меняется, вы в одной команде проекта на понятных вам местах. И поэтому может быть понятен интерес вас друг ко другу: вы хотите сделать проект успешным! Поэтому во всех договорённостях проверяется: если в них целевая система, или это частная договорённость, которая смоется в унитаз истории после того, как её перебьют другие договорённости — по поводу реальной целевой системы, а не того, что вы решили сделать маленькой вашей частной целевой системой в ходе переговоров. Целевые системы позволяют выстроить самых разных агентов в граф создателей, без их упоминания очень трудно договориться о том, какое отношение два агента из непонятных мест в графе создателей пока непонятно ещё чего (потом это будет целевой системой) имеют друг ко другу, о чём им надо договариваться, какой там может быть взаимный не только предмет интереса, но и интерес.

Повторимся: целевая система — это вид системы, поэтому для выявленной вами целевой системы **вы должны положительно ответить на все вопросы чеклиста выявления любой системы.** Собственно, наставления нашего руководства в существенной мере как раз и даёт вопросы для такого чеклиста. Вот только несколько пунктов:

- **Мы думали о ней прежде всего как о чёрном ящике в момент её работы** (а не прозрачном ящике в момент создания). Всё остальное мы думали только потом, после этого.
- **Это вещь, а не процесс/метод.** Но она, несомненно, что-то делает (или будет делать, если ещё не существует) в окружении, функционирует/работает по какому-то методу, выполняет какую-то функцию в окружении. Если речь шла о процессе/методе, то не забыли поинтересоваться двумя системами: одна агент-создатель (не меняет своего состояния, как молекула катализатора) и вторая — у которой меняется состояние. Но сам процесс не перепутан с системой! Это проверили три раза! Не забыли, что система может быть весьма неклассической (мероприятие, «договорка», игровая сессия), есть способы об этом думать как о вещи. Но метод — точно не система!
- **Это не описание.** Если уж речь зашла об описании, то проверили, что там за система описана этим описанием. Система физична, можно указать на место в пространстве-времени!
- **Если есть слово «система» в названии, то мы его игнорируем** (оно вполне может быть бытовым словом и сбивать с толку — типа «системы похудения»).
- **Где проходят границы системы и какая её функция — договорились,** она не «найдена объективно существующей». Что для вас «объективно», для других участников проекта будет пустой звук. Проверили, что эта договорённость известна всем участникам проекта и соблюдается.
- **Отслеживаем, не поменялись ли договорённости,** не поменялась ли система в ходе проекта: жизнь не останавливается, выявленные системы тоже меняются**.**

- **В названии системы звучат слова из какой-то предметной области.** Никаких «оборудование» или «потребительский товар» или даже «сервер» — любой такой широкий класс объектов в названии не участвует. Вы не назвали систему «зверь», поскольку один подумал о мышке, а другой — о тигре, и проблем разночтения с этим потом не оберёшься.

Про системы надо думать, применяя этот чеклист буквально. Если вы сказали, что «наш завод выпускает оборудование», то автор нашего руководства по системному мышлению ехидно заметит, что это оборудование хлебопекарное или сантехническое, а ваши сотрудники будут считать, что это оборудование, которое меньше всего вам придёт в голову. В руководстве по рациональной работе вам рассказывалось, что надо выбирать уровень абстракции, когда что-то говорите, нельзя использовать абстракцию на любом уровне, этот уровень надо отслеживать. Если фраза относится непонятно к оборудованию какого проекта (вашего или не вашего, слишком общо сформулировали), то лучше бы вы её не произносили. Это общая ошибка — говорить фразами из учебников инженерии и менеджмента, которые приложимы ко всем ситуациям вообще. Поэтому при обсуждении рабочих проектов такие фразы смысла не имеют, если вы не обсуждаете сами эти принципы инженерии и менеджмента, а обсуждаете свой рабочий проект.

Хорошо подвешенный язык не спасёт, спасёт следование наставлениям руководства, для этого их надо знать. Хорошо подвешенный язык просто продемонстрирует, что вы плохо формулируете мысли, мутно доносите содержание.

Буквальное применение чеклиста такое: если сказано, что система это вещь, а не поведение вещи, то «у меня системой является увеличение производительности труда» — показывает, что вы глаголы от существительных не отличаете (вы говорите «система = увеличение», но увеличение — это не вещь, место увеличения::процесс::поведение в пространстве-времени вы не покажете). Если попытаетесь заменить систему с «увеличения производительности труда» на саму «производительность труда», то производительность — тоже не вещь, а характеристика труда, а если попытаетесь заменить производительность труда на сам труд, то труд — это опять-таки поведение, а не вещь, тоже не проходит чеклист. И если вы не включаете думалку, то подряд несколько раз при попытке выявить систему (не только целевую, но любую) будете выглядеть нелепо, ибо покажете окружающим вас людям, что не в состоянии разобраться, что в мире занимает какое-то место в пространстве-времени, а что — только описания, свойства, поведение чего-то в материальном мире (есть много ещё чего, кроме вещей).

И так буквально надо воспринимать все пункты чеклиста по выявлению систем, а также все наставления наших руководств. Руководства говорят, как лучше делать вашу работу, чтобы быть умным. На работе надо работать: быть умным и не делать много ошибок, платят (даже если ваша работа заключается во владении фирмой) за это.

Целевая система зависит от того, кто её ищет

Обычное дело, когда на каком-нибудь заводе топ-менеджеры признают, что 50% сотрудников не знают, что выпускает предприятие — у них перед глазами только какие-то части каких-то систем времени создания, а вот что там из этого становится инструментами создания целевой системы, а что — самой целевой системой, за какие характеристики целевой системы платятся деньги, как связана их работа с

этими деньгами, как вообще выглядит целевая система в момент эксплуатации — это неведомо. Это плохо, ибо это означает, что многие люди принимают неоптимальные решения: они буквально не знают, что надо сделать, чтобы целевые системы становились лучше. Надо сделать вес детали поменьше, или прочность повыше, или цену поменьше, или класс обработки поточнее при заданном бюджете — что важнее? Это зависит от того, куда эту деталь планируется поставить. А если ты не знаешь? Решение будет «какое-нибудь», и так будут приниматься «как-нибудь» сотни и тысячи решений. По поводу целевой системы обязательно надо договориться.

Вы менеджер системы создания, хотите поднять производительность труда, поднять скорость выпуска продукта. Если это сервис, то там тоже надо поднимать скорость выпуска продукта, но продукт этот «на давальческом сырье» — вы просто меняете состояние не вашего материала, а материала чьей-то чужой системы. Но это и будет целевая система: то, что вы изменяете вашим проектом в целом, вашим мастерством и инструментарием на выходе большого графа создателей. Проект определится масштабом коллективных договорённостей в вашей организации. Так как определить скорость выпуска продукта, нельзя же её поднимать, не зная, что это? Выпуск — это количество экземпляров целевых систем, которые изготавливает проект системы-создателя (то есть количество экземпляров готовых систем на выходе всего графа создателей целевой системы). Скорость выпуска — это выпуск за небольшую единицу времени. Скорость — это название первой производной по времени от какой-то величины, а производные определяются по отношению к «дельтам» как малым величинам. Так что следим, чтобы это не было «систем в год», цельтесь в минимальную единицу измерения — секунды, дни, но не месяцы и годы, ибо вы слишком поздно будете знать ваши результаты, чтобы что-то менять как операционный менеджер.

Если неизвестно, что такое «целевая система» (класс/тип продукта/изделия) и сколько вы их выпускаете за период времени — неизвестна скорость выпуска. Если скорость выпуска (проход/throughput, но нас волнует не денежное выражение, а натуральное выражение) неизвестна, то нельзя сказать, как какое-то организационное изменение, которое вы затеяли, на неё влияет. Надо договориться, что же является целевой системой, все расчёты выпуска дальше вести для неё, а не для какой-то другой системы.

Первое, что должно интересовать менеджера — это целевая система, а не граф её создателей, за который менеджер формально ответственен. Менеджера не должно сначала интересовать предприятие::создатель::система. Ибо если не знаешь, что должен создавать, то никакое решение по поводу создателя не будет осмысленным. Сначала «что надо создать» — потом думаем о том, «каким методом будем создавать», потом «кто будет создавать, умеет ли он такое делать».

Надо ли нам увеличить число сварщиков, чтобы ускорить выпуск продукции? Ответ на этот вопрос будет понятным, только если мы пробежимся по системной мантре:

- Что делает целевая система в системном окружении.
- Из чего её надо делать, чтобы она могла это делать.
- Каким методом надо систему делать, нужны ли там вообще роли сварщиков для этих методов.
- Какой там граф создателей, который будет создавать и развивать систему, какими методами мы будем делать создателей. Какие границы «мы» в

исходном вопросе? Где «мы» на этом графе? Нужны ли там роли сварщиков для создания и развития систем в графе создателей?
- И только тут можно будет прикинуть, надо ли увеличивать число сварщиков по методу критической цепи[352] или какому-то другому методу принятия решений в операционном менеджменте.

Это очень тонкий момент: вам надо не целевую систему строить такую, какую может сделать ваша организация, а организацию (создателя) делать такого, какой сможет выпустить целевую систему. А целевая система должна быть такой, чтобы она была успешной. Если вы плохой повар и будете печь пирожки уж как сумеете (но шикарно организуете предприятие, будете печь плохие пирожки быстро и дёшево), то заниматься вы будете этим только пока деньги инвестора не кончатся. Вам придётся-таки разобраться с целевой системой:

- Какие пирожки должны быть, чтобы они были кому-то нужны (сегодня говорят ещё жёстче — не «кому-то нужны», а «снимали чью-то боль», то есть очень и очень нужны, а не просто «могли бы пригодиться»).
- Какое мастерство и какие инструменты должны у вас быть, чтобы выпускать хорошие пирожки.
- И только после этого можно заняться предприятием, до этого момента надо разбираться с пирожками.
- И пока вы будете заниматься предприятием, надо не забывать заниматься пирожками: они могут успеть перестать быть нужными или может оказаться, что нужны немножко другие пирожки, для которых нужно будет перестроить граф создателей (вместо пирожков с вишней начать выпускать пирожки с мясом — и там всё по линии приготовления начинки в графе создателей придётся переделывать).

Это крайне важно: в мышлении целевая система должна быть сначала, а предприятие-создатель (и даже весь граф создателей) потом. **Менеджер, который объявляет целевой системой свою организацию — ошибается, ибо он теряет возможность договорить всех в рамках всего проекта, в рамках всего графа создателей.** Как минимум, он может не договориться с клиентами, мнение клиентов по поводу его восхитительного предприятия будет негативным, поскольку клиентов волнует целевая система и не волнует предприятие. Поэтому лучше бы целевая система у менеджера и клиентов была одной и той же. Ещё ведь есть и инженеры-технари (они будут называться по-разному: садовники, учителя, конструкторы радиоэлектроники), но и с ними у менеджера хорошо бы, чтобы была общая цель. У менеджера булочной целевая система — булочки, а не булочная! А булочная? Она у менеджера — создатель целевой, её создадим и будем развивать как одну из систем в графе создателей. Но сначала — определимся, булочки там или бульдозеры. И договорим всех, что булочки, а не бульдозеры. Договорим всех — это сильнее, чем «договоримся со всеми», ибо если вы договоритесь с А и В, и они потом передоговорятся между собой, то ваша договорённость с ними нарушится. Если вы договорите не только себя с А и В, но и их между собой, то и ваши договорённости будут стабильней. По поводу целевой системы надо договорить всех в проекте, чтобы не было того, что половина людей в графе создания делает булочки, а вторая половина — бульдозеры, а информационные системы (включая AI) там поддерживают создание

[352] https://ru.wikipedia.org/wiki/Метод_критической_цепи

вертолётов (поэтому бульдозеры ещё хоть как-то получаются, а булочки — никак, и мало вообще понятно, что происходит и почему). Спросите, что именно (вот буквально: комплектация, характеристики, состояние в ходе работы) выпускает и с какой скоростью ваш проект — вы будете удивлены разнообразием ответа. Но каждый отвечающий будет уверен, что все остальные прекрасно понимают, что там за целевая система и какая скорость её выпуска! При этом понимаю, конечно, ровно так же, как они сами. Ваша задача — заранее знать, что понимание у всех разное, а дальше договорить всех по поводу этой целевой системы.

Целевая система определяется субъективно, «система в глазах смотрящего», поэтому вопрос определения «системы, которую вы хотите сделать» начинается с внимания к тому, кто такие «вы»: сколько вас, какие у вас интересы. Системное мышление — это мышление коллективной/групповой деятельности, с его помощью большие команды договариваются в проектах с целевыми системами на миллионы индивидуальных деталей каждый экземпляр, с сотнями и тысячами предприятий в графе создателей таких огромных систем. В разделении труда задание общей для проекта целевой системы важно, чтобы удержать внимание разных агентов в самых разных ролях на их общем важном (целевой системе!), чтобы без потери этого внимания к системному целому позволить заботиться о деталях подсистем отдельным членам каждой команды.

Что делать, если вы директор по развитию огромного холдинга, в котором сотня проектов по созданию и развитию сотни разных видов целевых систем, обширнейшая номенклатура продуктов и сервисов? Для начала надо понять, что разобраться разом со всеми этими целевыми системами не получится. Надо будет всё равно разбираться с каждой системой, не переобобщая. Иначе у вас получится какое-нибудь «оборудование» или «мастерство» — и невозможно будет осмысленно на этом уровне общности обсуждать хоть что-нибудь. Для начала надо разобраться с каждой целевой системой в каждом проекте, и только потом поглядеть на их общую систему-создателя, из частей которой будут составлены организации-создатели каждой из целевых систем — и обсудить, насколько общи получаются в своих частях графы создателей, насколько конкурируют за ресурсы проекты создания и развития отдельных целевых систем (и ещё проекты создания и развития систем из графа создателей, они ведь тоже берут ресурсы!).

Всё равно надо начинать с целевых систем, только не обобщая: брать проекты по одному. А если махнуть рукой и делать «просто холдинг»? Не получится! Если не знать, чем именно занимается холдинг, какая его стратегия (какими методами мы будем учить холдинг работать, какие целевые системы будут успешными, если действовать этими методами) — ничего не получится, кроме дикой путаницы и разбазаривания ресурсов на недоделанные проекты. Не надо лениться, надо разбираться с отдельными целевыми системами отдельных проектов, хотя и в этом деле можно задействовать разделение труда.

Для выявления целевой системы нет строгих правил, алгоритма, последовательности мыслительных шагов, порядка опроса каких-то людей, типологии случаев/cases (это предпринимательская/творческая/исследовательская задача! Нет алгоритма предпринимательства/творчества/исследования, есть только порождение догадок и отвод их критикой!). Так или иначе в решении по поводу выбора целевой системы будут участвовать самые разные агенты в самых разных ролях, пытающиеся удовлетворить самые разные интересы самыми разными методами работы, причём речь идёт не только о членах команды создателей

целевой системы, но и о внешних ролях. Если у вас проект, который вы можете сделать в одиночку, то вам не потребуется развёрнутого системного мышления, хватит его обрывков. Но если проект сложный и работает группа, то в принятии решений заведомо будет сложный переговорный процесс и о целевой системе придётся договариваться по ходу дела.

Обычно целевую систему нельзя определить, сидя на своём рабочем месте и размышляя: требуется не только читать самые разные документы и хорошо знать свою предметную область, но встречаться и разговаривать со многими людьми, а иногда и делать эксперименты, в том числе реализовывать макеты, прототипы. Иногда даже говорят, что пока не вылез из рабочего кресла и не пошёл общаться с разными агентами, системного мышления не происходит — ты ничего не будешь знать про целевую систему, ничего не будешь знать о том, по поводу чего все договорились, и договорились ли.

Не нужно путать:

- **«нашу систему»** (MySystem, OurSystem, engineered system «мою» личную, или «нашей маленькой команды») и
- **целевую систему**, которая общая для многих команд, и по поводу которой эти команды договариваются.

Без целевой системы нельзя разобраться и с «нашей системой»: для неё прежде всего должно быть понятно, где она находится: в подсистемах ли какого-то уровня целевой, в надсистемах целевой, в системах графа создателей. Если не разобраться, каким образом наша система даёт вклад в создание целевой системы — наша система не будет успешной, вы не сможете объяснить, почему команда-создатель вашей системы имеет какие-то вопросы к другим командам, создающим какие-то другие системы. А если есть целевая система — с кем бы вы ни разговаривали в проекте, вы члены одной команды создателей!

Да, если сделать «нашу систему» удовлетворяющей именно нашим командным интересам (или даже интересам непосредственного заказчика нашей системы — внешней проектной роли нашего подпроекта создания целевой системы) — мы будем счастливы, наши интересы учтены, и даже интересы заказчика нашей системы учтены, но успех всего проекта создания целевой системы в целом определяется главным образом успехом для внешних ролей проекта целевой системы, а не успехом для ролей нашей маленькой команды и даже внешних ролей нашей маленькой команды!

«Нашу систему» мы, конечно, сделаем, и потребуем оплаты за наш труд. Но деньги обычно платят за целевую систему: нет связи нашей системы с целевой — не будет денег за работу, проект нашей системы можно сразу не делать, а если мы ухитрились как-то договориться об оплате, то эту оплату найдут способ не сделать: если наша система никак не помогает создавать и развивать целевую систему, то как вообще можно за неё платить из денег общего проекта?

Похожее рассуждение будет и в тех случаях, когда речь идёт не об оплате деньгами, а преследовании каких-то других целей. Например, если делаешь какой-то политический проект и выполняешь в нём какую-то небольшую часть, инвестируя собственный трудовой ресурс и ожидая от реализации нашей системы и в конечном итоге целевой системы неденежных результатов.

Каждый проект уникален своей целевой системой и методами её

создания. Уникален не только вид целевой системы, уникальны и создатели в их графе, уникальны методы работы создателей и степень мастерства создателей в этих методах. Каждая ситуация в работе по каким-то методам с каким-то типом систем уникальна, так что ничего нельзя будет запомнить и применить потом в том виде, в котором запомнили — в одну реку нельзя войти дважды, одну и ту же систему нельзя сделать дважды одинаковым способом.

Даже если вы (как команда!) подряд второй раз делаете что-то очень похожее, то у вас по сравнению с первым разом появляется опыт, вы лучше знаете риски, чем это было в первый раз, имеете больше информации — и вы уже на основании этого опыта можете принять совсем другие решения, по-другому организовать работу, не повторить тех ошибок, которые были допущены в первый раз. Создание системы может задействовать какие-то одни методы работы, но развитие системы может подразумевать замену методов работы. Скажем, первый раз вы забивали гвозди микроскопом, но вот второй раз вы это вряд ли уже будете делать — вы учтёте неудачный опыт первого раза.

«Запомните метод, как мы это делали и всегда так делайте» — это не про системное мышление, системное мышление привлекает внимание к важным в мышлении объектам, но самого мышления не отменяет, и прежде всего мышления о системах в их окружении, а затем методах работы по созданию этих систем. Доставать голову из кармана и думать придётся в каждом проекте, для каждого создаваемого инкремента функциональности системы при развитии системы: системное мышление помогает ставить вопросы, но искать на них ответы, беседовать с самыми разными людьми, принимать решения после этих бесед всё равно придётся. Системное мышление даёт типы объектов для вашего мышления, но оно не мыслит за вас: что делать дальше с этими объектами — это всё одно остаётся вашему прикладному мастерству и вашему фундаментальному мастерству мышления (интеллекту, это его вопрос: что делать, когда неизвестно, что делать — каким методом работать, когда метод неизвестен).

И ещё раз напомним, что рассуждения по поводу достижения коллективных договорённостей по поводу целевой системы, «выявления системы» (как и все остальные рассуждения по проектированию системы) обычно делаются «в классах/типах», с типами систем, а не с индивидуальными системами-экземплярами/индивидами. Да, вы работаете в уникальной ситуации, но как бы ни были уникальны проекты и системы, всё равно делается попытка максимум знания перетаскивать из проекта в проект, накапливать знание, повторно использовать знания, акцентировать работу с паттернами, а не с индивидами. Даже если вы делаете что-то типа Эйфелевой башни, всё равно работать вы будете с типами, индивиды появятся только при изготовлении и сборке конструктивов разных системных уровней целевой системы.

Все системы в глазах самых разных смотрящих, так что потребуется

- самому делать догадки по поводу целевой системы (придумывать эти догадки самостоятельно, или беседовать с разными людьми, если вы недавно пришли в уже давно ведущийся проект). Это вы «выявите целевую систему» для себя самого, но это ещё не значит, что эта целевая система стала целевой системой всего проекта. Поэтому
- дальше вам надо договориться с командой и внешними проектными ролями

про целевую систему (а если вы организатор проекта, то не договориться с ними, а договорить их всех между собой — и заодно с вами), и про надсистему с системами в ближнем и дальнем окружении. Это будет рассмотрение целевой системы как «чёрного ящика», создание концепции использования.
- И только после этого обсудить подсистемы (целевая система как «прозрачный ящик», функциональная декомпозиция) и из чего их делать (концепция системы — какими конструктивами-аффордансами будете воплощать подсистемы).
- обсудить методы создания целевой системы, то есть способы изготовления, сборки, испытаний, пуска в эксплуатацию целевой системы, общий метод создания и развития (систему надо будет не только создать как MVP, но и развивать, выпускать много версий, может быть даже выпускать каждую версию мелкой серией или даже налаживать крупносерийное производство).
- И только после того, как методы создания целевой системы стали известны, можно обсудить, какие роли нужны для поддержки этих методов создания,
- и дальше уже обсуждать самые разные системы создания (включая команду как одну из систем создания!) по всему их графу, учитывая мастерство этих создателей в необходимых методах работы, а также наличие инструментария для этих методов,
- и ещё помнить при этом про «нашу систему» во всём множестве этих систем проекта.

Конечно, это мы опять в краткой форме изложили мантру системного мышления. Нужно только ровно так и рассуждать, во всех проектах, это и есть норма системного мышления: не только читать о том, что так надо делать, а брать — и делать. Не надо только читать, что обычно люди чистят зубы и принимают душ. Надо ещё и чистить зубы (хотя не хочется) и принимать душ (хотя лень). Вот с системным мышлением так же: лень, а надо. Вздыхайте — и думайте по системной мантре, пока не привыкнете делать это «на автомате», всё время, во всех проектах, подпроектах, разговорах с друзьями про их проекты, задаваясь этими вопросами при участии в беседах. Вот это — системное мышление!

Конечно, каждый участник проекта (а взаимосвязанных проектов тут может много обнаружиться, так что и участники других проектов тоже) имеет по поводу целевой системы проекта своё мнение, отличное от других. И участники проекта могут ещё и менять это своё мнение по поводу того, что является целевой системой проекта — прямо по ходу достижения между ними самых разных договорённостей (договорённости-то именно про это: про изменение мнений, было одно мнение — договорились о чём-то, это означает, что кто-то начальное мнение поменял).

Что делать в такой ситуации, когда мнения расходятся?! **Непрерывно договариваться, проверять сохранение договорённостей — и продолжать договариваться всё время проекта, а не только в начальный его момент. А если вы организатор — ещё и непрерывно договаривать всех вокруг себя.**

Вы к этому быстро придёте, когда поймёте, что любые ваши договорки будут нарушаться, если их участники в момент достижения ваших договорённостей ещё не договорились между собой. А потом они договорятся — и их договорённость окажется сильней договорённостей с вами, а когда вы договоритесь с ними по-новой, они опять передоговорятся — и опять ваши собственные договорённости с ними не устоят. Поэтому лучше бы вам самому их договорить — и достичь

коллективной договорённости, она будет более стабильной.

Системы создаются коллективно, в этих коллективах договариваются. Системное мышление помогает определить, по поводу чего договариваться. Прежде всего — по поводу того, что именно мы делаем: какая система является целевой, т.е. успешность какой системы будет ключевой для проекта. Нет договорённости по поводу целевой системы — коллективно не знаем, что делать, каждый делает что-то по своим представлениям. Добра в такой ситуации не будет.

Легко не распознать эту ситуацию, когда в речи и документах лишние генерализации. Если мы делаем «обувь», то это не означает, что договорились. Я считаю, что это женские босоножки 36 размера, соседи посчитали, что это мужские зимние ботинки — и разработали для них шнурки. Да, всё это «обувь», но при встрече этих уже разработанных шнурков с уже разработанным верхом босоножек у обеих команд будет сюрприз. А если подмётку «обуви» поручили делать третьей команде, то можно только догадываться, что им придёт в голову — валенки? Есть только один способ: специализировать в договорённостях целевую систему по максимуму и гарантировать, что результат договорённости по поводу целевой системы известен всем командам.

Почему так сложно определить систему? Потому как в культуре может ещё не существовать понимания удобного системного уровня, на котором определяется система.

Скажем, вы пытаетесь предложить новый способ создания стены из крупных блоков, но строят в этот момент только здания из кирпича (а понятия «стена» нет, только кирпичи). Мы приводили этот пример, когда говорили о системном мышлении, которое помогает бороться со сложностью: осмысленных операций с кирпичом в здании нет, но понятие «стена» может быть ещё не сформировано, поэтому с ним и не работают! Идея определить стену как систему, да ещё и дать её альтернативную конструкцию может быть очень, очень свежей!

Один из реальных кейсов: требовалось создать систему транспортировки самолётов с одного из аэродромов до завода-изготовителя, это нужно было для ремонта. Для этого где-то нужно было увеличить проём ворот, где-то добавить крепления на вагоне, где-то соорудить дополнительный пандус. Но все эти задачи решались как отдельные проекты, «по мере возникновения», у них не было общего финансирования (и деньги выделялись не сразу все, а на каждый проект отдельно, и нужно каждый раз было устно объяснять, зачем нужно потратить ту или иную сумму и выполнить ту или иную работу, общего списка работ не было). Не было понимания, что это всё нужно сделать, чтобы появилась «система транспортировки самолёта как целевая», не было в языке названия этой «системы транспортировки самолёта», не было проекта, не было менеджера проекта, был только человек, который был озабочен организацией транспортировки самолёта (хотя и проекта формально не было, и в голове этого человека не было «системы транспортировки самолёта», только где-то было желание ворота расширить, где-то крепление на прицепе переделать — но это не воспринималось как какие-то операции с «системой транспортировки самолёта»). Как только было осознано, что речь идёт просто о разных частях системы транспортировки самолёта, и если не сделать хоть один из этих проектов, то самолёты не будут проходить путь от аэродрома до завода, не будут ремонтироваться — всё волшебно наладилось, был назначен

менеджер проекта, выделено финансирование, нашлись внешние проектные роли. При этом целевой системой был именно самолёт в точке окончания транспортировки, а система транспортировки — «наша система», создатель «самолёта в точке назначения».

В человеческом движении уровень мышц и костей — это совсем не тот уровень, на котором мы можем говорить о том, как строить движения тела, в теле только мышц выделяют более семисот. Понятно, что при движении человек не управляет сотнями мышц, он удерживает во внимании другие объекты. А тело в целом — это слишком высокий системный уровень, двигаем-то частями тела, да ещё и воспринимаемыми изнутри. Пришлось найти пропущенный ранее системный уровень: **мышечные «ленты»** (часто говорят о «линиях», о «струнах», ленты мы говорим, чтобы легче было разговаривать о скрутках этих лент, а ещё «мышечность» их условна — там участвуют и мышцы, и связки, и фасции, и сухожилия, так что «опорно-двигательные ленты» было бы вернее), соразмерные человеческому телу, воспринимаемые изнутри как такой специальный функциональный орган, конструкцию которого выполняют в разных позах разные кости, мышцы, фасции, сухожилия. Этот функциональный орган присутствует в сознании движенца как один объект, с которым и работают. Объяснения того, как двигаться, становятся короче точно так же, как в случае присутствия стены между уровнями кирпичей и дома, присутствия транспортной системы между отдельными хитрыми приспособлениями на пути самолёта и «системы ремонта самолётов»::надсистема.

В одном из проектов менеджеры-основатели фирмы/предприниматели собирали по чуть-чуть железнодорожным транспортом отдельные фракции отходов нефтеперегонки в прибрежные огромные хранилища. Потом сливали эти фракции одной большой порцией/лотом (порция для торговли, для сделки, для обработки) в танкер, отправляли на завод (мелкие партии сырья завод не принимал), и там перерабатывали в весьма дорогой продукт. Какая была их целевая система? В чём был секрет их бизнеса? Они создавали **лот** (ибо меньше лота определённой величины заводы исходную фракцию нефти не принимали в переработку). Как только «лот» был осознан как целевая система, фирма запустила ещё несколько проектов по той же схеме. Без этого всё предприятие выглядело как случайная стратегическая/предпринимательская/визионерская находка, разовый проект. Конечно, лот упоминался во всех разговорах — целевые системы часто присутствуют в речи. Но дело двинулось только после того, как было осознано, что именно лот — целевая система, а не десяток других систем, которые мелькали в обсуждениях.

Целевая система в начале проекта находится вне внимания. Её упоминают обычно в первые пять минут разговора, иногда даже называют, но людям в голову не приходит, что они ищут именно её. Но когда эта система находится (на неё обращают внимание, именуют и вводят в проект явно как целевую систему, по поводу которой все договариваются), жизнь становится много, много проще.

Ещё раз: никаких алгоритмов, рецептов, лайфхаков по определению целевой системы нет, это исследовательские догадки/«предпринимательские гипотезы»/негарантированное визионерство, творчество, что «по алгоритму», «по рецепту», «по лайфхаку», «по процедуре» или ещё как-то «по правилам» невозможно. Наоборот, вместо помощи в догадках системное мышление даёт множество способов критики предлагаемых догадок по поводу того, что будет целевой системой в проекте, и реализовывать нужно (понимая, что это всё только

гипотеза/догадка) лучшую из выдержавших критику. Это подробно рассматривается в рамках методов исследований (на базе лучших объяснительных теорий того, как они устроены, сегодня это развитие идей попперианской эволюционной эпистемологии) и рациональности (на базе лучших сегодняшних объяснительных теорий о том, как строить действия в условиях неопределённого будущего).

Тем самым вы должны высказывать на базе вашей интуиции (возможно, коллективной! Хорошо организованные совместные рассуждения многих агентов, да ещё поддержанные компьютерным моделированием!) догадки/гипотезы о целевой системе и далее разворачивать от этих догадок системы создания и критиковать эти догадки, прогоняя их через многочисленные эвристики, которые приведены в наших руководствах по рациональной работе, системному мышлению, методологии, системной инженерии, руководствах по прикладным инженериям (например, по инженерии личности, системному менеджменту). Вспоминаем чеклист для проверки догадок по поводу систем из предыдущего раздела. Этот чеклист можно продолжать и продолжать, используя все подсказки по критике и поиску ошибок в договорённостях по поводу систем из нашего руководства.

Целевая система добавит ещё несколько пунктов. Например, несоблюдение вот этого пункта делает проект «дохлым»:

- **совокупная стоимость владения системой должна быть меньше оценки пользы от её использования/эксплуатации.**

Но если всё это оказалось соблюдено, то вы (вероятно! Это прогноз/гипотеза/догадка! Гарантировать ничего нельзя!) нашли целевую систему — и поэтому должны взять эту догадку всерьёз, то есть строить проект создания и развития системы исходя из этой догадки.

Моделирование: целевые системы

Заполните табличку по целевым системам от трёх рабочих проектов, в которых вы участвуете. Не забудьте проверить эти целевые системы по чеклисту.

Номер п/п	Название проекта	Целевая система	Функция/назначение	Надсистема	Сколько проверок (на физичность, на рассмотрение во время эксплуатации, на учёт графа создателей и т.д.) вы сделали при определении	На сколько процентов вы уверены в вашей гипотезе о целевой системе	Какой процент участников проекта будет с вами согласен по поводу целевой системы (как называется, что входит в целевую систему, что

					целевой системы		остаётся в окружении)

Системы-продукты и сервисы/методы провайдеров/создателей

Целевая система раньше часто мыслилась как поставляемый товар (какая-то вещь, изготовленная для продажи): **продукт/изделие**::система::«физический объект», в разных отраслях это может иметь разные названия, например, в строительстве — здание или сооружение, которые не принято называть продуктами или изделиями, но по сути — это то же самое**.**

Продукт/изделие изготавливается его создателем обычно из закупаемого этой командой сырья, а затем физически поставляется его конечному потребителю для использования, условно «пересекает границу предприятия», меняет собственность. И уже в собственности у потребителя этот продукт/изделие выполняет свою роль/функцию, работая в составе надсистемы в ходе эксплуатации. После перехода собственности потребитель продукта/изделия::система использует его двумя способами:

- в составе уже своей целевой системы как конструктив-аффорданс (например, ваши часы-продукт ваш клиент устанавливает в свой автомобиль-продукт, а затем продаёт автомобиль как продукт уже своему клиенту, и часы показывают время в ходе эксплуатации автомобиля)
- в составе систем из графа создателей его продукта (например, ваши часы-продукт он устанавливает в свой цех на стену, чтобы рабочие могли следить за временем в ходе сборки ими автомобиля как уже его продукта, а затем продаёт уже своему клиенту автомобиль без ваших часов или с какими-то другими, не вашими часами. Ваши часы оказываются эксплуатируемыми в составе цеха, системе из графа создателей автомобиля).

Альтернативный вариант — считать сразу, что целевая система — это будет продукт не у вас, а у клиента, который пришёл к вам со своим сырьём — недоделанным продуктом, своей целевой системой, **предметом сервиса/метода**. Работа ваша идёт не с вашим сырьём и у вас нет передачи вашего продукта заказчику, вы как создатель продукта своего клиента только работаете по методу-сервису, используя своё мастерство и свой инструментарий этого метода (сервисной, обслуживающей) работы. Работа идёт над сырьём заказчика ваших работ и продукт остаётся у заказчика ваших работ, в его собственности. Вы лишь меняете состояние каких-то систем, которые принадлежат на этот момент не вам и потом войдут в состав или продукта заказчика, или в состав инструментария заказчика (хотя тут надо быть аккуратней: если вы работали над обучением сотрудников заказчика, то рассуждение сохраняется, только мастерство сотрудников трудно считать инструментарием, да и как «сырьё» они очень условно принадлежат вашему заказчику). В принципе, рассуждение тут чуть сложней, потому как сервисы и методы работы, конечно, по цепочке отслеживаются вплоть до какой-то системы, но в жизни часто может встретиться и какая-то работа с информацией (описание системы — проектирование «железной» системы, обработка данных эксплуатации

и т.д.). Большинство рассуждений про методы и сервисы работает и тут, это необязательно именно изменение систем, но обязательно, что все эти изменения самых разных объектов как-то отслеживаются по всей (возможно, очень длинной) цепочке изменений до целевой системы. И если отследить не удаётся, то ставить выполнение этих работ под вопрос, может их и не надо делать, ничего ведь к лучшему не изменится. Контрольный вопрос: «а если работ этого метода/сервиса не делать, что будет с целевой системой?». Если это работы с создателем, то дальше по цепочке: «а потом этот создатель что лучше сделает для целевой системы»? Если рассуждение не дотягивается до целевой системы по какой-то цепочке причин-следствий, то всё плохо: вы, похоже, встретили маскировку никому не нужной работы (хотя, конечно, агенту, которому заплатят за ненужную для создания целевой системы работу, эта работа очень нужна — он хочет за неё денег, или хочет потренировать своё мастерство, или ещё хочет что-то локальное «для себя», но не глобальное — для всего проекта создания целевой системы).

Даже со всеми нюансами ситуация выполнения сервисных работ сводится к тому, что никакого продукта/изделия у вас в проекте не будет, но вы просто примете участие в работах создателей какой-то целевой системы по какому-то методу, и даже не непосредственно создателей, а каких-то работах создателей из графа создателей, достаточно далеко от целевой системы. Метод, которым вы меняете состояние чужих систем (чужой целевой системы, чужих конструктивов, которые потом составят чужую целевую систему, или даже чужого инструментария в графе создания), называется **сервис/service/услуга.**

Ваши тут будут только работы/поведение/изменения по вашему методу (то есть работы вашего мастерства и инструментария по знаниям/теориям/алгоритмам метода) вас как создателя над не принадлежащей вам целевой системой, изменение её состояния. Вы будете тут только системой создания, ваша команда будет предоставлять клиенту только внешнее поведение — работы сервиса.

В литературе (в том числе англоязычной, там та же ситуация, что и в русском языке) можно найти много разных мнений, что такое «сервис»[353], основные из них:

- Сервис/провайдер, роль агента, выполняющего работы над предметом метода работ — например, автосервис как провайдер работ по методам ремонта автомобилей, микросервис — софт как провайдер работ по методам обработки данных (редкое использование «сервиса» для неодушевлённых систем).
- Сервис/работа по методу над чьими-то предметами, например «получил сервис для клиентского автомобиля» (выполнены работы не указанного метода сервиса для автомобиля::предмет клиента).
- Сервис/метод выполнения работ над чьими-то предметами, например, «выполнил сервис клиентского автомобиля» (выполнил неспецифицированный метод работ, то есть работу по методу для клиентского автомобиля).

По-русски «службой» называют и service::процесс (метод, разворачивающийся во времени паттерн действия, наблюдаемый извне паттерн работы) «оказания услуги»::работа (слова оказание/проведение/выполнение тут указывает на то, что метод ещё должен быть реализован работой), и ту систему, которая вызывает это поведение, т.е. server/слугу/provider. И это иногда функциональный «слуга» (роль),

[353] https://yadi.sk/d/4hIEcpcn3Ny9iN

а иногда конструктивный (агент). Изредка «сервисом» называют ещё и интерфейс между «слугой» и изменяемым этим слугой внешним миром.

В нашем руководстве мы придерживаемся мнения, что сервис — это метод выполнения работ (различаем сервис как метод работ, общий для всех работ какого-то сервера/роли и работы сервиса/метода), а роль, выполняющая метод/сервис, называется server/«service provider»/сервер/служба/провайдер. Тем самым есть и синонимия роль/провайдер/служба, но там те же проблемы, что и с роль/stakeholder (то ли оргроль::«функциональный объект», то ли оргзвено-в-роли::«конструктивный объект», примером тут будет — «то ли принц Гамлет, то ли Вася Пупкин»).

Мы всё-таки придерживаемся мнения, что провайдер — это оргроль (сервер всё-таки чаще используется, когда речь идёт об инструментах какого-то агента, хотя в русском языке сервер часто переводят как «служба», и тогда это тоже может быть оргролью, а не ролью инструмента).

В проектах провайдер — это обычно условно-внешняя проектная роль. Условно-внешняя — внешняя, ибо работает «чужой над нашим материалом», а формально — внутренняя/командная, ибо в нашем графе создателей, работает в составе общей расширенной команды (состоящей из множества отдельных команд в большом графе создателей) над общей целевой системой. Если речь идёт о провайдере как роли, можно назначать сервис-провайдером какого-то сервиса/метода одно или другое оргзвено в ходе оргпроектирования.

Тем самым сервис пополняет синонимический ряд метод/способ/практика/стратегия/культура/стиль/сервис, причём используется главным образом тогда, когда речь идёт о работах создания над чужим материалом. Сервис — это метод для работ создания (самых разных, например, измерения/описания, изменения в физическом мире, предсказания/проектирование), проводимых над чем-то внешним, «чужим», «на давальческом сырье». Если речь идёт о сервисе «забивание гвоздя», то речь идёт том, чтобы взять оргзвено, исполняющее роль провайдера/службы и заставить его выполнить работы его метода/сервиса (работы забивания::метод/сервис) для нашего гвоздя. Если сами будем забивать гвоздь, то всё то же самое, только не будем использовать терминологию сервиса и провайдера/службы.

Конструктивный объект (оргзвено) как функциональный объект (служба/провайдер/роль) выполняет работы сервиса/метода над предметом метода. Наш плотник забивает наш гвоздь: «забивание гвоздя»::«метод работы для изменения состояния гвоздя», плотник::роль. Чужой плотник забивает наш гвоздь или наш плотник забивает чужой гвоздь: «забивание гвоздя»::«сервис/метод для изменения состояния гвоздя», плотник::провайдер/роль. Всё то же самое, слова другие и в словах — отсылка на «внешнесть» (при этом можно долго ещё обсуждать, что там «наше», а что «не наше», об этом надо будет коллективно договориться. Собственность в экономике определяется не как то, что вы считаете своим, как то, что считают вашим окружающие вас люди, с чем они готовы согласиться).

Для неодушевлённых создателей (инструментарий: станки, измерительные инструменты/instruments, инструменты/tools, гаджеты, оборудование и аппаратура) всё то же самое: молоток просто обзовут не провайдером сервиса (но так могут обозвать владельца молотка), а сервером (при этом даже «сервер сервиса» будет

звучать как «масло масляное»: «работник работ», «служение службы»). Молоток::сервер/инструмент::создатель в роли «забивала» выполняет сервис «забивание»::метод::поведение над меняющим своё состояние объектом «гвоздь».

Мы не очень различаем в рассуждениях про системы в графах создателей (если речь не идёт о проектных ролях, играемых агентами, вырабатывающими свои стратегии по достижению своих целей — молоток не может играть трудовую/проектную роль! Молоток не актёр, его не принимают на работу!) живые и неживые системы, сервером/службой/провайдером/слугой/server у нас может быть и предприятие, и человек, и просто какой-то инструмент, разница тут больше терминологическая. Их метод работы как создателя по изменению чего-то не их собственного — сервис.

Мы уже встречались с ситуацией, когда системы классифицировались на основании их принадлежности каким-то владельцам, это системы систем (SoS). С сервисами то же самое: речь идёт просто о работах каких-то создателей, только не над собственным продуктом или собственным создателем в графе создателей собственного продукта (например, инструментарием для реализации какого-то метода), а над чужим продуктом или чужим создателем, например, чужим инструментарием. Если вы пришли поточить кому-то ножи на его кухне, и это не ваши ножи и не ваша кухня — то это вы пришли «оказать сервис», «выполнить работы сервиса»/«выполнить работы по методу».

Сервис «забивание гвоздя» может оказаться функцией/методом/способом в составе более крупного метода, работающего над более крупной ситуацией: скрепления досок, подготовки крепления для картины на стене, устранения царапающего гвоздя и т.д.

Так что не удивляйтесь, когда про один и тот же предмет, один и тот же продукт и его одно и то же поведение вы услышите два совсем разных разговора по части терминов: это специфика интересов. Сервис — это обычно про поведение создателя, метод его работы над «не своим» предметом, который должен поменять состояние в ходе работ сервиса. Сервис при этом — метод (забить гвоздь), а работы — задействования/оказания сервиса (забить десять гвоздей — это десять работ по методу «забить гвоздь»). Как и в случае метода, различаем оказание сервиса (выполнение метода) и сам сервис (метод). Бытовые выражения типа «предоставить сервис» по факту означают «предоставить сервер, который выполнит работу сервиса» (то есть предоставить оргзвено или инструмент, который исполнит работы, находясь в роли исполнителя работ по методу).

Внимательный инженер-менеджер уже заметил, что мы тут просто тренируем его нейронную сетку десятком повторений чуть разными словами одного и того же: агент, находясь в роли, выполняет работы по методу, меняя состояние предметов данного метода. Разница только в принадлежности предметов, меняющих состояние в ходе проведения работ по методу: для разной принадлежности (своё или чужое) терминология разительно отличается, слова все разные, а суть дела — не меняется. И в случае «чужого» целевая система для провайдера — вот этот самый предмет его сервиса. Для плотника и даже молотка целевую систему искать нужно там, где гвозди, даже если они не свои, а тебе их дали для забивания! Если вас позвали покрасить чей-то забор, и даже дали краску, то можно дальше выбирать — целевой системой является слой краски в правильных местах, или забор в целом, переведённый из состояния «некрасивый» в «красивый» видом

методом/сервисом «покраска» (другой вариант методов/сервисов — «перестройка», «драпировка» и даже «ликвидация», ибо нет забора — нет проблемы его красоты).

В этом и сила системного мышления, сила мета-мета-модели: рассуждения с объектами внимания проводятся «в понятиях» на уровень абстракции выше, чем уровень рассказа о предметных областях.

Пример парикмахерской как сервис-провайдера

Давайте сделаем ход на конкретизацию, рассмотрим пример чуть сложней, чем плотник с молотком, забивающий чужой гвоздь. Парикмахерская (роль/провайдер, реализованная оргзвеном ООО «Чикчик») оказывает услуги/сервис по стрижке волос. Слово «сервис/услуги» указывает, что волосы там будут внешние, чужие, приносимые прямо на голове, «материал заказчика услуги». Рассуждения не меняются — идёт ли стрижка волос собственной головы или чужой головы, меняется только терминология, про свою голову говорят «провести работы стрижки::метод для волос», а про чужую — «оказать сервис стрижки волос». В «оказать сервис стрижки» и слово «сервис» тут как тип для «стрижки», наподобие того, как иногда для пущей убедительности говорят «система самолёт», а не просто «самолёт». Можно было бы сказать и короче, не упоминая тип «сервис» — «провести стрижку», и сразу становится неважным, это чужая голова или своя, язык методов и ролей более универсальный, чем язык сервисов/услуг и серверов/служб.

Дальше мы рассмотрим небольшой чеклист по важным объектам ситуации сервиса. Мы даём сначала тип более абстрактной мета-мета-модели (например, «сервис»), а потом указываем конкретизацию/специализацию этого типа мета-модели для мета-модели предметной области стрижки волос (например, «стрижка и укладка» для предмета сервиса «волосы», метаУ-модель, «из учебника парикмахерского дела», учебника метода, или даже метаС-модели, «из ситуации этой парикмахерской», из регламента работ по стрижке и укладке для этой парикмахерской).

Это мы так пытаемся показать, что сами объекты рассуждения задаются их типами мета-мета-модели мира («картина мира», upper ontology) из наших руководств, а потом под эти типы находим объекты в жизни/«рабочем проекте», а не берём первые попавшиеся объекты в жизни и пытаемся потом для них найти типы из руководств. Моделирование идёт не так, что вы смотрите на понравившийся вам объект и подыскиваете к нему тип, а наоборот: вы имеете уже абстрактный тип и подыскиваете к нему объект. Скажем, берёте тип «целевая система» и выявляете, что это такое у вас в проекте.

Зачем этот чеклист нужен, зачем его проходить в проекте? Чтобы структурированно обсуждать проект, не пропуская важного. В случае парикмахерской это легко, все знают эту предметную область. Но в случае новых видов сервиса всё не так очевидно, и нужно думать. Типы мета-мета-модели помогают не пропустить какие-то важные объекты в рассуждениях о предметной области, то есть мета-модели. А рассуждения по предметной области (мета-модели) помогают потом не пропустить важные объекты модели конкретной жизненной ситуации в конкретном рабочем проекте, в физическом пространстве-времени. Вот этот чеклист, и надо помнить, что это просто начало мышления, дальше системное мышление включает много больше понятий, заставляет подумать о много большем числе важных объектов:

- **Целевая система:** причёска (ролевой/функциональный объект).

Получается из волос (в данном случае — волос клиента, прямо на голове клиента, это важно для использования формулировок «сервиса») путём выполнения работ по методам стрижки и укладки. Другая формулировка: причёска получается путём оказания услуг/service парикмахерской для волос клиента. Ещё одна формулировка: парикмахерская выполняет сервис стрижки и укладки для волос клиента, на выходе имеем причёску.

- **Сервис:** стрижка и укладка волос (причём волос клиентских — тип «сервис» говорит, что предмет сервиса — чужой). Это методы/функции/способы работы парикмахерской как роли агента-оргзвена.
- **Предмет сервиса/метода:** сырьё/заготовка — волосы, целевая система/продукт — причёска. Можно обсуждать смену состояния волос в ходе создания («не стрижены», «стрижены и не уложены», «стрижены и уложены»/причёсаны) или причёски в целом («не имеет формы», «стрижка есть, форма после мытья головы важна как бытовая причёска», «целевая причёска есть»/«стриженые волосы уложены»). Неплохо бы договориться. Акцент на волосах даст подробности в работе парикмахерской, акцент на причёске — будет меньше проблем с тем, что «все работы с волосами выполнены, а причёска почему-то получилась плохая». Тут хитрость: парикмахерская делает стрижку так, чтобы волосы без укладки смотрелись как причёска (а уж после укладки вообще должны производить неотразимое впечатление). Если мастерства стрижки у парикмахера нет, стрижка в результате приводит волосы в такое состояние, что без укладки смотреть на причёску нельзя, «ужас-ужас». Ещё надо отметить, что **целевая система обычно физична, но если у вас сервис/метод на выходе даёт описание, то тут будет описание как предмет сервиса/метода, а целевая система будет отличаться.** Скажем, интернет-сервис подбора причёски будет иметь предметом эскиз причёски, а целевой системой — таки саму причёску. В любом случае работы по методу (в том числе работы сервиса) меняют состояния предметов предметной области метода/сервиса, и нас интересует конечное состояние предмета сервиса, того предмета, ради достижения ожидаемого состояния которого ведутся работы.
- **Надсистема в окружении целевой системы:** ближайшее окружение причёски, как ролевой/функциональный объект: «голова для свидания» (но не «голова для еды в неё», например, или «голова для занятий спортом». Определение объекта по назначению/функции/целевому поведению). Тут важно брать системное окружение как можно ближе к целевой системе: берём голову клиента в какой-то ситуации (над-надсистеме — свидании), а не всё тело клиента. Тут «ближе» — ближе по системным уровням, но помним, что системные уровни непосредственно связаны с близостью в пространстве-времени (ибо отношение композиции/«частей-целых», которым связаны системы разных системных уровней работает как раз над частями и целыми областей пространства-времени).
- **Потребность** (клиента как внешней оргроли, владельца головы с волосами на ней): «впечатлить партнёра на свидании красотой головы» (фраза описывает надсистему для целевой системы проекта стрижки-укладки. «Красивая голова» — инструментарий клиента для его целевой системы «впечатлённый партнёр». Причёска — часть/подсистема инструментария «красивая голова»::система, вторая часть будет — макияж::система (не путаем с «нанесением макияжа»::метод или даже работы по какому-то

методу нанесения с предметом этого метода «макияж»). Нанесением макияжа (изготовлением красивого лица для впечатления партнёра на свидании) клиент хочет заняться сам. Мы (персонал парикмахерской::служба/роль, «мы» обсуждаем этот проект — отслеживайте всегда состав случайно оброненного «мы» — кто это, сколько их, чем они заняты?!) не обсуждаем, хочет ли клиент впечатлить партнёра не только красотой головы, но и общей красотой всего драпированного одеждой тела.

- **Концепция использования:** сценарий подразумевает переход партнёра в состояние «восхищён» после демонстрации партнёру по свиданию головы с причёской. А также присутствуют самые разные другие сценарии (дождь и ветер по пути на свидание, сон перед свиданием, что будет с причёской после душа, когда пропадёт укладка и останется стрижка и т.д.). Мы, конечно, учитываем по возможности более и более дальнее окружение — сценарии, включающие голову с макияжем, всё тело с его драпировкой, окружающую ситуацию, например, помещение или открытая площадка с дождём и ветром. Наша целевая система должна хорошо работать в широком разнообразии ситуаций, и эти ситуации надо учесть — часть таких ситуаций можно узнать от клиента, но часть таких ситуаций учитывается из знаний/теорий метода, клиент может и не знать, что его от чего-то защитили, удовлетворили потребность, о которой он и не знал, и которая у него появится, если он вдруг попадёт в какой-то вариант сценария использования. Конечно, если учёт всё большего и большего числа ситуаций/сценариев использования в концепции использования ведёт к росту цены системы или удобства использования, то надо будет договориться с клиентом по этому поводу.
- **Архитектурная характеристика** выживаемости/survivability причёски как целевой системы парикмахерской: волосы с укладкой должны выглядеть сногсшибательно, а без укладки (если пришлось принять душ в ходе свидания) приемлемо.
- **Создатель целевой системы**: система, оказывающая сервис. Парикмахерская::провайдер/служба/server/provider/роль — это и есть создатель причёски::«целевая система». То, что внутри парикмахерской есть парикмахер, который и создаёт причёску, пользуясь своим мастерством и инструментарием — это нам сейчас неважно, у нас рассмотрение чёрного ящика, и мы не учитываем внутреннее разделение труда в парикмахерской, там и роботы могут работать. Создатель::роль — это нормально, это просто функциональное рассмотрение создателя (а не конструктивное, оргзвеном тут будет ООО «Чикчик»). Сервис (стрижка и укладка волос) — это не сама парикмахерская и даже не её работы. Сервис (стрижка и укладка волос) — это метод работы, экземпляров работ для экземпляров волос по этому методу будет много. Сервис — это поведение парикмахерской, описанное как содержание/метод работ, а не сама парикмахерская.
- **Пользователь целевой системы**: «клиент парикмахерской»/«причёсываемое лицо» (тут, похоже, нет специальных названий типа «игрок» или «сталевар», поэтому берём общие термины типов мета-мета-модели «клиент» или «лицо» и добавляем предметно-специфичные слова роли «парикмахерская» или указываем на объектность в предметно-специфическом действии — «причёсывамое»).

Сервис тут «стрижка и укладка» (можно рассматривать и как два отдельных сервиса, «стрижка» и «укладка»), обратите внимание на **отглагольное**

существительное (в английском это «-ing» форма слова»), это обычно для функций/методов.

Сложно? В первый раз — да, сложно. Потом будет проще и проще. И всё больше и больше ситуаций сервиса вы будете трактовать как подходящие под этот паттерн «парикмахерская» (пример был взят только для того, чтобы уж точно была «служба/провайдер», а не «товарное производство» — и примерно было бы понятно, о чём речь. Но мозг затем использует этот пример, рассуждая в рамках прототипной теории понятий — «все сервисы как-то похожи на парикмахерскую»).

Зачем так сложно, почему нельзя порассуждать про парикмахерскую, не поминая все эти типы мета-мета-модели из нашего руководства? Затем, что во внимании при обсуждении сервисов (да и просто любых методов работ) в самых разных ситуациях оказываются не случайные объекты, а главные/важные объекты ситуации. Относитесь к этому как фрагменту чеклиста тех объектов, которые надо обсудить, чтобы сорганизовать множество самых разных людей (создателей парикмахерской, сотрудников парикмахерской, клиентов парикмахерской) для коллективной работы.

Конечно, типов мета-мета-модели, которые указывают на вещи первостепенной важности для их коллективного понимания в проекте, много больше — в нашем руководстве их тоже много больше, этот чеклист тут выдан только как пример системных рассуждений. Но в жизни царит суета, суматоха, пожары-авралы, бессонница и затуманенное сознание от просто усталости. Важные объекты, которые берутся из понятий интеллект-стека в целом (и системного мышления в частности) помогают удержать рассеянное внимание на важном.

Например, в этом чеклисте есть потребности/интересы/предпочтения внешних (для проекта парикмахерской) проектных ролей — для чего клиент хочет обратиться к сервису. В чём счастье внешних ролей, а не счастье внутренних ролей команды (у которых, конечно, тоже есть интересы и предпочтения, хотя их редко называют «потребности»). Если причёска будет, но голова окажется после выполнения причёски не в порядке (например, в ходе стрижки отрежутся не только волосы, но и уши), то беда. Если причёска будет для занятий спортом, а не для похода на свидание — беда. Всегда обсуждаем/описываем в паре целевую систему (причёску, тут это только пример) и надсистему (голову, тут это только пример). Описание надсистемы — потребности внешних проектных ролей, описание целевой системы — концепция использования. **Концепция использования как-то прослеживает/трассирует описание функциональности системы к потребностям/интересам внешних проектных ролей. Хотите или не хотите, а концепцию использования вам нужно иметь.**

Выбранный клиентом сервис парикмахерской будет «стрижка» или «укладка» или оба вместе. Выбор метода — это стратегирование, для получения результата «впечатлённый партнёр» клиент выбирает метод/сервис «создание причёски» и обращается за работами по этому методу к провайдеру сервиса, организации в роли «парикмахерская». У парикмахерской есть мастерство выполнения работ по методу (у сотрудников парикмахерской — будь они люди или роботы) и инструментарий (зеркала, кресла, ножницы и машинки для стрижки, расчёски, шампуни и лаки — всё, нужное для «оказания сервиса», то есть выполнения работ по набору методов создания причёски, то есть методам стрижки и укладки).

Работа сервис-провайдера (работы сервиса, ибо можно говорить и «работы роли», и «работы по методу») занимает конкретное время, использует конкретное

оборудование и ресурсы, изменяет конкретную голову с конкретными волосами. А если говорить про функциональный аспект «как это обычно делается, как это описывается в учебниках», то сервис — метод/практика/культура работы, а не сама работа. Могло бы быть «мытьё головы и укладка у подруги», «закрытие головы шляпой дома, чтобы волос не было видно» и т.д. **Сервис выбирают в ходе стратегирования, дальше он выполняется как стратегия, для выбранного сервиса планируются работы, то есть планируется задейсвование конкретных экземпляров ресурсов.** Планирование идёт после выбора стратегии, то есть планирование идёт после выбора метода достижения результата «впечатлить партнёра», в том числе планирование включает выбор самостоятельного выполнения работ по методу или покупку этих работ у провайдера (покупку оказания услуг, предоставления сервиса). Нет выбора из нескольких методов получения результата (не было стратегирования — метода выбора метода) — вы мало подумали (подробней об этом будет в руководствах по методологии, системной инженерии, системного менеджмента). Но для выбора надо различать:

- Работу по достижению результата — она может быть проведена разными методами/способами, задействовать разные ресурсы. Дальше работает рациональность как метод мышления из интеллект-стека. Рациональность задаёт способ/метод порождения альтернатив и далее использование теории принятия решений, в которой вы выбираете лучшее из альтернативных решений и дальше действуете в соответствии с вашим выбором. В экономике, когда говорят об альтернативных издержках, то имеют ввиду ровно это: вы можете получить результат разными методами/способами, эти способы конкурируют друг с другом, но в конце концов будет задействован какой-то из них (самый выгодный по ресурсам), и это задействование метода/способа и будет работой. Работа с разными ресурсами может быть проведена много раз по одному и тому же способу. Гвоздь вы можете забивать молотком, камнем, микроскопом, но даже перед этим можете выбрать — забивать гвоздь, или ввинчивать шуруп или склеивать клеем.
- Метод/способ/культура проведения работ. По одному методу можно выполнить множество экземпляров работ, то есть множество раз задействовать ресурсы для получения результатов (изменения состояния предметов метода) выбранным методом. Если вы забиваете гвозди микроскопом, то этим методом можно забить множество гвоздей.

Голова (надсистема целевой) и волосы на голове (подсистема целевой) принадлежат клиенту, парикмахерская — создатель, реализованный организацией со своим мастерством и инструментарием (включая помещение, но возможна и работа «на выезде»). Парикмахерская — это роль создателя, которая может выполнять работы по типовому набору методов работы с волосами клиента: меняет состояние волос для создания причёски, причём может и добавлять к волосам свои какие-то материалы как части целевой причёски (ленточки, шиньоны, лак и т.д.).

Что является целевой системой? Причёска! А если мы организуем нашу парикмахерскую? По-прежнему — целевая система причёска, а парикмахерская тогда — «наша система»! С парикмахерской как системой-создателем мы выполняем полностью всё системное мышление, но она находится в другом системном разбиении, нежели причёска — парикмахерская относится к разбиению систем создания, которые рассматриваем во время/realm создания причёски.

Окружение причёски мы рассматриваем во время/realm эксплуатации/использования/функционирования/работы самой причёски, а окружение парикмахерской — во время работы парикмахерской (но не во время создания самой парикмахерской!). Отношение между парикмахерской::система и причёской::система — создание::отношение. По этому отношению выстраивается граф создателей, но по нему нельзя выстроить какие-то «системные уровни», ибо иерархия системных уровней выстраивается по отношению композиции, которое применяется на момент использования. В частности, сама парикмахерская как система может быть реализована очень по-разному: это может быть «частный парикмахер» с чемоданчиком инструментов, который выезжает к клиентам (причём клиенты любят вызывать таких парикмахеров ночью, этого не сделаешь с другим типом парикмахерской — «посетить парикмахерскую» придётся в других вариантах самим, она не приезжает к тебе), это может быть подразделение большого холдинга, это может быть семейное предприятие в одной из комнат частной квартиры. Парикмахерская (она тут служба/провайдер) может быть частью парикмахерского холдинга, городского квартала, у неё такие части как парикмахерское кресло, зеркало и ножницы. Когда речь идёт о сервисе, то мы обычно проектируем и изготавливаем нашу систему-службу/провайдера/сервер в составе систем создания, но выстраиваем всё мышление всё равно, думая о целевой системе, в которой сервис производит изменения, чтобы его предметы поменяли входное состояние на выходное. Во всех этих случаях разной организации парикмахерской (роль создателя как провайдера сервиса по стрижке и укладке чужих волос могут выполнят разные агенты) обсуждение того, что там происходит при выполнении методов стрижки и укладки — одно и то же. Это экономия мышления: ситуация может меняться, а способы мышления остаются стабильными и даже сохраняется часть результатов мышления, общая для многих ситуаций.

Мы будем создавать не только целевые системы, но и системы создания целевых систем — поэтому говорим также о графах создателей, а не только отдельных системах создания. Сами волосы себя не подстригут и не уложат в причёску, они не живые (это только бабочка сама себя создаст из куколки, куколка себя создаст из личинки, личинка — из яиц). Для проведения работ сервиса стрижки и укладки поэтому создаём парикмахерскую с её мастерством и инструментарием стрижки и укладки, а для этого создаём создателя парикмахерской (это может быть строительно-монтажная организация, специализирующаяся на создании парикмахерских. Но такая организация, понятно, сама не занимается стрижками, как не занимается стрижками и клиент парикмахерской — и вот мы вновь коснулись тематики разделения труда, специализации разных агентов на разных методах работы, на разных сервисах).

Если бы нужно было что-то делать с ракетами, а не с волосами клиента, то мы бы делали ракетное конструкторское бюро и завод ракет, а не парикмахерскую. Если бы нужно было что-то делать с мастерством, то открыли бы школу, или вуз, или курсы или программу наставничества. Сначала понимаем, что делаем (целевую систему в её окружении), затем понимаем, какими способами её можно делать (методы/сервисы), затем думаем о том, какой агент-создатель сможет своим мастерством и инструментарием выполнить работы этого метода/сервиса. Поставщиков/провайдеров работ по каждому сервису много, и они конкурируют. Но мы в том числе создаём этих создателей, если их ещё нет (иногда создавать приходится обширный граф создателей, которые создают, развивают и изменяют друг друга). Интеллектуальных агентов для этого учим мастерству (с учётом

разделения труда), снабжаем инструментарием (капитал) и организуем (делаем известным кто кому что может поручить сделать, кто чем будет заниматься).

В случае с созданием парикмахерской выбор тех или иных решений по оказанию сервиса и устройству парикмахерской в зависимости от варианта метода, лежащего в основе этого сервиса, отслеживается/трассируется и к потребностям надсистемы нашей системы (например, потребностям владельцев парикмахерской), и к потребностям владельцев целевой системы (потребностям владельцев волос, которым делают стрижку).

Описание системы делается и для причёски (целевой системы), и для парикмахерской (нашей системы). Воплощение системы будет и у парикмахерской (нашей системы), и у причёски (целевой системы). Внешние проектные роли с их предметами интереса и предпочтениями будут и у проекта создания парикмахерской, и у проекта создания причёски — роли будут разные, интересы их будут разные (даже при одинаковых предметах интереса!), забыть никого нельзя, ибо это связано и с успешностью причёски, и с успешностью парикмахерской.

Но целевая система будет — причёска, ибо только упустишь из виду, что делать в конечном итоге нужно причёску (а не «нашу» парикмахерскую, она тут только провайдер сервиса), парикмахерская будет отлично работать, только недолго, пока не кончатся деньги инвестора, ибо вся «клиентоориентированность» или «продуктоориентированность» будут утеряны.

Платят обычно за результаты работ создателей — за оказание сервиса, а не за создание создателя. Платят за то, что целевая система после того, как её изменили создатели (провайдеры сервиса), перешла в новое состояние, что-то во внешнем мире поменялось к лучшему, сделало шаг к удовлетворению потребностей, снятию чьих-то беспокойств, удовлетворению чьих-то интересов. Если сервис парикмахерской (то есть сервис стрижки-укладки), то платят за работы, дающие причёску, которая начинает потом эксплуатироваться в составе ухоженной головы. За эскиз причёски и разговоры парикмахера по поводу выбора модели причёски могут заплатить, но только в том случае, когда конечная причёска окажется годной к эксплуатации. Если окажется, что это в реальности не происходит, то заказчики вернутся в парикмахерскую и опротестуют и причёску, и эскиз, и результаты разговоров с парикмахером: это уже внутреннее дело службы, оказывающей сервис, разбираться, кто там в её внутренней структуре дал такой сбой, что внешнее поведение-сервис и его результаты оказались неправильными. Целевой же системой всё равно будут причёски, и от класса этих причёсок будет зависеть, брать за сервис стрижки будут 100 рублей, 1000 рублей или даже 10 тысяч рублей.

Интересно, что какой-нибудь инженер-конструктор всё понимает в этих рассуждениях про парикмахерские, про сервисы доставки потребительских товаров, но когда речь идёт о том, что он выполнил какой-то шаг проектирования своей целевой системы (например, промышленного трубопровода), то он хочет безусловной оплаты своего даже потенциально очень кривого и ошибочного труда независимо от того, будет ли в итоге хороша целевая система. Но он ничем тут не отличается от парикмахерской: по факту он хочет оплаты за эскиз трубопровода (не отличается от эскиза причёски — тут любая целевая система подойдёт), и ему всё равно, будет ли его «причёска» реализована, или нет. Беда ещё и в том, что многие конструкторы даже не знают, как выглядит в жизни их работающая целевая система — никогда не сталкивались с ней «в металле», причём не на заводе, а

прямо у клиента. Нет, целевой системой тут будет работающий промышленный трубопровод, если он будет плохо работать — то денег за него не заплатят, и конструктор за свою работу долю своих денег не получит. Все бизнесы на свете — это такие «обобщённые парикмахерские», все прикладные рассуждения о бизнесе и инженерии похожи друг на друга, на уровне мета-мета-модели нашего руководства (в типах мета-мета-модели) это одно и то же рассуждение.

Сервис-ориентация. Мир провайдеров.

Переход в мышлении от поставки продуктов как целевых систем к оказанию сервисов (выполнению экземпляров работ услуги/метода провайдером/поставщиком услуги/метода над чужим предметом работы) оказался очень продуктивным и в современном языке называется **сервис-ориентацией.** И мы видим всё больше и больше провайдеров — «интернет-провайдеры» как раз отсюда, «поставщики сервиса доступа к интернету». Предмет их сервиса, он же целевая система — «доступ к интернету», подразумевающий определённое физическое состояние аппаратуры клиента и аппаратуры провайдера, обеспечивающей передачу данных между аппаратурой клиента и аппаратурой всех остальных клиентов всех остальных интернет-провайдеров мира, имеющих «доступ к интернету» в мире.

Поставщики софта, который работает как облачный инструментарий, обрабатывающий по какому-то методу чужие данные, используют бизнес-модель SaaS, software-as-a-service. Тут надо учесть, что в информационных технологиях сервис часто рассматривается как сервер (микросервисы — это вообще-то микросерверы, а не микросервисы как функции/методы серверов), в крайнем случае — неодушевлённый «станок», но не метод/поведение, метод вообще может быть опущен в рассуждениях, и сразу речь идёт о «вызове сервиса» как проведение работы провайдером, метод этой работы не обсуждается отдельно. Всегда пытаемся уточнить, что же имеет в виду собеседник, когда поминает термин «сервис», помним про множество словарных значений каждого термина! Софт-как-сервис — это софт, который оказывает сервис, софт тут — служба/сервер/провайдер сервиса. Но провайдером могут назвать и организацию, у которой есть мастерство проведения сервиса, а софт тут — инструментарий. Если бы это была мастерская, где точили ножи, то рассмотрение тут было бы то же самое — «станок-как-сервис», можно было бы принести свои ножи, как в софт в модели SaaS несут свои данные, поточить ножи, как в модели SaaS обрабатывают свои данные, и забрать ножи после заточки, как в модели SaaS забирают данные после их обработки. Плата только за выполнение (работ) услуги/сервиса, не надо платить за ножи или данные.

Все проекты перехода от товаров/продуктов/изделий к сервисам (сервис-ориентация) похожи в том, как о них думать. Сервис-ориентация начала затрагивать и традиционные продукты и изделия, то есть физические объекты, изготавливаемые как «товар» — то есть поставщики товаров начали тоже переходить на бизнес-модели поставки сервиса (вернее, поставки работ сервера по методу сервиса).

Например, возьмём чугунные чушки, «чугун серый литейный»:

Если учесть сервис-ориентацию, то услугой/сервисом будет «поставка», предметом сервиса будет «чугун» (чугунные чушки) и лучше говорить «поставка чугуна», ибо обычно оставление слишком общих слов в названиях «методов работы»::«паттернов поведения» (поставка, разработка, изготовление, создание, покупка) не позволяет вести содержательного разговора, не хватает конкретизации хотя бы до уровня метаУ-модели, предметной области.

Глаголов обычно мало, и они очень общие, это наследуют и отглагольные существительные (-ing форма глагола в английском), их мало и они очень общие. Поэтому для конкретизации указывают предметы методов, на которые указывают глаголы — и специфика предметной области вносится терминами предметов метода, а не терминами самих методов/функций/культур/способов/стратегий/сервисов.

Переход от «продажи чугуна» к «поставке чугуна» увеличивает возможности предприятия, превращающегося из «поставщика собственного товара» в «службу» или «провайдера доступа к чугуну»: сам чугун, как продукт, остаётся тот же самый — но если обратить внимание не на сам чугун, а на метод поставки, то могут существенно меняться состояния объектов в предметной области поставки. К спецификации чугуна появляется спецификация сервиса поставки, «условия поставки» (в сервис-ориентированном подходе говорят часто о SLA, service-level agreement[354]). Поставка точно вовремя для производства (just in time[355]), через склад предприятия или прямо к точке переплавки, поддержание минимального запаса в течение года или просто поставка одной крупной партии, возможности кредитования при поставке, поставка от третьих фирм, и т.п. — вариантов методов «поставка» для предмета «чугун серый литейный» множество. Получается, мы резко расширили ассортимент за счёт этих вариантов сервиса поставки, хотя предмет сервиса (чугун, который в конечном итоге будет «поставлен», перейдёт во владение получателя чугуна) оставили тем же самым.

Сервисы/функции/методы обычно легко распознать в языке — это отглагольные существительные, в которых скрыт глагол (поведение), но тем не менее в языке сохраняется словоупотребление по их поводу, как для вещи. Если действие/процесс «обучить», то сервис будет «обучение». Если действие/процесс «поставить», то сервис будет «поставка». В английском языке сервисы обычно выражены глаголами в их ing-форме. При этом не надо обольщаться высоким уровнем обобщения в

[354] https://en.wikipedia.org/wiki/Service-level_agreement
[355] https://en.wikipedia.org/wiki/Lean_manufacturing

названиях метода/сервиса (обучение, поставка). Помним, что обычно указывается и предмет: обучение азбуке Брайля, танцам, системному мышлению проходит абсолютно разными методами, но термин в названии метода будет «обучение». Поставка свежеприготовленных гамбургеров, литейного чугуна, коробочного софта проходит тоже абсолютно разными методами, но термин в названии метода будет «поставка». Интересуйтесь предметами метода, а также требуйте указать конкретный вариант метода (скажем, «чистка» может быть трубопровода и зубов, «чистка зубов» может быть с использованием разного инструментария — «ультразвуковая глубокая чистка зубов», «чистка зубов щёткой и зубной пастой», «чистка зубов зубной нитью»). Уточняйте: и когда сами говорите, и когда вам говорят про методы/сервисы/культуры! Если не уточните, то вас наверняка поймут неправильно, или вы поймёте собеседника неправильно. Выбирайте всегда правильный уровень абстракции для обсуждения!

Переход от «вещей» к «поведению», изменениям, динамике оказывается очень продуктивен, поэтому сервис-ориентация стремительно вытесняет ориентацию на продукт. Операционная система Windows 10 уже рассматривалась фирмой Microsoft не столько как продукт (целевая система, которая передаётся клиенту), но как сервис, то есть метод/культура поддержания продукта в постоянно обновлённом виде (целевая система как предмет сервиса, то есть Windows 10, принадлежит клиенту, но Microsoft меняет версии, добавляя новые возможности и исправляя ошибки — улучшает целевую систему своим внешним поведением). Microsoft из поставщика «коробочных» программных средств стал провайдером программных сервисов. Переход к Windows 11 по факту означал просто маркетинговое событие, смену имени, ибо ничего принципиально нового не привнёс: Windows 11 по факту просто очередная версия того же сервиса от фирмы Microsoft. А дальше Microsoft переводит клиентов-покупателей бывшего коробочного продукта MS Office на SaaS бизнес-модель, то есть от офисного софта как «товара» предоставляет только внешнее его поведение, делает серверный/облачный MS Office (ныне Microsoft 365) своим инструментом, размещает этот софт в своих датацентрах, оставляя на компьютерах клиентов-абонентов/подписчиков сервиса (а не покупателей коробочного софта) только знакомый им интерфейс MS Office. Предмет этого сервиса — данные клиента (обратите внимание, что методы/сервисы работают не только с системами, но и с самыми разными объектами предметной области методов/сервисов, которые называем предметами, и которые могут менять своё состояние с исходного на входе метода на выходное после применения метода — клиенты в ходе их обработки мастерством и инструментами Microsoft::провайдера::создателя меняют своё состояние с исходных на обработанные выходные после применения софта Microsoft. Предметы, а не целевые системы — методы/сервисы могут работать и с описаниями).

Сервис-ориентация чудесным образом расширяет возможные рынки: разных типов/видов/категорий/классов методов в мире не так много, несколько тысяч (и поэтому глаголов в языках всего несколько тысяч), а вот разных видов вещей — миллионы и миллионы (поэтому существительных миллионы). Мышление о методах/сервисах экономно: с миллионами и миллионами разных товаров работают одни и те же всё более и более универсальные сервисы. Эти сервисы соответствуют самым разным ожидаемым от создателей-провайдеров ролям в самых разных графах создателей. Поставщик чугуна обычно просто поставляет чугун-товар. Если же речь идёт о сервисе, который специализируется именно на методах/сервисах поставки, то легко можно кроме чугуна поставлять что-то ещё. Можно вообще стать

«маркетплейсом». Если говорить о платежах, то можно платить за что угодно. Если говорить о поддержке облачного софта, то по большому счёту этот софт может быть каким угодно. Не только с одним продуктом можно делать::метод разное, но одно и то же делание::метод оказывается хорошо приложим к разным продуктам.

Современный магазин (скажем, интернет-агрегаторы розничных торговцев, «маркетплейсы» — «ходят» сегодня туда, Yandex.market, Wildberries, Megamarket, Ozon, Amazon, они сопряжены со службами доставки и сетью пунктов выдачи заказов с примерочными) тем самым поставляет не столько покупаемый продукт, сколько предоставляет сервис (метод работы с каким-то предметом метода/сервиса) по продаже::сервис и доставке::сервис продукта::предмет. Мастерство и инструментарий маркетплейса как провайдера сервиса «поставка против платежа» с предметом «поставка против платежа». Тут легко запутаться с одноимённым сервисом и предметом сервиса, ибо «поставка» и «платёж» с одной стороны отглагольные существительные (имена методов работы с товаром/продуктом::«предмет метода» и деньгами::«предмет метода»), с другой стороны — система в физическом мире, где продукт доставлен и перешёл в собственность клиента, деньги платежа доставлены на счета поставщика продукта и перешли в его собственность (про безналичные деньги думаем примерно так же, как про наличные: там с нюансами, но всё сводится к кучке золота в подвале банка, так что деньги — это про физический мир, а не описания).

Нужно при этом понимать, что целевая система проекта создания и развития провайдера сервиса и предмет сервиса находится у клиента. Сервис провайдера что-то меняет в системах клиента (целевой или каком-то создателе в графе создателей, например, инструменте какого-то метода создания) или даже не системе, а описаниях системы клиента (иногда по длинной цепочке «описаний описаний»). Например, сервис может менять описание целевой системы, отражая это в проектной документации. Сервис может менять содержание своей базы данных с описанием клиентской целевой системы — а затем это описание может быть использовано для физических изменений воплощения целевой системы (изготовления или ремонта), и вся затея с задействованием провайдера может быть именно для этого конечного физического изменения. Все такие связанные с провайдерами сервисов/службами/серверами цепочки изменений, тянущиеся до физического мира, обязательно нужно отслеживать и продумывать.

Если вам поручили «сделать сервис» (обычно это означает «сделать провайдера, который будет предоставлять услуги, то есть проводить экземпляры работ по какому-то методу работ»), то можно совершить крупную ошибку: считать, что целевой системой является сам провайдер: его инструменты и мастерство. Нет, целевой системой является та система, которую мы изменяем сервисом/методом (возможно, по цепочке описаний, если предметом сервиса является какое-то описание, а не система, а иногда предметом будет мастерство или инструментарий создания целевой системы).

Создаваемый сервис в данном случае — это метод::поведение нового создаваемого провайдера (который работает над какой-то чужой целевой системой, по факту входя в команду), и этот провайдер — «наша система». А ещё вы сами создатель для «нашей системы», да вам нужно ещё и самому создать себя как производителя работ по созданию провайдера, то есть речь идёт о каком-то пути создания в большом графе создателей целевой системы. Целевая система остаётся той системой, по поводу которой нужно договориться всем задействованным

создателям в графе создателей — ваш проект создания нового провайдера имеет смысл ровно в той мере, что кому-то нужно что-то сделать с его целевой системой, и он приглашает «вашу систему» (провайдера) поучаствовать в проекте.

Нужно потом будет ещё убедиться, что сделанный вами провайдер/служба/сервер работами своего сервиса (работами своего метода, то есть работами своего мастерства с инструментарием) таки создаёт целевой продукт/систему с нужным уровнем качества, а ещё вы демонстрируете достаточные скорость и эффективность создания вами самого этого провайдера.

Почему графы создателей такие сложные? Вот один из путей/цепочек создания в графе создателей причёски (не парикмахерской, ибо целевая система в проекте создания парикмахерской — причёска, парикмахерская тут одна из «наших систем», а в графе создателей есть и другие «наши системы»):

- Система-1: вы, как создатель команды создания парикмахерской. Вы организуете небольшую инженерно-менеджерскую команду и поставщиков товаров (кресла, ножницы, лак) и сервисов (ремонт помещения) в проект, который создаст парикмахерскую (и мы даже не рассматриваем начало цепочки в графе создателей: вам ведь «поручили создать» парикмахерскую, то есть вы ещё и не в начале цепочки, но вас кто-то создал как создателя!)
- Система-2: инженерно-менеджерская команда и поставщики товаров и сервисов, которые создают парикмахерскую (эту команду организовали вы)
- Система-3: парикмахерская (люди с парикмахерским мастерством, помещение с инструментами, менеджеры), созданная вашей инженерно-менеджерской командой
- Система-4: причёска, целевая система всего проекта, создаваемая парикмахерской в ходе кейса/проекта для каждого клиента парикмахерской.

Вот это цепочку/путь на графе создателей хорошо бы себе ясно представлять. Без чёткого понимания кто какую систему создаёт, за что отвечает, нельзя разобраться в проекте, риск провала будет велик.

Так что в проекте создания парикмахерской::провайдер для выполнения работ сервиса стрижки и укладки целевой системой (продуктом) будет причёска чужих волос, которая меняет своё состояние с «не было» в «стало». Частные методы, составляющие все полные методы работ (сервис) по стрижке и укладке — выбор фасона, стрижки волос, укладки, возможного смешивания волос с лаком и всякими украшающими ленточками и цветочками (если они предусмотрены). Это и есть создание и (поскольку эти причёски будут делаться снова и снова, и будут становиться всё лучше и лучше) развитие причёски. Работы по созданию и развитию причёски и есть оказание/работа сервиса/метода ООО «Чикчик» со всеми его людьми-парикмахерами и помещениями с инструментами парикмахеров в роли парикмахерской.

Шикарный сервис стрижки (метод работы парикмахера с волосами клиента золотыми ножницами на кресле-троне под живой оркестр) с результирующей плохой причёской — это не то, что нужно, это явно не будет успешной целевой системой. В сервисе главное — достижение состояния целевого предмета сервиса (предмета метода работы, то есть предмета создания причёски — достижение целевого состояния причёски). Причёска для спорта вместо причёски для свидания — это ведь тоже не то, что нам нужно? Когда мы занимаемся системой создания, чтобы получить сервис создания целевой системы (разрабатываем

серверный софт, открываем сервисную фирму-провайдера), никогда не нужно забывать о целевой системе и её надсистеме.

Очень легко забыть, для чего вообще существует создатель-провайдер, для чего будет задействован сервис — и после того, как это забыто, целевая система перестаёт быть успешной (а ведь всё делается для неё), довольные клиенты вдруг становятся недовольными, и вы их теряете. Если менеджер думает о заводе по выпуску самолётов как своей целевой системе, то завод может быть супер-пупер каким продвинутым, люди супер-пупер какими сотрудничающими, рабочие процессы супер-пупер какими отлаженными, но если выпускаемые самолёты не летают (целевая система — используемый, то есть летящий, самолёт), то и завод по выпуску таких нелетающих самолётов не нужен (более того, он вреден: прожигает деньги инвесторов).

Если вы думаете о целевой системе и внешних проектных ролях проекта по её созданию и развитию, то думайте в письменном виде: документируйте эти мысли и обсуждайте их с командой и этими внешними проектными ролями. И проверяйте по самым разным чеклистам, обо всех ли важных объектах проекта вы подумали, не пустили ли вы дело на самотёк, не сделали ли вы ошибок в своём мышлении, не пропустили ли чего в своих работах.

Примеры сервисов и их провайдеров

Вам совсем необязательно проговаривать типы из трансдисциплины системного мышления, как это мы делаем тут в учебном тексте: «фрукт яблоко», «мебель стул», «система самолёт», «сервис-провайдер парикмахерская» и т.д. В жизни важно просто обращать внимание на типы объектов (удерживать эти типы объектов в голове, молча), чтобы убеждаться в правильном выделении важных объектов из многообразия окружающих объектов и проверять, что вы выполняете правильные операции с этими типами (не ожидаете, что описание будет создателем, не ожидаете, что целевая система — предприятие, и т.д.).

Вопрос в запоминании чеклиста из типов, о типизированных ими объектах которых вы должны подумать, выделить их вниманием в ваших рабочих и личных проектах. Все эти типы из нашего руководства связаны между собой отношениями, и легко проходить в мышлении по этим отношениям, перенося дальше внимание с этих отношений и типов мета-мета-модели на конкретные объекты вашего проекта.

Если вы выявили целевую систему вашего проекта, то для неё обязательно должно быть окружение и функция/метод/сервис (тут длиннейший ряд синонимов) в этом окружении. И ещё у неё должны быть подсистемы, которые каждая имеют свою функцию в целевой системе. А ещё должен быть создатель/constructor, и часто не один, а целый граф таких создателей — граф по отношениям создания. Да, это мы опять в кратком виде упоминаем мантру системного мышления.

Всё это работает итеративно, и отношение проходится в разные стороны. Если есть какой-то создатель, то у него (возможно, по длинной цепочке через граф создателей) обязательно будет целевая система, а у неё надсистема. А наша система может быть какой-то вообще третьей системой (например, подсистемой в целевой, или подсистемой какого-то создателя в графе создателей).

Если вы не различаете все эти виды систем в проекте, вы будете пропускать во внимании важные объекты. Если у вас проект создания учебного курса (скажем, вы хотите делать onboarding, знакомить новых сотрудников с вашей организацией), то

должна быть и целевая система — мастерство (скажем, мастерство надлежащего поведения сотрудника в организации). Если у вас проект какой-то системы доставки (провайдер сервиса доставки), то это создающая система, а целевая — «доставленная посылка» (вручённая в руки получателю, а не которая «в пути»). Главное тут: выявить все эти системы и убедиться, что вся команда проекта понимает, каковы эти системы, даже если вы каждый раз не приговариваете тип из мета-мета-модели (то есть говорите «мы все тут делаем причёски, наша парикмахерская нужна только для причёсок, а не сама по себе», а не «у нас целевая система причёска, наша система парикмахерской нужна только как создатель целевой»).

Это только гении могут вспоминать всё нужное вовремя и «по наитию» (хотя и они не могут, они тоже обычные люди). Системные мыслители обычно не гении, поэтому они поступают не «по наитию»: они используют перечисления понятий системного подхода как чеклисты для объектов своего мышления.

Типы понятий из системного подхода (например, предлагаемые нашим руководством) сами по себе полезны в момент обучения, но потом для обсуждения в рабочем проекте того, о чём уже подумали, а о чём хорошо бы подумать, надо ещё и пойти побеседовать со знающими людьми в проекте и не только (с некоторыми можно беседовать путём чтения их книг, статей, регламентов), потому как многое не нужно выдумывать, а нужно выявлять/discover: искать в мире объекты уже заданных в наших руководствах типов, а не вообще «какие-то» объекты, которые просто бросаются в глаза. «Какие-то» объекты могут оказаться как важными, так и неважными. **Объекты тех типов, которые задаются системным мышлением, с максимальной вероятностью являются важными.** Их типы и важность кажутся банальными, но попробуйте спросить членов вашей команды, что же в проекте создаётся и развивается (в наших терминах — целевая система, а в небольшой команде в большом проекте — «наша система»), и вы удивитесь разнообразию мнений. Как можно что-то создавать и развивать быстро и эффективно в таких условиях всеобщей недоговорённости?

Небольшие нюансы каждого проекта (например, по-другому организованная команда проекта) дают большую разницу в том, как удобно выделить разные системы в проекте. Каждый раз для каждого проекта нужно заново делать работу по выявлению целевой системы и нашей системы, определению потребных сервисов, и всему остальному системному мышлению. Поэтому дальше идут не «типовые ситуации» и «алгоритмы мышления», а просто примеры. Берите их как начало рассуждений и обязательно адаптируйте.

Рассмотрим простейший пример: проект, полностью эквивалентный созданию парикмахерской (и мы говорили, что ситуации сервисов на уровне мета-мета-модели все эквивалентны, объекты там типизированы одними и теми же типами и связаны одними и теми же отношениями) — но вы будете делать шлифовальную мастерскую. Удивительно, но редко сталкивающиеся со шлифованием люди мало задумываются над ровно теми же вопросами, которые им могут прийти в голову, когда они думают над хорошо знакомой им ситуацией стрижки в парикмахерской, просто потому что с парикмахерской они все сталкивались, а со шлифовальной мастерской — нет. И вот это «задуматься над важным, когда оно пропущено» — в этом и проявляется сила системного подхода, сами понятия системного мышления, типы из его мета-мета-модели (система в вариантах её видов, функция/метод/сервис/культура, работа, роль, агент и т.д.) тут являются

ключевыми подсказками, чеклистом «о чём надо подумать», «о чём надо рассказать, когда спросят, чем занимаемся», «о чём надо спросить, чтобы разобраться».

Шлифовальная мастерская как провайдер сервиса шлифования имеет какую целевую систему? Шлиф: та самая зашлифованная работами сервиса шлифования поверхность (физический объект! Шлиф как целевая система занимает место в пространстве, характеризуется своей шероховатостью, с одной стороны прилегает к телу шлифуемого объекта, с другой стороны касается какой-то системы в окружении). Так что шлифовальная мастерская в целом (или шлифовальный станок в её составе, это смотря какой у вас проект) — это система создания шлифа.

Обязательно нужно рассмотреть надсистему: для чего шлифуем? Что будет соприкасаться со шлифом, или шлиф просто должен блестеть для красоты? Нужно ли закрывать шлиф лаком, чтобы он не окислялся? Каков класс обработки поверхности? Важно, что все эти вопросы автоматически придут в голову опытному шлифовщику. Но мы-то не опытные шлифовщики, а нам надо, чтобы эти все вопросы тоже пришли к нам в голову! Системное мышление даёт ровно это, когда заставляет подумать

- и о целевой системе (это не так просто понять, что это шлиф — хотя это полностью эквивалентно «причёске» в случае парикмахерской, только вместо стрижки и укладки — шлифовка и полировка, а вместо волос на голове — поверхностный слой вещества на заготовке),
- и о надсистеме (шлиф в соприкосновении со шлифуемой деталью с одной стороны и пока непонятно чем с другой стороны шлифа),
- и о потребностях внешних проектных ролей (зачем шлифуем? Что интересно получить инженеру от надсистемы, если в составе надсистемы будет шлиф?)
- и о концепции использования шлифа (какое поведение шлифа хотим получить в ходе его задействования в надсистеме, что от него ожидается?)
- и концепции шлифа (каким будет шлиф, не будет ли дополнительных покрытий, напылений, лакировки?),
- и о том, как обосновать результат (как убедиться, что все интересы удовлетворены?).

Если речь идёт о сервисе шлифовки, то можно подумать и о функциях создателя шлифа (заметили, что «функции создателя шлифа» это синоним «сервиса шлифовки», а также «метода работы шлифовальной мастерской»?), которые он будет выполнять по отношению к заготовке целевой системы в надсистеме создания детали в целом: функция «сделать блестящую поверхность» — для этой работы можно просто покрасить поверхность, или никелировать её, или даже заклеить блестящей плёнкой; «сделать гладкую поверхность» — для этой работы тоже можно предложить ряд способов, кроме шлифования.

Как выбирают обращение к шлифовальной мастерской:

- сначала определяется конечное состояние предмета метода/сервиса/функции,
- а затем можно выбирать из ряда вариантов возможных методов/сервисов/функций/практик.

Ещё один вариант сказать то же самое, но чуть подробней (близко к системной мантре):

- сначала определяется функциональный объект как предмет метода («гладкая поверхность» — и определяется, что там за свойства гладкости нужны в окружении этого предмета метода в момент использования этого предмета), концепция использования поверхности
- а затем предлагается конструктивный объект для воплощения функционального объекта (шлиф на болванке, гладкий слой краски, наклеенная сверху пластиковая плёнка и т.д.), концепция поверхности.
- Дальше думаем, каким сервисом/способом/методом это можно осуществить, (шлифовка! А предмет метода шлифовки — шлиф. С этого момента «гладкая поверхность» конкретизируется: «шлиф»::«гладкая поверхность»).
- Дальше думаем, кто может быть создателем, то есть провайдером такого сервиса (провайдером сервиса по созданию шлифа может быть шлифовальная мастерская, или даже шлифовальный станок, или даже шлифовка своими силами вручную, хотя в последних двух случаях «наших сотрудников, наших инструментов» продолжаем говорить о методе, но в первом случае «внешней мастерской» говорим о сервисе).
- Выбираем конструктивный объект, выполняющий роль шлифовальной мастерской (например, ООО «Вжжж»).

Системное мышление заставит подумать и о внешних ролях (инженер::клиент, которому нужен шлиф), и о внутренних/командных::роли (кто будет шлифовать?) и о внешних по отношению к создателю::система (кому нужна шлифовальная мастерская, но не нужны сами шлифы). Заставит документировать интересы внешних проектных ролей и концепцию использования шлифа (описать надсистему шлифа — что хотят внешние проектные роли от шлифованной поверхности плюс того, что к ней прилегает), концепцию самого шлифа, которая даёт ожидаемые свойства (класс обработки поверхности, необходимость какого-то покрытия после шлифовки, например, лакирования: как обеспечить те характеристики шлифа, которые важны в момент использования).

Ровно такие же рассуждения нужно делать и в других случаях. Так, вы делаете систему оформления командировок, и вы подозреваете, что вашим клиентом тут является бухгалтерия. Системное мышление заставляет пройти по следующим рассуждениям (подробней эти рассуждения по поводу служб администрации приводятся в руководстве по системному менеджменту, они типовые для почти всего, что связано с администрированием):

- Поскольку речь идёт о софте (хотя мы пока и не знаем, «система оформления командировок» включает ли в себя людей, не является ли каким-то оргзвеном, или это таки просто софт-инструмент без отдельного оргзвена для него), то этот софт что-то описывает в своих данных. Что описывает? Командировку, или на формальном языке — «деловую поездку». Итак, целевая система (ради которой всё затевается, включая оформление через ваш будущий софт) — деловая поездка, которая понимается как набор/composition вполне материальных объектов: человек (необязательно сотрудник!), который куда-то едет, пункт начала поездки, пункт конца поездки, транспорт, зарезервированное место в транспорте (описывается административно значимой записью на серверах перевозчика, а раньше это было «билетом», описание собственно «езды»), место проживания (описывается административно значимой записью на серверах гостиницы, но обычно достаточно выписки — квитанции) — и всё это на каком-то отрезке

времени.

- Уточняем теперь целевую систему. Кандидатов в рассмотрение целевых систем становится три: 1. Удалённая работа (поездка тут не волнует, волнует то, что надо добиться каких-то изменений в объектах, хотя эти изменения без поездки невозможны). Например, начальник поручает сотруднику удалённую работу, или сам сотрудник решает, что надо провести работу. 2. Собственно, деловая поездка — как её для себя определяет сотрудник. В неё входит выполнение работы, а также собственно поездка — транспорт, ночёвки. 3. Оформляемая часть деловой поездки — это тот объект, который интересует бухгалтерию: только те части поездки, которые как-то связаны с оплачиваемым (а не всем) транспортом и оплачиваемыми (а не всеми) ночёвками. В большинстве связанных с администрированием ситуаций надо выделять похожие три целевых системы.
- Клиент при этом не бухгалтерия, а те, кому нужны удалённая работа (например, начальники командируемых) и деловая поездка (командируемые), а бухгалтерия оказывается в подчинённой роли внешнего провайдера сервиса по созданию оформляемой части деловой поездки. Так что дальше надо определяться: софт этот как инструментарий чей — инструментарий начальника для оформления удалённой работы, сотрудника для помощи в деловой поездке или бухгалтерии для оформляемой части деловой поездки.
- Если это понять, чей инструментарий из этих троих создаваемый софт и поддержать начальников и сотрудников в первую очередь, а бухгалтерию — в последнюю очередь (а не наоборот — удобный для бухгалтерии метод оформления, неудобный для начальников и сотрудников как клиентов бухгалтерии, а затем загонять начальников и сотрудников в неудобный сервис под угрозой отнять их деньги и нервы, зато сэкономить нервы бухгалтерам), то команда проекта с результатами её работы будет нужна прежде всего всем начальникам (которые хотят удалённой работы) и сотрудникам (которые хотят отправиться в деловые поездки), а не только бухгалтерам фирмы, которые эти поездки должны оформить, но им этого по большому счёту не надо («не смог оформить — не поехал, последствия неисполнения удалённой работы разгребаем не мы, бухгалтерия, так что заткнитесь и выполняйте любые дурацкие правила оформления»). Это существенно меняет переговорные позиции в проекте! Из врага всех начальников и сотрудников команда разработки инструментария (софта оформления удалённых работ/деловых поездок) становится другом всех начальников и их сотрудников! И дружат они все против бухгалтерии, для которой ноль поездок (ноль работ по предоставлению сервиса, да это и не классический сервис, они за него денег не получают!) — это идеал/предпочтение, ибо уменьшает количество их забот по оформлению. Такие развороты ситуаций типичны, когда речь заходит о системном мышлении: приходится честно отвечать на вопрос, кому что нужно и договариваться о балансе интересов для создаваемой системы, какой бы она ни была. При этом помощь со стороны типов мета-мета-модели в том, что становятся понятными объекты, по поводу которых надо договариваться (предметы договорённостей, обычно это предметы каких-то методов, которые используют договаривающиеся агенты-в-ролях) и понятно, кому договариваться.

- Какие методы работы нужно задействовать, чтобы целевая система «оформляемая часть деловой поездки» была создана, предмет этих методов из состояния «оформляемой части деловой поездки нет» превратился в «оформляемая часть деловой поездки создана», затем через несколько стадий «оформляемая часть деловой поездки выполнена»? Что содержательно нужно сделать, то есть каким способом поменять состояние каких предметов/объектов, (но пока не рассматриваем агента: кто или что будет проводить работы по этим методам)? Создатель с помощью инструментов (люди бухгалтерии с софтом, но мы уже выяснили, что там надо учитывать и командированных, и их начальников и даже случаи «сам себе начальник», самопосыл в командировку) должен зарегистрировать потребность в поездке и открыть мини-проект удалённой работы, выделить финансирование на поездку (но там может быть ещё и часть финансирования на сами работы!), купить билеты, возможно, забронировать гостиницу, выдать командировочные, оформить документы (командировочное удостоверение, билеты, бронь в гостиницу, грузовые документы для какой-нибудь провозимой с собой аппаратуры, если нужно), принять отчёт о поездке — и тут есть множество вариантов, зависящих от принятых методов ведения командировок. Где-то не нужны командировочные удостоверения, где-то не нужны даже отчёты, где-то юристы больше защищают сотрудников, где-то юристы настаивают на выполнении стандартов деловых поездок ещё советского времени (когда полёт на самолёте был чем-то экстраординарным).
- Какой создатель должен эти методы выполнить? Софт (и там создатель софта — команда программистов, «создатель создателя»), или служба/«провайдер сервиса» оформления (софт на серверах плюс люди из бухгалтерии, работающие с этим софтом как операторы? Терминология «службы» и «сервиса» будет задействована только в этом случае, но и тут варианты: чем более внешняя эта служба, например, внешний провайдер сервиса бухгалтерии, тем больше будет уместна эта терминология, а вот если это делают сотрудники той же фирмы, которая оформляет для других своих сотрудников командировки — вот тогда слово «сервис» для выполняемого там метода работы или «служба» для самих этих людей с их мастерством и инструментами-софтом можно не услышать, «язык сопротивляется»)? Что будет сочтено успехом проекта создания софта, создания службы, создания деловой поездки? Какие роли требуются для этих разных проектов создания создателя деловой поездки (создания службы оформления, а там внутри ещё и создания инструмента оформления, то есть софта), создания деловой поездки (и мы сходу там нашли оператора службы, начальника командированного, самого командированного — а роли программистов и менеджеров службы остались в прошлом проекте создания службы и там внутри создания софта)? Кто исполняет эти роли в команде программистов, в команде создающих службу менеджеров, в команде создающих деловую поездку (даже если в этой последней команде нет ощущения команды и даже проектом это никто не считает, но это же проект, и участвует в нём много людей, им надо договариваться и уметь сыграть свою ролевую игру!). Беда, если вы считаете «нашим проектом» проект создания софта, а в момент приёмки проекта в эксплуатацию, или в момент возникновения трудностей в связи с тем, что нет исполнителей других ролей в полноценной службе оформления поездок (время эксплуатации системы создания поездок)

спрашивать начнут с текущего разработчика софта! Объяснений, что «мы считали, что от нас нужен только софт, а людьми займётся кто-то другой», слушать ведь никто не будет! В конечном итоге нужны оформленные деловые поездки, или даже сами полные деловые поездки (в которых сотрудники ухитрялись сходить в неоплачиваемый ресторан или переночевать в гостинице, которая не выдала никаких платёжных чеков), а то и выполненные удалённые дела — ибо важны в конечном итоге именно они, и если они сорвались по причине трудностей деловой поездки или оформления деловой поездки, то это «внутренние ваши трудности, а вот за выполненные дела отчитайтесь»! В зачёт идут только ожидаемые общие результаты деловой поездки, и если без каких-то людей-сотрудников службы оформления деловой поездки софт их сам выдать не может, то нужно будет «изготовить» и этих сотрудников, т.е. обучить новым методам работы ещё и каких-то людей, а ещё нужно будет обучить новым методам работы и начальников командированных, и самих командированных агентов!

Заметим, что всё сказанное по поводу проекта создания софта для оформления командировок не даёт чёткого ответа на вопрос: как же организовать оформление командировок. Но зато показано, как системное мышление даёт набор объектов, который удобен для обсуждения альтернатив. Можно не обсуждать то, что же именно нужно обсуждать. Обсуждать нужно сразу системы, их окружение, внешние проектные роли, графы создателей и отдельных создателей в них (как инструментарий, как провайдеров/службы с их сервисами, как агентов индивидуальных или коллективных), методы работы, роли и т.д. — мышление задействует объекты не любых типов, а именно этих типов, и эти типы берутся как раз из руководств по программам рабочего и исследовательского развития).

Ещё один пример: магазин предоставляет сервис покупки — он создаёт покупку как какой-то набор товаров, находящихся в собственности покупателя в его руках на выходе из магазина, и деньги покупателя на счетах магазина (о деньгах думаем как о физическом золоте, можем не отвлекаться на безналичные расчёты и их «нефизичность»). Магазин помогает сделать/создать/изготовить (инженерная терминология!) покупку: он сводит покупателя со своими полками и кассовыми аппаратами.

И тут надо различить два значения термина «покупка»:

- Процесс/метод/сервис, шоппинг/shopping — метод работы магазина, меняющий состояние товара с «выложен на полке» на «стал товаром в покупке», а денег «в кошельке покупателя» стали «платежом за покупку». Например, «чтобы завершить покупку, нажмите кнопку «оплатить»».
- Товар, который был куплен и собственность на который перешла в руки покупателя. Например, «разверни свою покупку, посмотрим, что купил».

Если это интернет-магазин, то «покупка» в этом значении становится наряду с платежом «поставкой», и там такая же путаница поставки::метод/сервис товара::«предмет поставки» и «платежа»::сервис денег:: «предмета платежа» с поставкой::«предмет поставки-сервиса» как уже бывший товар, а ныне собственность покупателя у него в руках, а также платежа::«предмет платежа-сервиса» как деньги, но уже на счетах у продавца. Иногда вводят и термин «поставка против платежа» — это тоже иногда конечное состояние товара и денег, но иногда — сервис интернет-магазина или маркетплейса (агрегатора магазинов

плюс службы доставки).

Сервис::поведение и «предмет сервиса»::«предмет поведения» в бытовой речи тоже нещадно путаются, как и сервис::повдение и сервис-провайдер (в IT сервис-провайдер может быть назван микросервис или сервис, но не микросервер или сервер). поэтому внимательно следите за типами: что там объект, а что поведение объекта. В случае проекта магазина путаются метод/процесс покупки/поставки и предмет, который куплен-поставлен (его тоже назовут «покупка» или «поставка»). Если вы купили три килограмма сахара, потратив на это десять секунд в интернет-магазине, то в одном случае можно будет сказать «моя покупка заняла десять секунд», в другом случае — «моя покупка весит три килограмма».

Сервис покупки (поставки) и платежа — поведение провайдера сервиса в части паттернов этого поведения (способ/метод работы провайдера, когда он работает над чужим предметом метода, подразумевается, что магазин не производитель товаров/изделий/продуктов, а именно сервисная организация, «переработка чужого»), а в итоге у нас в руках (а у магазина на счетах) покупка (поставка) и платёж как материальные объекты — это результат предоставления/оказания сервиса/услуги, проведения работ сервиса: (бывший) товар в руках покупателя и деньги на счёте магазина.

В парикмахерскую приходят за стрижкой как работой сервиса или за причёской как результатом стрижки, итогом предоставленного сервиса? По большому счёту, за причёской. За результатом сервиса. Но главное там — стрижка и укладка как процессы/методы/сервис. В магазинах (и практически везде в других системах создания, по факту это часто предприятия или их части) то же самое. Сервис доставки в интернет-магазине имеет целевой системой «доставку», понимаемую как уже доставленный и оплаченный пакет уже в руках счастливого получателя этой «доставки». Этот интернет-магазин с его сервисом «доставки» (процесс/метод/способ работы) задействуется для получения «доставки» (бывший товар в его конечном состоянии нахождения у покупателя) примерно так, как парикмахерская с сервисом стрижки и укладки задействуется для получения причёски и шлифовальная мастерская с её сервисом шлифования задействуется для получения шлифа. Само слово-термин «доставка» в разных случаях может обозначать как сервис, так и целевую систему, обрабатываемую этим сервисом, так что его значение нужно обязательно уточнять. **Отглагольных существительных нужно бояться: иногда они означают поведение/«глагол», а иногда и какую-то вещь/«существительное» как результат этого поведения.**

Этот сервисный подход (сдвиг с изделий/продуктов как наших или целевых систем на заказ отдельных работ по каким-то методам для их получения) сдвигает внимание с рассмотрения только собственных проблем создателей «меня волнует мой завод, производящий товар/продукт» на рассмотрение также и проблем многочисленных подролей «клиента/заказчика», делает шаг к более детальному описанию подролей. В сервисе «продажи» магазина-провайдера (тут ещё важен и референтный индекс, то есть от чьего лица делается референция/«описание предметов/референтов» — для магазина этот сервис «продажи», для клиента во всех его подролях — сервис «покупки») надо бы учесть минимально:

- Бенефициар («конечный пользователь» создаваемой системы, хотя и пользователь тут слишком общее имя — мы говорили, что лучше для игры говорить не «пользователь», а игрок, для токарного станка говорить не

«пользователь», а токарь, для мороженого — едок, а не «потребитель»). Скажем, девочка, которая будет носить купленное в магазине платье.
- Плательщик — с чьего счёта (кредитной карты) уйдут деньги. Например, кредитка папы, оплатившего покупку платья для девочки.
- Собственник, на чьё имя оформляется купленная вещь, выписывается гарантия. Это и будет обычно «покупатель». Например, это мама, покупающая платье девочке.
- Получатель (представитель) собственника. Например, дедушка, который встретит курьера из магазина — ибо девочка в садике, мама и папа на работе.
- Консультант. Например, подружка, которая запретит девочке носить платье, ибо оно не удовлетворяет её представлениям о моде. И поэтому уговаривать купить платье (магазин и производитель платья ведь занимаются маркетингом!) надо в том числе и консультанта!
- … и этот набор подролей «покупателя/клиента» можно продолжать.

Провайдер сервиса (участник графа создателей целевой системы) со всем своим мастерством и инструментами проведения работ по сервису выживает в долгосрочной перспективе только в том случае, когда нет вопросов к качеству результирующей целевой системы как результата сервиса. Если целевая система в итоге оказания сервиса получается плохая, а создатель-провайдер при этом восхитителен, то жизнь этого провайдера сервиса закончится в момент окончания инвестиций её создателя[356].

Если владелец магазина будет заботиться не о состоянии покупателя и размере покупки в момент ухода покупателя из магазина, а просто о своём торговом зале, то покупатели будут только мешать торговому залу быть в идеальном состоянии! Корабли лучше всего сохраняются в порту, в гавани. Но их строят не для того, чтобы они стояли в гавани: их место в открытом море, где много опасней и хлопотней, но без этого не будет создана целевая система: груз в месте его назначения. В конечном итоге нужен не столько нетронутый инструмент и незадействованные работы мастерства, сколько, целевая система и какая-то работа и тем самым износ создателя целевой системы — та самая «покупка» или «доставка» для магазина, та самая «оформленная деловая поездка» для бухгалтерии. Тут, конечно, требуется лидерство, ибо если магазин не будет торговать, его просто закроют, а вот если бухгалтерия не будет оформлять деловых поездок, будет делать всё, чтобы этих деловых поездок не было и ничего не надо было бы оформлять, то её могут и не закрыть. Но это уже вопросы системного менеджмента, само по себе системное мышление только даёт объекты, которых достаточно для подобных обсуждений.

Давайте ещё раз разберём пример целевой системы «лот», там тоже акцентируем внимание на сервисе. Нефтеперерабатывающий завод производит небольшие порции какой-то фракции перегонки нефти. Эта фракция почти ничего не стоит, «отходы производства». Провайдер сбора::сервис лота::целевая система железной дорогой доставляет эти порции к месту на побережье, где сливает их в арендованный огромный резервуар. Потом, после заполнения этого резервуара провайдер сбора лота переливает фракцию в танкер и отправляет на переработку на один из химических заводов. И получает на выходе весьма дорогое сырьё. Почему ему нужно собирать лот на побережье в резервуаре, который ему даже не

[356] Социализм этим и отличался: отсутствие инвесторов позволяло почти не обращать внимания на целевые системы, основное внимание было направлено на системы создания, см. http://odesskiy.com/zhvanetskiy-tom-3/parovoz-dlja-mashinista.html

принадлежит? Химзавод не примет маленькую партию фракции, ему подавай сразу большой танкер. Почему не собирать маленькие порции в танкер? Танкер дорого стоит, и лучше бы ему быть в пути, а не стоять в порту и ждать прибытия маленьких порций фракции-сырья. Целевая система для провайдера сбора (предмет сбора) — лот::«единица торговли или переработки». Если не будет собранного лота, то не будет переработки фракции химзаводом. Вот поэтому провайдер сбора изготавливает лот путём накопления маленьких порций дешёвой фракции, получая большую порцию, которая позволяет ему заключить контракт на переработку: эта большая порция и есть лот. Следовательно, у нас в рассмотрении — провайдер сбора::сервис лота::«предмет сбора» (и этот же «лот» — целевая система для проекта создания провайдера — а не организация провайдера!). Провайдер изготавливает этот лот путём ведения работ по сбору лота как прохождения «сеансов» сбора как закупки и перевозки к резервуару отдельных порций сырья.

Провайдер обучения::сервис изготавливает мастерство::система, которое обучаемые будут эксплуатировать после обучения. Мастерство определяем как специализированный вычислитель (устройство), реализуемое частью мозга, а сам мозг рассматриваем как универсальный вычислитель. Вычислитель можем рассматривать вместе с устройствами ввода-вывода (в том числе манипуляторами и датчиками, ушами-глазами-руками-ногами-зубами — телом), а если нужно как-то задействовать силу тела, то обучение тренирует не только мозг, чтобы в нём выросло мастерство, но и тело, чтобы как-то наросли мышцы и фасции, нужные для эксплуатации мастерства. Телу можно дать инструменты, так что мастерство задействует не только себя самого (только обработка информации) или себя с телом (даже ввод-вывод информации для размышлений — это уже действия не только мастерства-вычислителя, но и тела), но и инструменты. Тело тут можно считать тоже инструментом, а мастерство — контроллером всех телесных и внешних (экзотелесных) инструментов. Подробней об этом всём говорится в руководстве по инженерии личности, ибо личность — это совокупность всех видов мастерства какого-то агента.

Создание провайдера обучения::сервис/метод мастерству::«целевая система» какого-то метода («создание создателя мастерства», но в этой формулировке нет сервиса/функции/метода создания, то есть не упомянуто «обучение») тоже может служить примером, который легко модифицировать как прототип для разных других проектов обучения::сервис: от классического программирования компьютера (трудно считать это «обучением», тем не менее) до корпоративных/организационных изменений (самых разных людей в фирме надо научить выполнять с удовольствием свои рабочие роли), и даже маркетинг можно рассматривать как частный случай (научить человека использовать какой-то продукт или пользоваться сервисом, начиная с того, что научить его покупать, как только увидишь, а если не увидел, то ещё и искать, чтобы купить!). Создание провайдера чего угодно имеет целевой системой не самого этого провайдера, а предмет его сервиса или какую-то иную целевую систему, которую помогает создать предмет его метода (если, например, предмет его сервиса/метода — описание, то целевую систему надо искать где-то ещё дальше в графе создателей).

Хорошие целевые системы — игровые сессии, которые создаются провайдерами игр (провайдер сервиса игры, игровая сессия как создаваемая сервисом система, чем больше скорость выпуска игровых сессий, тем больше может быть доход провайдера). Игровая сессия состоит из игроков в определённом состоянии,

игрового оборудования в определённом состоянии (скажем, игровой компьютер, датацентр, связь между ними) и административного и игрового софта на этом оборудовании в определённом состоянии.

Если речь идёт о провайдерах безопасности, то никто не понимает, что это такое, договориться невозможно. Но как только вы скажете, что делаете, например, сигнализацию для каких-то понятных неприятностей (а непонятное слово «безопасность», похожее на название свойства, табуируете) — всё волшебно налаживается. И тут надо разобраться, что вообще обсуждается:

- Событие «неприятности» — можно ли тут рассуждать о нём как о мероприятии, по типу «танцевального перформанса», или это просто момент во времени и такого рассуждения не получится?
- Сигнализация — это что: инструмент, который выдаст сигнал, или метод/функция/сервис такого поведения? С отглагольным существительным надо быть осторожным.
- Тем самым: вы делаете сигнализатор::инструмент или провайдера сервиса сигнализации (например, людей плюс сигнализатор::инструмент, и это будет «служба сигнализации», а не «сигнализатор»)?

Целевые системы бывают самые разные, сервисы и их предметы бывают самые разные, типовых в плане объектов предметных областей (метаУ-модели) и тем более в ситуациях конкретных предприятий (метаС-модели) вы не найдёте. Но рассуждения про целевые системы, сервисы и их предметы — однотипны, если вы используете типы мета-мета-модели из наших руководств. Вопросы к самым разным видам сервисов задаются по одинаковым шаблонам, осмысленность ответов на эти вопросы легко проверяется.

Вы вашим сервисом будете менять состояния предметов сервиса, но каковы бы ни были эти предметы (описания, системы), вам надо добраться мыслью до каких-то систем, чтобы выбраться из мира описаний. Вы (в конечном итоге по цепочке описаний, или непосредственно, если речь идёт об обработке материалов) можете менять два типа систем вашим собственным сервисом:

- Какого-то из создателей целевой системы (команду с мастерством в каком-то методе и/или инструментарий). Создатель в ходе его работы (тут реже говорят «эксплуатации») будет задействован во время создания целевой системы, находиться где-то в графе создателей. Это часто будет называться «наша система», если вы создаёте и развиваете этого создателя, да и вы сами попадаете в граф создателей, разве что в нескольких отношениях создания от целевой системы.
- целевую, которая будет задействована где-то в потребительской надсистеме в ходе эксплуатации/использования/функционирования целевой системы. Это означает, что вы в графе создателей стоите в одном шаге/ребре от узла целевой системы (или её частей).

В первом случае будет граф создателей: вы как оргзвено-1 создаёте и развиваете оргзвено-2, которое в свою очередь создаёт и развивает целевую систему. Например, вы консультируете пиццерию (проводите серии консультаций::сеанс), которая выдаёт после этого пиццы всё лучшего и лучшего качества. И тут можно рассматривать пиццерию как производителя пиццы::системы, а можно как провайдера сервиса быстрого питания::поведение.

Это могут быть и очень длинные пути/цепочки создания в довольно «кучерявых» графах. Вот пример такого пути/цепочки: вы консультируете совладельца холдинга, который проталкивает проект реорганизации цеха красок в штаб-квартире красильного холдинга, штаб-квартира занимается организацией реорганизации цеха красок, потом цех красок занимается собственной реорганизацией, потом реорганизованный цех выпускает новую краску. Вот эта новая краска — это и есть целевая система! А вот это путь в графе её создателей: консалтинг — совладелец холдинга — штаб-квартира холдинга — цех красок как реорганизатор — цех красок как выпускающее краски оргзвено. Но для всех них выпускаемая краска — целевая система, по поводу которой все и договариваются. А те системы, которыми они занимаются при создании целевой по этой цепочке создания, разные «наши системы» (engineered systems) для разных команд в этой цепочке. Подчеркнём, что целевая система, в которую целится весть граф создателей, одна: выпускаемая новая краска, если об этом не помнить, то трудно обсуждать, что связывает в один проект все эти проекты создания и развития «наших систем» в графе создателей. При этом у реорганизованного цеха «наша система» совпадает с целевой.

Не забывайте, что в обширных графах создателей все системы создают одна другую только для того, чтобы надлежащим образом создать и развивать целевую систему. Повторим, речь не идёт обязательно о «создании целиком с нуля», но может быть и создание MVP, затем продолжительное развитие системы, а ещё создание и развитие может быть частичным: может быть только описание создаваемой системы, изменение какой-то одной характеристики, настройка-регулировка, но всё это будут действия создателя во времени создания и развития создаваемой системы.

Если целевая система оказывается не нужна каким-то клиентам (не учитывает интересов клиентов), то у вас довольно скоро не будет денег поддерживать работу всех систем в графе создателей — неучёт интересов внешних проектных ролей в пользу облегчения работы команды ведёт к созданию неуспешной системы. Большая вероятность того, что облегчившая себе жизнь в ущерб клиентам команда денег не получит. Клиенты (внешние проектные роли) просто не заплатят ей за работу и поток денег на весь граф создателей окажется нулевым.

При этом всё сказанное про формулирование целевой системы как принадлежащей клиенту и обслуживаемой/изменяемой каким-то провайдером сервиса создания (в том числе не полному созданию, а части создания, например, только описанию, или такому частному изменению как модификация/настройка, проведение одной производственной операции, например, операция окраски для красильной службы), не является «объективной истиной» или даже «типовой ситуацией». Может быть и просто поставка товара как целевой системы из собственного сырья поставщика товара, а не провайдера сервиса создания. Не всё в мире пока сервис-ориентировано. Но это для системного мыслителя практически не означает смены мышления. Если понимать, что метод и сервис — это синонимы, разве что принято говорить о них немного по-разному, ибо речь идёт по большому счёту только о том, «чьё сырьё и чей продукт», а это не так уж много меняет в проекте.

Разнообразие ситуаций в самых разных деятельностях очень велико, так что конкретные решения по составу систем проекта и по выбираемым терминам мета-мета-модели и мета-модели придётся принимать самостоятельно: в каждом конкретном проекте, с учётом интересов всех и внешних, и внутренних ролей в проектах по всему графу создателей.

Конечно, в этом подразделе был дан только минимальный набор понятий онтики сервиса. Дальше, конечно, нужно рассматривать множество самых разных ситуаций, к которым эта онтика может быть как-то адаптирована. Скажем, прохождение сервиса будет у сырья/заготовки::«предмет сервиса», а также может быть не одиночной заготовкой, а их партией/batch, заготовка и конечный предмет сервиса/метода могут быть на разных системных уровнях (если это сервис сборки, то это и есть цель), сам сервис может состоять в неоднократном приложении каких-то методов к предмету сервиса (сеансы).

Вот несколько примеров рассуждений, где мы покажем использование разной лексики и чуть более детальные описания с использованием онтики сервиса с адаптацией к различным проектным ситуациям:

- Шлифовка (шлифование) — это операция::сервис/метод провайдера шлифовки, который изготавливает шлиф::«предмет метода» в заготовке. Заготовка тут — сырьё для будущей готовой системы как носителя шлифа, надсистемы шлифа — то есть тут готовая система «отшлифованная деталь» на системный уровень выше, а целевой системой операции шлифовки будет шлиф — если провайдер шлифовки мастерская, то целевая система у неё — шлиф, а если это просто шлифовальный станок где-то на заводе, то даже готовая «отшлифованная деталь», где шлиф только её часть — это вряд ли целевая система, но может быть «нашей системой») в ходе операции::сервис шлифовки. Заготовка принимает участие в сеансе шлифовки, равно как и провайдер шлифовки (подключаясь на интерфейсе к болванке). Будет ли переименована болванка/заготовка детали в деталь как готовую систему после шлифовки, зависит от того, насколько выполнение операции шлифовки и создания шлифа::«функциональная часть готовой детали» будет последней работой/сервисом (грубо говоря, «гусеница» переименовывается в «куколку» только если метаморфоз завершён). Дальше можно рассматривать, что тут провайдер шлифовки: шлифовальный станок, шлифовальный станок и обслуживающие его люди («оборудование и приписанные к нему люди») или люди и приписанный к ним шлифовальный станок (это одно и то же, но если «станок и его люди», то проще обсуждать автоматизацию, устранение людей, а если «люди и их станок», то проще обсуждать, с кем договариваться, если станок не работает). Чем больше людей в составе «провайдера», тем больше шанс, что это будет сервис и провайдер сервиса (будут какие-то передачи материалов в шлифовку, затем будут забирать результаты — у людей, а не у станка), если нет — то просто инструмент какого-то создателя как сотрудника и метод работы этого создателя (никто ничего не «отдаёт в работу» и не «забирает из работы», это всё внутренние передачи между своими, ничего «чужого», никто никого не обслуживает, просто совместная работа в команде).
- Курс обучения — это сервис провайдера курса, который изготавливает мастерство в студенте::заготовка (тут тоже студент — надсистема-носитель для будущего мастера::«готовая система», носителя целевого для курса мастерства) в ходе «прохождения курса», состоящего из занятий::«сеансов сервиса». Студент::роль в 4D принимает участие (participation, специализация part_of) в «прохождении курса», то есть наборе занятий, равно как и провайдер курса, говорят о «наборе занятий». Будет ли агент, отыгрывающий роль студента переименован затем в мастера:: «готовая система» — точно так же зависит от того, последний ли это набор занятий

курса в какой-то их серии курсов. Моделировать ли курс как мероприятие (система) или как процесс (прохождение курса студентом у провайдера курса) — обсуждаемо, лексика будет чуть разной. Несколько студентов, проходящих курс и их преподаватель можно представить, например, потоком::системой (с точки зрения провайдера курса, разные группы — разные потоки), группой — с точки зрения деканата, ибо группа сохраняется, проходя разные курсы в какой-то последовательности, как «сеансы». В потоке агенты, отыгрывающие роли студентов::«заготовки мастера» взаимодействуют не только с преподавателем (или компьютером, к которому приписан преподаватель — если планируем автоматизацию и в конечном итоге убрать людей), но и друг с другом при групповой работе. Подробней это всё разбирается в руководстве по инженерии личности.

- Игра — это сервис провайдера игры, который стимулирует продуцирование дофамина в организме игрока в ходе работ по задействованию сервиса («прохождение игры», состоящее из «сеансов» игры::сервис/метод работы провайдера игры). Игрок принимает участие в прохождении игры через участие в сеансах, равно как провайдер игры (и в групповых играх участники могут взаимодействовать и с провайдером, и друг с другом). Игрок не становится после игры «готовой системой» (хотя меняет состояние! Устаёт, или отдыхает, или злится от проигрыша) и тем более не становится «готовой системой» после сеанса игры (нет аналога переименования «гусеницы» в «бабочку» с возможными промежуточными состояниями, ибо нет какого-то радикального изменения, даже такого не слишком выраженного, как превращения «студента» в «мастера»).
- Курс лечения — это сервис провайдера лечения (курс считаем сервисом::поведением, а не мероприятием), который изготавливает здоровую часть организма в пациенте::«заготовка будущего здорового человека, носителя уже здоровой после курса лечения части») в ходе прохождения курса, состоящего из «лечебных процедур»::«сеансов лечения».

Ещё нужно смотреть на способы моделирования проектов по организации сервис-провайдеров: проекты можно моделировать как работы по каким-то методам (проект::поведение), но можно и считать организацией проекта (проект::организация). Будьте внимательными к типу того, что у вас называется «проект»:

- Проект организационного совершенствования/улучшения/improvement — это экземпляр работы (проект::работа::поведение с чётким сроком начала и конца, понятным результатом принятого метода/стратегии выполнения проекта, известными ресурсами) сервиса службы/провайдера орг-развития, который в ходе прохождения проекта улучшения за время ряда шагов улучшения (сеансы) вносит какие-то настройки/переключения в мастерстве как предмете метода/сервиса без изменения онтики этого мастерства (например, в настройки шлифовального станка вносятся коррективы, а обслуживающие станок операторы проходят тренинг, чтобы не смотреть каждый раз в инструкцию, а делать всё «по памяти». В этой ситуации не появляется никаких новых понятий и отношений, онтика шлифовки не меняется. Вот и в ходе оргсовершенствования ровно такие изменения, а не замена, например, шлифовки на напыление — и поменяется буквально всё, людей и AI надо будет переучивать на новый метод, инструментарий метода

заменять). Альтернативный вариант моделирования ситуации — это проект как организация (люди, AI, инструментарий с понятными полномочиями по распоряжению трудом и капиталом). Проект тут провайдер, работы и сервис/метод плохо различимы — оргсовершенствование (работы одного задействования метода/сервиса оргсовершенствования). Как надо? Надо понимать, что бывают эти два варианта. Более классический, конечно, когда проект::«работа по методу»::поведение, а провайдер проекта — рабочая группа проекта.

- Проект оргразвития — это экземпляр работы сервиса офиса/службы::провайдер развития, который изготавливает новое мастерство (развивает уже имеющееся мастерство, то есть добавляет фичу, «модернизирует», что-то делает с текущей онтикой мастерства) и добавляет инструментарий в «новой оргвозможности»/«new capability»::«готовая система» в ходе прохождения проектом оргразвития ряда шагов развития, состоящих из курсов::проект обучения и лидерских мероприятий::проект для людей, а также проектов подготовки инструментов/оборудования (это специализации «шагов развития»::сеанс). Альтернативный вариант моделирования ситуации — проект это рабочая группа проекта (организация) оргразвития, включающая переучиваемых или вновь обучаемых людей, а также работников службы оргразвития. Тут опять-таки будут сложности с тем, что там сервис (оргразвитие::сервис::поведение) и кто над кем трудится. По этой линии (проект как организация) есть сложности моделирования (которые можно преодолеть по линии «программа проектов и портфель проектов — это программа и портфель организаций», но разговаривать в таком языке будет сложно), проще считать проект экземпляром работы по выполнению метода (это классика дисциплины project management, по этой линии проблем не будет).

Так что онтика сервиса/метода вроде как одна и та же для самых разных ситуаций, но всегда будут нюансы, поэтому будет меняться не только терминология, но и сами типы объектов в онтике.

Моделирование: сервисы в компании

Опишите от пяти сервисов в вашей компании:

Номер п/п	Сервис	Сеанс	Сервис-провайдер	Предмет сервиса: заготовка	Предмет сервиса: готовая система	Целевая система сервис-провайдера	Надсистема целевой системы сервис-провайдера

Ты — член команды

Целевая система обычно выявляется для команды большого проекта и становится известной для координации работ внутри проекта. При этом, конечно, допускается, что большая компания может выпускать несколько видов продуктов и их не удастся представить одной целевой системой, это не слишком меняет ход рассуждений.

Ошибка в том, что целевая система выявляется индивидуально в каждой маленькой команде, а уже эти маленькие команды пытаются связаться в один большой проект. Получается обычно так, что если одна команда заявила, что её целевая система «печень», другая — «кишки», третья — лекарство от болезней печени, то они потом будут пытаться объединиться в какой-то проект, но это будет трудно: печень окажется свиная, кишки — полевой мыши, лекарство будет от болезней лошадиной печени, а проект вдруг окажется лечением человека. Нет, тут делается всё-таки ход на то, что целевой системой будет здоровый человек, на входе — человек с больной человеческой печенью и дальше будет выстраиваться граф создателей, исходя из этой целевой системы. Ветеринарные подразделения будут или перепрофилированы, или выкинуты из проекта за ненадобностью — сколько бы они ни настаивали на том, что они системно выявили свои целевые системы. Свои системы — это не целевые, считать свои системы целевыми обычно в проекте могут не слишком много людей, у остальных это — эгоизм, попытка перетянуть одеяло на себя.

Напомним: целевая система вводится не как что-то «объективное», а для того, чтобы скоординировать работу в сложном графе создателей, чтобы было понятно, чем там все эти люди, AI, организации занимаются, что там должно появиться в результате проекта.

Системное мышление будет координировать в проекте работы самых разных агентов в самых разных ролях по самым разным методам, а не просто давать мыслительное превосходство одних агентов над другими. Это не боевое/спортивное мышление соревнования и победы (хотя выигрывать в конкуренции оно помогает), но мышление договорённостей и сотрудничества.

Вы как «член команды»/«внутренняя проектная роль» совсем необязательно заняты работой над целевой системой в её целостности. Вы можете быть заняты разработкой подсистемы для целевой системы, или даже подсистемой подсистемы, или даже созданием какого-то создателя (инструмент или даже оргзвено с людьми, AI и инструментами) в графе создателей — но это будет просто ваше личное мнение, что вы занимаетесь «собственной целевой системой», вы можете оставить это мнение при себе.

Вы занимаетесь «вашей системой»: это, конечно, по типу тоже система, но просто не целевая система, а важная для вас и вашей команды. В рамках проекта, в общении с вашей командой вы будете обсуждать вверенную вам подсистему или даже подсистему подсистемы, или даже создателя какой-то подсистемы в составе целевой, или даже клиентуру как систему. Вы вполне можете называть такую вашу систему «моя система», это же никого не смутит, это чистая правда. Но не надо требовать от команды признания вашей системы (система, которая только для вас engineered system, маленький винтик в большом аппарате, или пять подпрограмм в огромном программном комплексе, или даже вся фирма, которая занята выпуском целевого продукта/изделия или оказанием сервиса для чужих целевых систем) полноценной целевой системой для всех. Нет, вы должны делать три вещи:

- Уверенно называть целевую систему. Если вы не понимаете, что там целевая система, то что бы вы ни принесли в проект, чем бы ни занимались — это может оказаться ненужным. Поэтому сначала давайте поймём, что все вместе делаем в полном проекте по созданию целевой системы, а только потом

будем разбираться с вашими проблемами вашего мини-проекта.
- Уверенно называть вашу систему, и тут всё то же самое, что для целевой, только речь идёт о вас или о вашей маленькой команде маленького проекта в составе большого проекта.
- Не забыть о том, что ваш маленький проект нужен только для того, чтобы была успешна целевая система. Поэтому потрудитесь показать причинно-следственную связь с успехом целевой системы того, что вы делаете и чем заняты.

Так, у топ-менеджера может быть три проекта: А, Б, В. И он неделю убил на разбирательство с проектом Б. На это время он называет «своей системой» проект Б. Проверяем:

- Уверенно ли он называет целевую систему, для которой работает в конечном итоге проект Б? Или он исторически занимается проектом Б, который уже вообще никому не нужен?
- Что там его система? «Проект Б» — это поведение рабочей группы проекта? Тогда проект не система, с самого начала ошибка. Меняем понимание: система этого менеджера — организация: команда проекта Б, «рабочая группа проекта». Это уже лучше, но если это топ-менеджер, то лучше бы он своей системой для проекта считал бы продукт/изделие для проекта Б, а если речь идёт о сервисе, то его система — это та система, состояние которой меняет своим сервисом команда проекта Б. Нацеленность должна быть сначала на результат, а не на создателя, а на создателя — после понимания ожидаемого результата! Отлично, вот это уже похоже на системное мышление.
- И вот тут третий пункт: надо показать причинно-следственную связь того, что вот прямо сейчас занятие нашего менеджера проектом Б как-то поможет выпуску целевой системы. Или качество целевой системы будет лучше, или цена станет дешевле, или снимутся возражения каких-то внешних ролей для всего проекта (а не системы проекта Б), или скорость выпуска целевой системы вырастет? Для этого нужно понять, где в графе создателей находится система этого менеджера и выдать причинно-следственную связь, рациональное обоснование решения. Если окажется, что ничего в целевой системе не поменяется, то менеджер выделил время на ерунду, безделицу, хобби. Он должен обосновать, почему не работал над А или В, а работал над Б. Для этого можно обратиться к обоснованию со стороны теории ограничений (операционный менеджмент, в руководстве по рациональной работе вы знакомились с современной версией этой теории, TameFlow[357]). Тревожный знак тут будет, что менеджер, занимаясь Б, подтвердит, что никакого немедленного эффекта его действия на целевую систему не дадут, а он делает «заделы». «Заделы» — это вред, это омертвление капитала, это work in progress, и операционный менеджмент с этими «заделами» борется, это привнесение рисков в работу. Вот эта **связь «нашей системы» с «целевой системой» по цепочкам отношений создания, развития (в том числе проектирования, изготовления, изменения состояния и т.д.) и часть-целое — она должна предъявляться, а выделение времени на занятия «нашей системой» (одной из всех возможных систем проекта) — обосновываться. Почему занимались нашей**

[357] https://tameflow.com/

системой, а не какой-то другой, не помогали соседям, например? Если связь не предъявляется, а потраченное на нашу систему время не обосновывается, то вы не можете сказать, пользу приносит эта работа, или вред.

Вы можете быть также заняты разработкой системы создания (например, реализовывать проект развития какого-то оргзвена: парикмахерской), то есть непосредственно сами не будете заниматься целевой системой. Но это не значит, что вы будете называть парикмахерскую целевой системой во всём проекте. Во всём проекте вам нужно создавать причёски, а в вашем проекте — создать парикмахерскую. Не нужно путать эти два проекта, считать их одним. Вам нельзя считать, что целевая система для вас это та, которая для всех остальных будет системой создания. Нет, ваша целевая система та же, что у всех, а именно, которая изменяется работами систем создания, в том числе теми системами создания, которые делаете вы в графе создателей, то есть вашими системами. Вы в проекте создания причёсок, вот и объясняйте, как ваши действия с парикмахерской что-то дают для создания причёсок. Например, увеличивают выпуск причёсок в уже действующей парикмахерской, или приближают выпуск причёсок в ещё не введённой в строй парикмахерской, или дают возможность выпуска более выгодного типа причёсок, или удешевляют процесс выпуска причёски, и т.д. — везде тут важно, что речь идёт о причёске, и если причёски в рассуждениях нет, то к вашей работе по улучшению парикмахерской сразу будут вопросы.

Если организуете авиазавод, то целевая система — самолёт, если консультируете организаторов авиазавода, то лучше бы вам считать целевой системой самолёт, чем этих организаторов! Самолёт, который летает (в момент эксплуатации) — вот целевая система, поэтому расскажите, как вы добиваетесь того, чтобы самолёт летал, и летал хорошо, и был дешёвым. Если консультируете организаторов завода — как ваши консультации помогут самолётам? Если никак, то и без ваших консультаций всё останется как было — разве что вы консультируете не организаторов авиазавода, а, например, владельцев каких-то пакетов акций, которые они хотят перераспределить. Но тогда сам авиазавод тут не при чём, речь идёт о консультировании по поводу инвестпроекта, и нужно честно назвать вещи своими именами — эта честность поможет всем.

Как в картах, каждый человек не строит карту с полярными координатами вокруг себя. Договорились в масштабах планеты о Северном полюсе как начале координат — и все на планете пользуются единой системой отсчёта, полюс никуда не скачет по прихоти какого-то одного человека или даже одной команды путешественников, он общий для всех. В системном подходе рекомендуется именно такой подход: договориться об общей целевой системе для всей большой команды проекта. И всем агентам договариваться о совместной деятельности, показывая причинно-следственные связи их работы с успешностью целевой системы.

Команда проекта — это совсем необязательно группа проекта под твоим руководством! Скорее всего, «ты» в этом «мы» занимаешь пару-тройку ролевых позиций, делаешь часть большого дела, не удерживаешь в сознании целую систему, а занимаешься либо какой-то отдельной практикой/культурой/видом труда (теплотехникой, менеджментом, программированием, хореографией), либо какой-то частью системы, либо клиентурой системы. Если у тебя часть целой системы, или система в графе создателей — это будет «наша система» для маленькой команды

или «моя система» для тебя лично.

По поводу неё вполне можно и нужно развернуть полное системное мышление, вот прямо как про целевую систему, разве что не надо считать её целевой и называть целевой. Первый шаг тут — найти настоящую целевую систему всего проекта (как обычно в системном мышлении: смотреть не в детали, внутрь, а от границ — наружу, в данном случае из границ «нашего проекта» наружу, в большой проект создания целевой системы). Целевую систему надо выявить, чтобы определить относительно неё место «нашей системы» — и дальше договориться со всеми по поводу нашей системы (квалификация «практик» в организационном развитии) и даже договорить всех (квалификация «мастер» в организационном развитии).

Нужно тщательно выявить ту целевую систему, с выбором которой будут согласны все участники проекта, синхронизироваться в действиях по отношению к которой будут все, и дальше разбираться уже с той системой, которая к тебе ближе всего (ну, или с той системой, которая для проекта сейчас нужней всего — если ты топ-менеджер, и ещё обосновать, почему ты занимаешься какой-то выбранной одной системой, а не соседними).

Большая ошибка — это считать себя объективным и нейтральным системным инженером, системным менеджером, эдаким системным рыцарем, который весь в белом и с нимбом над головой ведёт команду к успеху проекта. Вовсе нет, всё не так. **Системное мышление чаще всего нужно именно для того, чтобы поднять свою осознанность: позволить осознать, какие у тебя роли в большом проекте по созданию и развитию целевой системы, какими методами ты работаешь с какими предметами этих методов, какое мастерство тебе нужно и инструменты, какие твои ролевые интересы, какое твоё место в большой команде и полномочия по распоряжению ресурсами большой команды, какой подсистемой целевой системы твоей большой команды или подсистемой одного из многочисленных создателей в графе создателей целевой системы ты занимаешься.**

Тут нет никаких рецептов, всё определяется текущей ситуацией, никаких алгоритмов на эту тему нет — но есть набор объектов мета-мета-модели, которые задают список важных объектов проекта, о которых вам надо подумать и дальше договориться в масштабах как большой команды всего проекта по созданию целевой системы (например, механических часов), так и вашего частного проекта (например, проекта создания и развития стрелок для механических часов).

Команда проекта может быть и на пару тысяч человек сотрудников предприятия, а может состоять и из пары человек, и даже иногда из одного человека. Хотя в случае одного человека думать системно — это может казаться перебором, но поглядите на множество внешних проектных ролей, поглядите на множество создателей::системы (например, оргсистемы-создатели: ваши контракторы-подрядчики, поставщики инструментов и оборудования, провайдеры всевозможных сервисов), и вы не будете считать, что вы такой уж одинокий в проекте якобы «на одного», даже если вы один будете исполнять все многочисленные инженерные и менеджерские внутренние проектные роли, выполнять все нужные для успеха проекта методы работы. А с учётом внешних проектных ролей мышление в проекте всё равно будет коллективное, а не «личное», с внешними проектными ролями придётся договариваться и ещё договаривать их между собой.

Разделение на внешние проектные роли и команду может быть даже более дробным: внутренние проектные роли *предприятия* (юрлица) как «большой команды» и команда оргзвена (отдельного проекта или подразделения внутри *предприятия*) как «малой команды». Внутренние проектные роли предприятия как большой команды (например, финансисты, юристы, операционные менеджеры предприятия) заинтересованы в успехе целевой системы, но активно преследуют совсем иные интересы, нежели малая команда проекта для её «нашей системы». Если у вас в команде появятся лишние $1000 в проекте — вы куда их потратите? Это непростой вопрос:

- Улучшите на эту сумму целевую систему, не увеличивая её стоимости — т.е. подарите клиенту?
- Улучшите ресурсы команды — наймёте более квалифицированных людей, закупите более дорогое оборудование?
- Отдадите на дивиденды владельцам предприятия? Потратите на нужды «малой команды»?

Интересы всех ролей разные, причём эти интересы могут быть к разным системам — и действия по преследованию этих интересов поэтому оказываются разными, часто очень изобретательными, неожиданными для окружающих. Лучше бы вам всем в проекте договориться перед тем, как эти действия начнутся, или начнётся активная реализация самых неожиданных стратегий самых неожиданных агентов. Но для этого надо понимать, что «договориться в проекте» — это договориться по поводу улучшений в отношении выпуска целевой системы. Если ваша связь с целевой системой непонятна, то вы чужак, доверия к вам нет, договорённости с вами имеют слабую силу — и будут немедленно нарушены, если выяснится, что для успеха проекта в целом (успешности целевой системы) надо заниматься чем-нибудь другим.

Обычно много желающих сделать заведомо провальный проект, ибо исполнители ролей в этом проекте хотят получить зарплату, а результат самого проекта их мало волнует. Актёры с удовольствием сыграют свои роли в пустом зале: то, что за этот спектакль театр не получит деньги, их мало волнует (дирекция всё равно заплатит), а то, что зрителей нет, так ещё лучше: можно халтурить в исполнении ролей! То же происходит с заведомо провальными проектами. **Системное мышление помогает осознать, что проект в части успешности целевой системы провальный, а затем настаивает на этическом рассмотрении ситуации. «Наш проект» может быть не провальный! Но этично его не делать, если он не ведёт к успеху целевой системы.**

Нужно коллективно/группой договориться, что считать целевой системой, какие цели преследует проект, в чём состоят возможности проекта. И ещё нужно понять, какой именно коллектив/группа договаривается о том, чтобы создать и развивать целевую систему, это ведь тоже часто сходу непонятно. Но сначала нужно таки разобраться, что там за целевая система, а потом уже с «мы»: если непонятно, что мы делаем, то непонятно, кто вовлечён в это делание, кто такие «мы», кто с каким мастерством (и даже с каким уровнем мастерства, это ведь тоже важно) в проекте потребуется, а кто просто пришёл покушать/попилить ресурсы, а не вести целевую систему к успеху, и не вносит никакого вклада, или даже приносит вред.

Каждые пять минут менять то, что ты/вы называешь/называете целевой системой — это неправильно. **Релятивизм** (когда всё на свете считается субъективным и относительным, зыбким и неопределённым — заведомо отсутствует точка отсчёта, любой ученик пятого класса уважается в его мнении по поводу строительства атомной электростанции, любой дикарь уважается в его выборе целевой системы, да ещё и каждые пять минут все они меняют своё мнение, ни с кем не договариваясь) в мышлении не поможет. Договорённость по поводу целевой системы должна достигаться из каких-то принципов, а не произвольно, и на достаточно долгое время — желательно не менять целевую систему во время всего проекта, хотя понимание того, что это за целевая система, может уточняться постоянно. Конечно, в бизнесе рассматривается и вариант pivot/вираж, когда стартап меняет целевую систему. Но не обольщайтесь: во время этого «пивота» все понимают, какой новой системой они будут заниматься. И не все «мы» в этих проектах переживают вираж.

Моделирование: целевая система и ваша система

Заполните табличку для проектов создания трёх систем, в которых вы принимаете или принимали участие.

Номер п/п	Проект	Целевая система	Ваша система	Основная ваша роль в проекте вашей системы и метод, которым работает роль	Как связан проект вашей системы с успехом целевой системы

Признаки целевой системы

Целевая система выявляется/discover ситуативно, о ней договариваются командно, нет никакого алгоритма или «типовых ситуаций». Тем не менее, есть **эвристики** (правила, которые срабатывают часто, но не всегда[358]), которые помогают найти целевую систему какого-то коллективного проекта среди многочисленных самых разных систем в самых разных вариантах системных разбиений. Да, в системном

[358] https://en.wikipedia.org/wiki/Heuristic

мире царит вроде как хаос из совокупности самых разных субъективных мнений, но человек склонен упорядочивать и хаос, находить в нём паттерны — символ хаоса совсем не случайно выглядит весьма структурированно:

Надёжного алгоритма выявления целевой системы нет, вы просто будете делать много догадок и затем критиковать их, чтобы принять всерьёз ту догадку, которая пережила критику. Принять всерьёз — это действовать так, как будто эта догадка и есть верная. И так действовать, пока не появится догадка получше — тогда быстро бросать старую догадку и переходить на новую, действовать в реальном мире, опираясь на эту новую догадку. Есть множество приёмов мышления, которые помогут быстрее формулировать, а затем оценивать догадки по поводу того, что же в вашем проекте будет целевой системой.

Слово «**выявление»** (discovery) тут использовано не случайно: считается, что целевая система есть, она задана типом мета-мета-модели «целевая система»::система и её нужно только найти/выявить своим вниманием, определить её границы — возможно, поговорив при этом с большим количеством самых разных исполнителей самых разных ролей. Нас тут не смущает, что речь идёт о будущем — в пространстве-времени эта система уже существует, только в будущем отрезке времени, не прямо сейчас, это не мешает описывать целевую систему и её роль в надсистеме уже сегодня. Не смущает и то, что «поговорить со всеми» — это необязательно узнать их мнение, можно и наоборот — договориться с ними, что они согласны с вашим мнением о целевой системе. Вообще, неважно, кто именно высказал хорошую догадку о том, какова целевая система в вашем проекте. Это можете быть вы, это может быть какой-то AI, это может быть какой-то участник проекта. Главное, чтобы догадка была высказана, а потом она пережила критику.

Концепция использования как описание функциональности «чёрного ящика» системы в надсистеме — её не «выдумывают», не «разрабатывают», её как раз выявляют, договариваются по поводу того, что там будет с функциональностью системы, причём обязательно это документируют. Неправильно использовать вместо «документируют» термин «фиксируют», т.е. закрепляют описание. Концепция использования — это догадка, она будет уточняться и корректироваться до самого конца проекта, в модели системы как «чёрного ящика» всё время будут вноситься изменения, ничего «фиксированного» нет!

Договориться по поводу целевой системы, чтобы потом по отношению к ней

договариваться обо всех других важных системах, в том числе о «нашей системе» — это подготовить концепцию использования, в которой описать целевую систему как вписанный в надсистему «чёрный ящик», удовлетворяющий интересы внешних проектных ролей. Основное в концепции использования — это сценарии использования, то есть поведение целевой системы в разных ситуациях, которые предъявляет ей окружение. А уж как целевая система будет устроена — это потом, это будет уже концепция системы. В концепции использования становится понятным, что в мире относится к целевой системе, а что попадает в её окружение в момент использования. Решения о том, как будет устроена система (концепция системы) и как будет устроено её создание и развитие (решения по составу и структуре графа создателей) логически принимаются потом, хотя обычно все они разрабатываются параллельно: концепция использования учитывает пределы возможного (чтобы не оказалось, что это концепция какой-нибудь скатерти-самобранки или ковра-самолёта, которые непонятно как сделать и непонятно кто будет изготавливать), но всё-таки вопрос ставится как «сначала опишем, что система должна делать, то есть какие объекты в окружающем мире в какое состояние приводить, а затем — из чего её делать и как она должна быть устроена, затем — какими методами её делать, и только затем — у кого есть мастерство и инструменты для поддержки таких методов, которыми систему можно сделать».

Например, вы хотите поручить каким-то другим людям-подрядчикам создать какую-то систему в вашем проекте (инструмент создателя в вашем графе создателей целевой системы или просто подсистему целевой системы), чтобы не делать её самостоятельно, то есть хотите купить уже готовую систему или попросить разработать и сделать систему «под заказ». Вы формулируете и документируете, что именно хотите купить у подрядчика, это и есть формулируете концепции использования этой заказываемой системы (раньше на основании этой концепции делались требования, в них часто отфильтровывалась и искажалась информация концепции использования — тот самый «испорченный телефон»). Для вас это будет система-инструмент создателя в графе создателей вашей целевой системы или просто подсистема целевой системы (или даже всё это не целевой системы, а вашей системы), а вот для подрядчика — это будет его целевая система! И он должен будет разобраться не в «изготавливаемой им системе» (время изготовления), а в «используемой вами системе» (время использования, и ровно это вы в своей концепции использования системы для подрядчика и прописываете).

На основании концепции использования вы делаете закупку. Сегодня обычно это не разовая и окончательная закупка, но предполагающая сопровождение — постоянные улучшения системы после момента её создания, продажи и начала эксплуатации. То есть предполагается, что вы не продукт покупаете, а работы сервиса по улучшению продукта, мир становится не продукт-ориентированным, как раньше, а сервис-ориентированным.

Целевая система находится в физическом мире, это точно не какое-то описание — хотя если вы типография или художественная мастерская, то вашей целевой системой вполне может книга или картина, но вас тогда не будет волновать описанное в книге или картине (документированное описание), но только «документ», а уж что там за описание — это проблемы авторов и читателей, не ваши. Но если вы не типография, и вас интересуют работы с содержанием информации (о чём документ: книга, картина), а не носителем информации (из какой бумаги), то сразу смотрите на содержание, а не на носитель информации —

систему надо искать в том, что описано, система не ваше описание! Если вы говорите «моя система — проектная документация», то вы курьер, или владелец принтера. Если вы инженер, который занят проектной документацией — то первый же кандидат в рассмотрение будет та система, которая описана в документации.

Если вы делаете какие-то информационные модели, пишете тексты, создаёте базы данных (например, разрабатываете софт), рисуете схемы или чертежи, то это вы обычно делаете только описание системы, воплощённое в физическом мире как документация системы/системная документация, но не как сама система! Описание, даже документированное — не целевая система! Задайте себе вопрос: а что описано в документе, в базе данных, в информационных моделях, отражено на графиках и диаграммах, описано в текстах? Возможно, там описано другое какое-то описание (текст, таблица, диаграмма, описывающие другие тексты, таблицы, диаграммы — объекты ментального/понятийного мира, имеем дело с «описанием описания»), но, возможно, там будет описана ваша целевая система, или хотя бы её подсистема, или создатель из их графа (всё это — системы физического мира).

Уберите внимание к носителю информации (компьютер с софтом, сам софт), обратите внимание на данные этого софта: там наверняка будут подсказки, где искать целевую систему в физическом мире, эти данные описывают какие-то системы. Если данные софта описывают процесс сборки самолёта, то это описание поведения самолётного завода (время создания, описаны операции создателя — сборка::метод самолёта::«предмет метода»), но если у вас самолётный завод, то вы обязательно вспомните о самолёте (это же будет «предмет метода» для сборки) — и дальше вы просто задаёте вопрос о времени использования самолёта: понимаете, что его место не на заводе самолётов, а в небе, чтобы лететь куда-то с грузом и пассажирами. Ради этого «самолёт, который летит с грузом и пассажирами» всё это и делается: завод самолётов, а для завода делается описание процесса/метода сборки самолёта. Целевая система самолёт, завод — создатель в графе создателей самолёта, платят люди-потребители не за завод, а за сервис со стороны авиакомпании, самолёт — инструмент авиакомпании как провайдера сервиса, а для завода — это целевая система (он выпускает инструментарий авиакомпаний, ему платят за это).

Если вы говорите, что «система находится в физическом мире» — и показываете на носитель информации, то вам поверят только, если вы производите сам носитель, а не содержательно занимаетесь информацией на нём! Целевая система «книжка как документ» не у писателя, а у типографии или интернет-магазина электронных книг. А у писателя? Если это инженерный проект, то это было бы воплощение фрагмента мира, описанного в книге. Форма книги тут была бы не важна, а вот принятые по устройству этого фрагмента мира решения — важны. Если это образовательный проект, то можно было бы говорить о том, что создаётся при помощи книги участок мозга ученика (специализированная «мокрая нейронная сеть») как исполнитель какого-то метода работы. Этот участок мозга потом будет проявляться как провайдер сервиса мышления (или выполнения какого-то другого метода работы, с учётом задействования не только вычислителя как части мозга, но и тела, и даже инструментов как датчиков и эффекторов) в какой-то ситуации, для которой и было обучение этой части мозга — и целевая система тут не просто «участок мозга», но «участок мозга в момент его задействования для выполнения метода». Этот участок мозга в его функциональной роли вычислителя для выполнения метода работы мы и назовём «мастерство». А если брать всё

мастерство агента, выполняющее все методы, которыми владеет этот агент? Это будет личность агента (и кроме личности там будет ещё и организм/тело). Если у нас в руках учебный курс или руководство, то мы смотрим прежде всего не на их носитель (компьютер датацентра, мозг преподавателя/наставника, книжка), но на содержание курса/руководства/регламента — знания/теории/объяснения/алгоритмы, которые описывают какую-то предметную область. И там два кандидата в системы, которые мы ищем:

- Мастерство, для создания которого этот учебный курс и был создан
- Предмет метода, описанного в курсе (скажем, если курс/руководство/регламент по выращиванию клюквы — то это клюква), а курс мы рассматриваем как справочник по предмету метода — системе. Не все методы работают с системами, часть работает и с описаниями систем или даже описаниями описаний, но часть предметов метода — это таки системы.

Такой мыслительный приём/метод уподобляет проекты по изменению поведения людей через изменение «мозгового софта» проектам по разработке компьютерного софта (можно было бы назвать это «нейролингвистическим программированием», но этот термин уже был занят в середине 70-х годов прошлого века конкретным методом такого программирования, описанным Джоном Гриндером и Ричардом Бэндлером, и тем самым оказался недоступным для общего пользования[359]. И да, Джон Гриндер и Ричард Бэндлер использовали системный подход первого поколения, им в этом помогал один из видных системных мыслителей этого поколения системного подхода — Грегори Бейтсон[360]). Этот приём/метод нередок при моделировании изменений, проходящих с людьми (образовательные проекты, продвижение каких-то продуктов и сервисов, психотерапия и т.д.). И помним, что целевая система определяется на момент её эксплуатации: то есть на тот момент, когда мастерство как соответствующая функциональная часть мозга (реализуемая на сегодня малопонятными конструктивными частями, но вполне материальная!) будет задействовано в выполнении работы с тем методом, которому мозг был научен в ходе обучения/терапии/программирования/кодирования (да, использовалось и слово «кодирование»!) и т.д. Мы уже неоднократно упоминали в нашем руководстве, как думать о системах, которые не изготавливаются из отдельных деталей, а обучаются. Более подробно об этом будет рассказано в руководстве по инженерии личности, а учитывая то, что организация тоже учится работать новыми методами (хотя она условно «собирается» из отдельных людей, AI и их инструментов, но всё равно люди и AI в ней потом научаются работать по новым методам), то и в руководстве по системному менеджменту.

Что будет целевой системой в детском садике Монтессори[361]. Ребёнок? А родители тогда что делают/изготавливают (создают и развивают), его же, ребёнка? Детский сад сервис, который выполняет какой метод работы над ребёнком, который не выполняют его родители? Садик — склад для ребёнка, где его хранят в то время, когда родители работают? Эвристика: в детском садике Монтессори идёт учебный процесс, ребёнка учат. Это значит, что целевой системой скорее всего будет какое-то мастерство: функциональная часть мозга, которая будет включаться в

---

[359] Что происходило с нейролингвистическим программированием и подобными подходами можно посмотреть по ссылкам в тексте «психопрактики движения по шкале формальности мышления», https://ailev.livejournal.com/1461939.html

[360] https://ru.wikipedia.org/wiki/Бейтсон,_Грегори

[361] https://www.livingmontessori.com/montessori-kindergarten-vs-traditional-kindergarten/

определённых жизненных ситуациях, уже за пределами садика. А детский садик будет только провайдером сервиса обучения. Эксплуатация мастерства как вычислителя/части мозга — за пределами садика.

Скажем, ребёнка должны там научить вести себя по методу разумной осторожности (не прятаться всё время за родителей, но и не лезть на рожон в любое доступное ему пекло), читать стихи, не теряться в незнакомой обстановке, быть собранным и не терять внимания. Это означает, что мастерство (то есть подготовленный таким образом вычислитель как часть мозга) должно не просто на выпускном утреннике в знакомой обстановке актового зала учебного заведения выдать чтение стишка с табуретки, но уверенное чтение стишка на каком-нибудь фестивале, где ребёнка поставят на железную бочку, фотовспышки будут со всех сторон, родителей оттеснят куда-нибудь в сторону, но обученный в садике кусочек мозга включится, даст спокойствие, уверенность и прочтение того самого стишка. Это и есть мастерство как результат обучения, вполне материальный объект (реализованный конструктивными частями мозга). И это будет через год после выпуска, мастерство должно быть изготовлено (инженерная терминология! Она позволяет говорить более точно!) так, чтобы через год умение не терялось. Весь персонал детского садика Монтессори должен вот это всё понимать: что результаты их работы надо будет смотреть не в самом детском садике, а за его пределами, возможно, через год. Впрочем, примерно то же относится и к школе, и даже к вузу, хотя в вузе даётся прикладное образование и поэтому можно задавать вопросы о ситуациях, где применяется мастерство, полученное в вузе. Но вот со школой и детским садиком такое рассуждение сложнее: ответ обычно «мы готовим к экзаменам», а экзамены как раз можно рассматривать как часть учебного процесса, а не использование мастерства в жизни (если научили не тому, что надо, то и экзамен можно организовать по тому, что не надо — мастерства в методе, которому учили, не будет, но будет мастерство сдавать экзамены).

Зачем так подробно это расписывать? Теперь вы знаете, какая целевая система в вашем учебном проекте (выполнили творческую/исследовательскую/предпринимательскую работу, ответили на вопрос «чему учить», какому мастерству учить — и ответ более-менее детальный, с перечислением того, что именно должен будет уметь делать выпускник), и только после этого вы должны задавать вопросы к системе создания: какие методы могут создать/изготовить (в нашем случае — обучить) такую функциональную часть мозга? Как должно быть устроено обучение для того метода, которому выбрано учить? Учить ведь математике, выращиванию морковки и рассказыванию стихов на фестивалях нужно разными методами — и только после того, как выяснится, что учить нужно именно этому (например, что детей надо учить именно рассказывать стихи, тратить время их детства именно на это умение)!

И дальше обсуждать уже методы обучения и ситуации этого обучения, каждый раз задаваясь вопросом о том, как это влияет на результат: мастерство, которое будет проявлено вне ситуации обучения. И если школа говорит, что «мы учим детей сдавать ЕГЭ», то можно усомниться, что «мастерство сдачи ЕГЭ» является именно тем мастерством, которому должна учить школа. А после обсуждения вопроса о том, какое же это должно быть мастерство, можно обсуждать методы изготовления этого мастерства: методы изготовления мастерства сдачи ЕГЭ при этом могут оказаться совсем другими, нежели методы изготовления жизненного мастерства как какого-то набора прикладных видов мастерства и фундаментального мастерства

мышления, которые могла бы изготавливать школа.

Вот этот ход на то, что целевая система должна рассматриваться не в тот момент, когда она создаётся, а в тот момент, когда она эксплуатируется — он крайне важен и часто тут делаются ошибки. Причёска рассматривается в ситуации, когда идёт встреча с женихом, а не любая причёска на момент нахождения владельца волос в парикмахерской, когда там идёт стрижка и укладка и наблюдаются какие-то промежуточные состояния причёски. Шлиф рассматривается в момент его использования (то есть уже после шлифования и даже не в момент отдачи шлифа клиенту, а в тот момент, когда клиент уже установил деталь со шлифом в свою систему, и она работает). Все методы шлифования подбираются так, чтобы с этим использованием шлифа было всё в порядке. То же самое верно и для обучения людей, и для любых других проектных ситуаций. Вы должны дотянуться мыслью до ситуации использования/эксплуатации целевой системы, а когда договариваетесь про целевую систему со всеми остальными участниками проекта, то надо, чтобы и они признали целевую систему на момент использования, а не на момент её создания или продажи, или даже наладки. Нет, целевая система должна быть представлена в момент её эксплуатации/работы/функционирования/использования.

Целевую систему всегда придётся выявлять в ситуации недостаточной информации и разнообразия мнений на этот счёт. В каждом проекте она будет уникальна, справочника «типовых решений» и магического алгоритма её выявления тут нет, можно только давать примеры — в надежде, что эти примеры не будут восприняты как «делайте именно так», но нейронная сетка всё-таки заметит какие-то паттерны в этих примерах.

Системное мышление задаёт лишь набор типов, для которых надо найти в проекте объекты, чтобы обязательно подумать об этих объектах. Оно просто управляет вниманием в проекте, оно не указывает, как и о чём думать (указывает о чём думать в мета-мета-модели, но чтобы подумать хотя бы об объектах мета-модели, надо отождествить эти объекты с типами мета-мета-модели: выполнить выявление систем, разобраться с отношениями систем. Например, в учебном проекте понять, какое мастерство будет целевой системой — и дальше всех договорить по поводу этого мастерства). Если вам трудно принять решение по поводу целевой системы — это означает, что вы что-то плохо понимаете про ваш проект, и лучше бы вам подумать ещё, а также собрать больше информации, поговорить с исполнителями разных других ролей в проекте.

Документация (бумажная или электронная) — не целевая система, даже если речь идёт о толстой пачке проектной документации для высотного здания или даже если речь идёт об исходном коде компьютерной программы, или даже если речь идёт о каком-то сценарии городского праздника, или даже если речь идёт о какой-то компетентности в чьей-то голове. Возможно, целевая система — это то, что вы описываете на этом носителе. Возможно, вы описываете даже не целевую систему, а систему в окружении целевой системы, или подсистему целевой системы, или систему создания где-то глубоко в графе создателей. Главное тут — не путать системы и описания (даже если описания даны нам как документация). **Целевая система — это не описание! Целевая система чаще всего — воплощение того, что описано!**

Целевая система — это обычно то, что делает (то есть полномочно меняет сама, а не «влияет на то, что кто-то другой поменяет») команда проекта. Если завод выпускает конфеты, то конфеты — это и есть целевая система (при этом не эшелон конфет в момент заключения контракта, не ящик конфет в момент отгрузки магазину, не упаковка конфет в момент продажи, а вот прямо одна конфета, причём в момент её поедания и переваривания).

Если команда делает описание, по которому потом сделают систему — это непосредственное стратегирование (описываем методы/функции), а затем и планирование (кладём метод на ресурсы, превращаем в работу) изменения физического мира. Это не уговоры кого-то, чтобы он изменил свою систему, не «рекомендации аналитика», который обычно не имеет полномочий реализовать свои рекомендации даже для системы, которой занимается его команда. У команды есть какие-то полномочия по отношению к её системе, а также в силу участия в проекте целевой системы — по отношению к целевой системе. В отношении других систем (систем окружения, в том числе надсистемы, систем в графе создателей) полномочия команды сильно ограничены, влияние много меньше.

Целевая система — это очень часто то, за что команде в конечном итоге (после того, как система окончательно изготовлена и начинает успешно эксплуатироваться) платят деньги. Если команда делает информационную модель велосипеда, то в конечном итоге заплатит деньги потребитель за велосипед. Вот целевой системой тут и будет велосипед, деньги именно за это — информационная модель велосипеда тут просто описание велосипеда (даже если это документ, то есть описание на носителе). Но если команда делает алюминиевую раму велосипеда, и эта рама продаётся только в составе велосипеда (то есть продаётся велосипед, а не отдельно рама), то и рама не будет хорошо выбранной целевой системой. Она будет подсистемой целевой системы. Вместе с тем двигатель самолёта — хороший кандидат в целевые системы. Двигатель обычно (с небольшими модификациями) может поставляться для включения в состав разных моделей самолётов, двигателестроители работают в составе независимых более-менее автономных предприятий, поэтому двигатель — хороший кандидат. Но Роллс-Ройс продаёт авиастроителям не авиадвигатели, а часы работы двигателя, так что и тут возможны варианты: двигатель тут неразрывно связан с системами его обслуживания, и двигатель тут может быть только подсистемой.

Целевая система-продукт часто пересекает границу предприятия, идёт её «поставка» (delivery, при этом обычно в delivery включают и ввод в эксплуатацию, а не просто «логистику доставки на место»). Подсистемы обычно границу предприятия сами по себе не пересекают, сами по себе не разворачиваются и не вводятся в эксплуатацию, это с ними происходит только в составе целевой системы.

У разных компаний/групп/команд проекта могут быть разные представления о том, что такое целевая система, каковы её границы, какой её уровень в системном разбиении, и даже какое там вообще системное разбиение. У поставщика авиадвигателей целевой системой является двигатель. Но у завода авиалайнеров авиадвигатель — это заказная подсистема.

Если вы обнаружили, что целевой системой у вас является какой-то процесс/метод, то подумайте дважды или трижды, какая система осуществляет это поведение. Если отдых, то вашей целевой системой, скорее всего будет отдыхающий (причём созданный, то есть отдохнувший). Если жизнь — то живой агент, её проживший и

не жалеющий о процессе жизни. Если танец (помним о многозначности слова! Тут это — перформанс, танцевание), то зритель, который воспринял этот танец, а иногда и сам агент-танцор, который «зритель себя», это ведь всё ролевые рассмотрения, один агент может играть в ходе перформанса две роли — и танцора, и зрителя. Мероприятие (фестиваль, перформанс, совещание — какие-то события с некоторой протяжённостью во времени и понятным составом участников) ещё можно представить системой, там можно ткнуть пальцем в составляющие этого мероприятия, собрать их в одном месте и одном времени — и затем использовать. А вот метод — это не мероприятие, а чистой воды поведение, его не надо делать системой, вы не сможете ни с кем договориться, вас не будут понимать, вы не сможете «изготовить метод». Хотя разработать описание метода — почему бы и нет, но дальше изготавливать по этому описанию вы сможете только мастерство агента и инструментарий этого агента, а потом агент встанет в роль и задействует метод в своих работах. А вот «изготовить мероприятие» — сможете, участники проекта будут вас как-то понимать.

Если у вас какой-то «рабочий процесс» (сервис/метод), то скорее всего интересует результат, предмет метода. Часто этим предметом метода будет система, «наша» или даже «целевая». Следите за типами: целевая система не стрижка::процесс/метод, а причёска (стрижка — один из процессов получения причёски, другой — укладка), не шлифовка::процесс, а шлиф (физический объект, на который легко указать в пространстве-времени, причём на момент его работы, а не изготовления), не доставка::процесс, а доставка как коробочка в руках клиента в месте назначения. У танцевания нельзя просто показать его материальный результат как достижение ожидаемых состояний предметов этого метода (хотя зритель в состоянии «доволен» или «заинтересован» и танцор в состоянии «без травм» — это вполне себе может быть результат), а вот у «рабочего процесса» или чего-то подобного результат (предмет какого-то метода в каком-то заранее оговорённом состоянии) обычно можно указать, вот и опишите этот результат в материальном мире — это более вероятный кандидат на целевую систему.

Целевая система — точно не процесс, но она наверняка как-то себя ведёт в ходе функционирования/работы, то есть меняет состояния чего-то вовне себя. Если нашли какой-то глагол, важный для описания результатов (приведения каких-то объектов в ожидаемые состояния — самолёт летает, компьютер вычисляет, мастер задействует своё мастерство), задайте вопросы по этому глаголу: 1. кто себя так ведёт? и 2. для какого результата, какой объект будет приведён в какое состояние? Получите сразу двух кандидатов в целевую систему.

Ответы хорошо бы получить «в ролях»/функциональных объектах, а если речь идёт об агентах, то одним агентом могут выполняться множество ролей. Один агент может сыграть роль многих систем. Все эти рассмотрения множества ролей агентов — первый шаг к рассмотрению возможности разделения труда, что важно для инженерии систем создания. Но для начала нам надо не столько заниматься ролями, методами этих ролей, разделением труда, сколько искать целевую систему. Если я причёсываю себя, то я в роли парикмахера и клиента одновременно. Но можно и разделить труд: отдать роль парикмахера агенту, который умеет причёсывать меня лучше, чем я сам. А результат? Причёска, целевая система! С неё и надо начинать.

Если договориться о результате проекта (какая система в каком состоянии должна оказаться), то тогда можно говорить и о разных рабочих

процессах, к этому результату ведущих, разных сервисах разных создателей, которые помогут этот результат (целевую систему) получить.

Частые ошибки в выявлении целевой системы

Признаков целевой системы не так много, но всё равно агенты (люди, AI и даже целые их организации) делают ошибки, выявляя целевые системы своих проектов. Проблема в том, что **системное мышление не подсказывает, что является правильным выбором целевой системы и не подсказывает, что будет в окружении целевой системы. Но польза системного мышления в том, что системное мышление указывает на то, что это важнейший объект в проекте — и заставляет выбрать, не даёт от этого выбора уйти!** Как это понимать? Примерно так же, как вы заходите в огромный продовольственный магазин с пониманием, что надо купить «еды»: в магазине тысячи и тысячи продуктов, которые все подходят под понятие «еды» — но какая именно еда нужна вам в данный момент времени? Тип «еда» — это всё что угодно, что можно съесть. Но вам вот прямо сейчас что именно надо съесть? Три банки горчицы? Или сдобную булочку? Копчёную скумбрию? Гречку-ядрицу в расфасовке по 1кг пакетик, и потом есть даже без варки, потолочь в муку — и съесть сырой? Если вы не перейдёте на описание отдельных видов продуктов и не будете понимать, что потом делать с купленными конкретными продуктами (с каждым продуктом — что-то делать разными методами! Что-то кусать и жевать, что-то сначала подвергать «кулинарной обработке» разных видов, что-то пить), задача купить «еды» не будет решена. Вам надо определиться, что именно покупать, общий тип «еда» не подойдёт. Но это точно будет не табуретка, не автомобиль, не забор, ибо это всё — не типа «еда».

С системным мышлением примерно так же: для типа мета-мета-модели «целевая система» вам надо найти тип мета-модели для системы, который должен быть много конкретней, чем просто «целевая система», принадлежать уже к какой-то предметной области. Но это не должны быть объекты не типа «целевая система».

Мы уже писали о многих ошибках выявления систем, которые вы хотите объявить принадлежащими типу «целевая система моего проекта по созданию и развитию

этой целевой системы», но повторим ещё раз, чтобы под рукой был маленький чеклист того, из-за чего студенты вуза уходят на пересдачу предмета «Системное мышление» (а если три раза не пересдадут — вылетают из вуза, как не готовые учиться, и прецеденты такого были!), **а в рабочих проектах на пересдачу проекта уходят целые команды, и даже очень большие команды, и тоже вылетают в случае нескольких пересдач, в рабочих проектах не поощряют такого**:

- **Вы выдаёте описание системы за систему.** Программисты считают, что их система — исходный код, а не программа в момент её выполнения (когда все переменные программы хранят состояние программы!), и уж точно не служба предприятия (оргзвено: подразделение, рабочая группа проекта), которая выдаёт результат, требуемый от программы, и без этой службы программа этот результат выдать не способна. Но исходный код — это описание, это информация, это не физический объект, который может хранить состояние мира в момент его использования и менять состояние мира вокруг. Проектировщики и конструкторы считают, что это проектная и конструкторская документация (неважно, в электронной или бумажной форме). То же самое: «документация» — это носитель информации, а сама информация — это не система. Сценарист считает, что это сценарий спектакля или фильма, а не сам спектакль или фильм в момент его просмотра зрителями. Хореограф считает, что это его тщательно продуманный набор движений, который он показывает танцорам (сам хореограф, демонстрирующий этот набор движений, и есть «документ описания», видеофильм, записи в какой-то нотации хореографы делают крайне редко), а не тот перформанс/«танцевальная композиция», который потом станцуют артисты-танцоры, хотя и с тем, что перформанс тоже целевая система, но его описание — оно «главнее». Модельер данных считает, что его целевая система — это модель данных, а не данные, структурированные в соответствии с моделью (например, база данных, собранных по модели/схеме данных), да и тут не делается последнего шага: целевая система обычно описывается этими данными, она не сами данные! Всё это, увы, более чем типичные ошибки. Мысль о целевой системе нужно доводить до момента изменения реальности в ходе эксплуатации системы, до воплощений, а не до описаний изменения реальности! Описать — это не сделать, не изменить мир! При этом недостаточно даже использовать «проектирование для изготовления» (design for manufacturing), хотя и это тоже нужно, тоже важно. Проектирование должно быть прежде всего озабочено эксплуатацией, изготовление тут только промежуточная стадия. Описание чего-то, удобного для изготовления — это не изготовленное физическое что-то, что приносит свою пользу в момент эксплуатации! Нельзя считать, что «воплощением занимается другой человек, организация, инструмент, а я готовлю описание». Нет, готовите вы описание только для того, чтобы потом оно было воплощено. Вот этот воплощённый объект — это и будет ваша система или целевая система, даже если не вы лично своими руками её изготавливаете. Вы принимаете участие в коллективном проекте по изготовлению какой-то системы. И вот эта система — ваша (и всех остальных) целевая система. Описания — это не система. Но если вы работаете каким-то методом создания описаний, то это описание вполне может быть предметом метода. Некоторые предметы метода — системы, но не все. Часть из этих предметов метода вполне могут быть описаниями, методы работы задействуются для самых разных объектов.

- **Создатель/«система создания»/инструмент/«провайдер сервиса», как целевая система.** Это типичная ошибка менеджеров: принять организацию, создающую целевую систему, за саму целевую систему (и при этом ещё добавить «это моя целевая система» — добавочная путаница между «наша/моя система» и «целевая система всего проекта»). Ошибка ведёт к процветанию организации на короткий период, пока не кончатся деньги инвестора. Если такому менеджеру поручить построить авиазавод, то он построит авиазавод, который будет восхитительно работать и все рабочие там будут сыты и ухожены — но самолёты этого авиазавода летать не будут, ибо завод будет строиться как целевая система, центром внимания всего проекта и его участников будет именно завод. Выпускаемые заводом самолёты не будут в центре внимания этого менеджера и менеджер не будет выделять достаточно ресурсов для выпуска качественных самолётов, он будет озабочен качественной заводской жизнью, а самолёты для него будут лишь цифирками «выпуска продукции». Типичный и обычный результат такого подхода — денег на компьютеры для финансовой службы менеджером выделяется больше, чем на компьютеры для инженерных расчётов целевой системы. Справедливости ради нужно отметить, что инженеры часто делают обратную ошибку: не замечают систем создания, без которых целевая система невозможна (то есть у них системное мышление ещё первого поколения, до важности обсуждения создателей и методов их работы они в мышлении не дошли), но об этом в нашем руководстве мы будем говорить позже.
- **Одинаковые имена целевой системы и её части** (или целевой системы и надсистемы), скорее всего нужно уточнить какое-то из использованных имён, обозвав иначе неполную систему (скажем, «дом состоит из коробки/строительной части дома, коммуникаций, отделки» говорить можно, а «дом состоит из дома, коммуникаций и отделки» говорить нельзя. «Ячейка состоит из корпуса и прокладки» говорить можно, а «ячейка состоит из ячейки и прокладки» — нельзя).
- **Одинаковые имена для самых разных систем.** Помним, что «командировка» — это могут быть 1. Удалённо делаемое дело, включающее деловую поездку к месту выполнения дела. 2. Деловая поездка как всё, что происходит в этой поездке. 3. Оформляемая часть деловой поездки как только то из деловой поездки, что можно провести по бухгалтерскому учёту (скажем, поездку на поезде — можно, но если пришлось ехать на такси — вот это уже нельзя, в оформляемой части деловой поездки этой поездки на такси не будет, а вот в деловой поездке — будет). Кто-то считает «мы продаём смартфоны» как «продаём смартфон с блоком питания», а кто-то — «продаём смартфон без блока питания». Термин «смартфон» тут будет обозначать разные системы (с блоком питания и без) с разной ценой, разным весом, разным поведением в момент зарядки.
- **Слишком высокий системный уровень (отсылка «на деревню дедушке Константину Макарычу»)**: игнорирование ближайшей надсистемы при определении целевой системы. Хвост у стада, кирпич в доме, целевая система «лампочка» неожиданно работает не в комнате, а в «умном доме». Маскируется словами «используется везде» (отсылка к каким угодно надсистемам), или совершенно неопределённым адресом. «— Где используется ваш сверлильный станок? — В машиностроении!», «— Какие танцы вы будете принимать к

постановке? — Любые!». Если видите в качестве надсистем названия отраслей, министерств, стран (это огромнейшие надсистемы, слишком крупные!) — бойтесь таких надсистем. Бойтесь слов с кванторами всеобщности и утерей специфики предметной области («любой», «каждый», «всякий», «универсальный», «многоцелевой» и т.п.). «Изготовлено в Китае», «используется в Сербии» — это очень плохие отсылки.

- **Сверхобобщение (указывается какой-то надкласс вместо конкретного класса систем).** Ошибка — вместо «стиральной машины» или «трансформаторной подстанции» говорить и писать «оборудование», вместо горчицы и жареной курицы говорить и писать «еда». Если вы понимаете, что сказанули (да, не «сказали», а «сказанули») термин из учебника, который относится и к системе из вашего проекта, и к тысячам других систем, которые на систему из вашего проекта ни разу не похожи — это ошибка. «Моей целевой системой является оборудование» — за такую общую фразу надо увольнять сразу, это чистая правда, но бесполезная правда, демонстрация неумения мыслить. Рассказывать об отношениях в мета-мета-модели не надо, это уже рассказано в наших руководствах. Расскажите об отношениях объектов вашего проекта, вашей предметной области! Если в вашей речи и документах не появляется терминов вашей узкой предметной области — вы просто не мыслите, не владеете операцией присвоения типа. Сказать, что «наше предприятие выпускает изделия» — это ничего не сказать!

- **Релятивизм (ошибка всех тех, кто не уверен в принадлежности к какой-то команде).** При релятивизме что угодно определяется «целевой системой», игнорируя наличие своей команды, своего собственного оргзвена, у которого может быть какая-то «наша система», совсем не целевая. Системное мышление в этом случае перестаёт служить целям координации коллективной/групповой деятельности. Если вы соглашаетесь, что «любая система в нашем проекте может быть целевой» — то это плохое понимание рекурсивного системного рассмотрения (по поводу любой системы можно провести системное рассуждение). Если у вас проект по созданию целевой системы, то укажите, для какой целевой системы (и это должно быть указание на работающую/функционирующую уже целевую систему, время её использования) в проекте объединяются все остальные системы! Да, целевая система — тоже система, но все остальные системы проекта — они собираются в проекте именно для её создания и развития. И надо договориться, для какой именно системы всё собрано, не удастся просто соглашаться, что «все системы важны, давайте всем подряд заниматься». Нет, заниматься надо не всем подряд, а тем, что ведёт к успешности целевой системы — одной для проекта.

- **Целевая система как все самые разные виды систем всех самых разных проектов, которыми занимается агент (чаще всего — крупное предприятие, у которого много видов продукции).** Это очень частая ошибка топ-менеджера, к которому попало очень много проектов. Это очень похоже на ситуацию сверхобобщения. Если к вам попал проект по изготовлению домашней электромясорубки (одна целевая система) и новогодних фейерверков (другая целевая система) просто на основании того, что какой-то собственник купил две компании, или по другим каким-то причинам, не пытайтесь изобразить их как целевую систему. Так и оставляйте их двумя разными целевыми системами! Не сваливайтесь в ошибку сверхобобщения, ибо закончится тем, что

вы обзовёте якобы целевую систему «оборудованием» — и дальше всё будет плохо, потому как что хорошо из вариантов методов работы для электромясорубок, не будет хорошо для фейерверков и наоборот, за малым исключением. Выделяйте из вашего всеохватного графа создателей его части для проектов создания разных систем, честно называйте эти системы как отдельные — и потом уже обсуждайте, выгодно ли (и кому именно выгодно) иметь большую общую часть графа создателей разных целевых систем, или не очень выгодно и лучше бы избавиться от части проектов, потому как легко попасть в ситуацию jack of all trades, master of none («занимался всем, мастер ни в чём»). Всё-таки сначала надо определить, какую систему ты делаешь, а затем трезво оценить, можешь ли ты эту систему делать — или тебя просто взяли «на слабо» и поручили заведомо проигрышное дело, а ты взялся.

- **Игнорирование первичности основной функции/метода работы системы.** То есть система выявляется не по её основному действию в окружении (метод/функция::поведение, которым она меняет объекты своего окружения в составе надсистемы в момент эксплуатации), а по каким-то иным соображениям. Например, вы пытаетесь сказать, что какая-то группа активов::«физические предметы», принадлежащая одному собственнику — это ваша целевая система. Отношения часть-целое вроде как есть, «актив в составе группы активов», но взаимодействия в такой группе и взаимодействия с системным окружением нет, новых свойств от объединения активов в группу не появляется, то есть — не система, нет системного эффекта. Например, «информационная система предприятия» очень часто определяется именно через отношение принадлежности к предприятию, а не по основной функции этой информационной системы, и это ошибка. Сказать, что «информационная система предприятия информирует» — это ничего не сказать про её роль/функцию/назначение в надсистеме (скорее всего, этой надсистемой тут будет предприятие в целом!). Это слишком общая характеристика. Пример с собакой обычно все понимают в части соблюдения этого принципа: нужно говорить «сторожевая собака», а не «домашнее животное Василия» (отношение собственности) или «дворовой зверь» (сверхобобщение и отсутствие роли в надсистеме). Но вот метод «сторожит» (а не «красиво выглядит», как у пуделя) — это явное указание на то, что можно ожидать от собаки. Указывайте на то, что можно ожидать от системы, а не называйте любым словом. Для какой-то инфраструктурной или программной системы часто упускается эта необходимая первичность рассмотрения функциональности целевой системы, «что она умеет», протягивание мысли к эксплуатационному времени (что она умеет изменить в окружении), а не ко времени создания системы или даже выходящими за рамки времени проекта отношениям владения.

- **Пропущенная система.** Отдел работы с инвалидами составляет для них учебную программу, эта учебная программа передаётся в учебную часть вуза, учебная часть вуза посылает преподавателей преподавать инвалидам по спецпрограмме (ибо они часто не в силах идти по общей программе). Целевая система тут, похоже, обученная функциональная часть мозга инвалида (назовём её «мастерство»), которая будет включаться в тех ситуациях, для которых инвалида учат. Но кто учит инвалида, какая система создаёт его мастерство? Бригада преподавателей, которую посылает к инвалиду учебная часть! Вот это оргзвено и есть провайдер обучения как сервиса создания мастерства («станок», инструментарий учебной части/деканата, но можно рассматривать учебную

часть/деканат как организатора/создателя для этого преподского оргзвена, это уже не так важно). Сообразить, что группа преподавателей практически неотличима от бригады программистов, которые программируют какое-то приложение — очень трудно, ибо посылаемые отдельно учебной частью преподаватели из общего пула вузовских преподавателей не воспринимаются как оргзвено. И напрасно: вы пропустили существование важного предмета обсуждения, пропустили систему. Она тут не целевая (целевая — это мастерство), но без её обсуждения нормального графа создателей не получишь.

- **Флоты, пулы, агентуры, клиентуры.** Ещё часто пропускают какой-нибудь «флот»::«активные транспортные средства — электровозы, тягачи» или «пул»::«пассивные системы — вагоны, тележки» как отдельную систему со множеством других систем как её части. И даже агентура как какое-то множество агентов и «клиентура» как множество всех клиентов. Взаимодействие частей такой системы тут неочевидно, но оно устроено хитрым образом — через совместную обработку, совместное описание, совместную конкуренцию за ресурсы, совместное создание (общий граф создателей). Более того, есть эмерджентные характеристики (например, для клиентуры важен её рост, этой характеристики нет у отдельных клиентов, а вот суммарное число клиентов характеризует клиентуру в целом и может интересовать рост клиентуры, скорость роста клиентуры, ускорение роста клиентуры — и создание и развитие клиентуры создаёт не только отдельных клиентов как подсистемы клиентуры, но и клиентуру в целом, которая и может быть одной из систем в проекте, хотя целевая система обычно — это то, что или приобретается клиентурой у нас как производимый нами продукт/изделие, или принадлежит клиентуре, если у нас сервис и клиентуре надо выполнить для своих предметов метода работы того метода, в котором у нас есть мастерство и инструменты). Тут очень спорный вопрос, считать ли такие «коллективные системы», как клиентура, полноценными системами системного подхода, но современное системное мышление всё ближе и ближе к так называемому «конструктивному» определению системы — и система определяется не просто как существующая в мире объектов и отношений, но как результат операций по построению такой системы из частей, а описание системы — результат операций по построению описаний агентом-создателем. И если будут новые свойства на каждом системном уровне — то всё ОК, а взаимодействие систем друг с другом рассматривается опосредовано — через агента-создателя, проводящего какие-то операции создания. Эту линию «конструктивизма» в мереологии (учения о частях и целых) проводит Kit Fine[362], мы расскажем об этом подходе как о современном тренде в системном подходе в последнем разделе нашего курса.
- **Указание на какую-то связанную с целевой системой, вместо той, которая целевая.** Например, вы говорите «заказ Уралнатрия» вместо указания системы. Потом выясняется, что заказов у Уралнатрия множество, они меняются, а систему уже сделали и перепродали Сибирькалию в совершенно другой комплектации. Хорошо бы понимать, что «заказ» — это не система, но это может быть указанием на договорённость поставщика и заказчика о сделке. А сама система — это не «заказ». Обычно всё это проявление метонимии: косноязычия

[362] Kit Fine, Towards a Theory of Part, 2010, https://as.nyu.edu/content/dam/nyu-as/philosophy/documents/faculty-documents/fine/accessible_fine/Fine_Theory-Part.pdf, Kit Fine, The Identity of Social Groups, 2020 https://metaphysicsjournal.com/articles/10.5334/met.45/

людей, когда они вместо одного объекта называют другой объект, связанный какими-то часто не очень определёнными отношениями с тем объектом, который они пытаются назвать. Если вы выявили системой «заказ Уралкалия», то сколько людей поймут, что вы имеете в виду состояние вашей компании и Уралкалия, которые вошли в сделку по изготовлению чего-то неизвестного (слово «заказ» может подразумевать заказ чего угодно — свиней, башенных часов, проходческих комбайнов для солевых шахт, обучения пары тысяч сотрудников китайскому языку — заказ/сделка/договорённость), а сколько человек будут понимать, что это названа целевая система — скажем, тот же проходческий комбайн, который и будет целевой системой, и который поминается в «заказе», причём в заказе будут ещё и деньги, и сроки, и много чего ещё, но вот сама целевая система — это как раз вот этот проходческий комбайн для солевых шахт, считать «заказ» целевой системой — явная ошибка.

- **...** множество других ошибок, проявляющихся во множестве других ситуаций. Повторимся, что системное мышление просто определяет вам объекты внимания, но само оно не пошаговое/алгоритмичное, не даёт чёткой формулы получения результата. Вам надо догадаться, сделать гипотезу о том, что такое у вас целевая система, а вот набор понятий системного подхода дальше даст материал для критики этой гипотезы. Опровергнуть ваши идеи системное мышление может, а вот сгенерировать идеи — нет. Нет строгой последовательности действий, которые дадут «правильный» ответ. Правильных ответов вообще не бывает, бывают полезные и бесполезные (или даже вредные) ответы на вопросы системного мышления, которые системное мышление признает прошедшими его критику. Вот эти бесполезные и вредные ответы на вопросы, которые системное мышление не признаёт — это и есть ошибки. Какие вопросы? **Первый же вопрос системного мышления: что у вас целевая система? На него нельзя не отвечать!** И лучше бы не делать частых ошибок.

Например, возьмём сверхобобщение: указание системы так неконкретно, что её невозможно найти среди всех объектов окружающего мира. Можно считать, что называя систему, вы даёте её адрес в понятийном пространстве. Адреса составлены обычно так, чтобы разные уровни адресации позволяли чётко найти объект, объект уникален внутри какого-то уровня адресации, его легко найти — это и есть **принцип почтальона** (был предложен А.Нечипоренко в ходе одного из наших семинаров в Новосибирске).

Так, адрес пластиковой накладки на ножке стола по сути перечисляет объекты в системном разбиении, позволяющие перейти от внимания ко всей Вселенной к вниманию к этой пластиковой накладке:

- Вселенная,
- комплекс сверхскоплений Рыб-Кита[363],
- Ланиакея[364],
- сверхскопление Девы[365],
- местная группа галактик[366],
- Млечный Путь[367],

[363] https://ru.wikipedia.org/wiki/Комплекс_сверхскоплений_Рыб-Кита
[364] https://ru.wikipedia.org/wiki/Ланиакея
[365] https://ru.wikipedia.org/wiki/Местное_сверхскопление_галактик
[366] https://ru.wikipedia.org/wiki/Местная_группа
[367] https://ru.wikipedia.org/wiki/Млечный_Путь

- Рукав Ориона[368],
- Солнечная система,
- планета Земля,
- Россия,
- Ростов-на-Дону,
- ул. Большая Садовая,
- дом 105,
- аудитория 323,
- первый стол от доски у окна,
- левая ножка стола,
- пластиковая накладка.

Адрес «Вселенная, Солнечная система, Дом 105, пластиковая накладка» — это неправильный адрес целевой системы, неправильное указание на целевую систему, ибо «домов 105» много, непонятно какой именно дом в Солнечной системе вы указываете, а в каждом доме множество пластиковых накладок — непонятно, какую из них вы имеете в виду.

В системном разбиении надсистема должна определяться по возможности близко по уровням отношений «часть-целое» от целевой системы, быть в ближайшем окружении. Тут возможно много разных типов ошибок, они всё те же, но повторим:

- **Пропуск уровней.** Обычно заявляется слишком далёкий от целевой системы системный уровень, например, вместо целевой системы называется её надсистема, или даже система на несколько системных уровней выше надсистемы. Типичный пример: я делаю автомобиль! После чего следует длинный рассказ о морозоустойчивых аккумуляторах. Почему? — Ну, меня интересуют морозоустойчивые аккумуляторы, они используются в автомобилях. — Могут ли использоваться, например, для гайковёртов? — Да, вполне. — А если гайковёрты в Якутии? Да, это много лучше автомобиля, для моих морозоустойчивых аккумуляторов это отличный рынок! Я занимаюсь гайковёртами! (после некоторых вопросов становится понятным, что и аккумулятор тоже не интересует, речь идёт о морозоустойчивом электролите — он и будет целевой системой! Проект занимается морозоустойчивым электролитом, хотят производить его, а аккумуляторы пусть производят все желающие покупатели электролита!).
- **Сверхобобщение.** Это в некотором роде нарушение адресации, но отношение не часть-целое, а уровня обобщения/специализации в классификаторе. Например, не «одноместный учебный самолёт», и даже не просто «самолёт», а «летательный аппарат». Что ты создаёшь? Летательный аппарат! А в реальности это игрушечный самолётик на резиновом моторчике. Ну и назови его по обычной роли «быть игрушкой»: «игрушечный самолёт». Верно ли будет «летательный аппарат»? **Для математика — верно! Но у нас тут не математика, у нас системное мышление. Нам все названия и результаты размышлений нужны для работы, для создания и развития успешных систем, а не для «логической истинности». Правильный в математике или физике ответ не означает, что он правильный в системном мышлении!** Для понимания, что можно сделать с создаваемой системой, имя «игрушечный самолёт» много лучше

[368] https://ru.wikipedia.org/wiki/Рукав_Ориона

«летательного аппарата».

- **Указание не всей системы, или соседней** (в окружении или даже в числе создателей) системы, в надежде, что целевая система будет собеседником определена самостоятельно, умозаключением. По сути, речь идёт о **метонимии**[369] — один физический объект (или класс таких объектов) заменяется другим (или его классом), находящимся в какой-то связи/отношении с называемой целевой системой. И вместо названия/указания одной системы появляется вдруг название этой другой системы. Связь эта может быть как «часть-целое», так и любой другой. «Я три тарелки съел» — не тарелки, а находящуюся на тарелке еду. «Ведро расплескалось» — не ведро, а вода в ведре. В случае метонимии целевой системой могут назвать что угодно, находящееся в каких-то отношениях с целевой системой: всё, что попадает в сферу внимания. **Проверять метонимию нужно, отслеживая, что речь идёт именно об отношениях «часть-целое» и учитывая время рассмотрения (эксплуатация или создание),** и уже дальше уточнять, какая система в итоговом системном разбиении целевая, на каком системном уровне мы с ней работаем, каким языком мы о ней говорим.

Не расстраивайтесь, если вы поначалу будете постоянно совершать все эти ошибки, даже зная, что это ошибки (обычно тут помогает наставник, который будет на разборах ситуаций вам на эти ошибки указывать: увы, знание про ошибки не убирает эти ошибки из моделей, для исправления собственных ошибок нужно внешнее «зеркало», изнутри себя ошибки обычно не видны). **Профессионализация в каком-то деле — это то, что вы перестаёте делать новичковые ошибки. Конечно, вы будете делать и другие ошибки, но уже не новичковые.**

Беглость и безошибочность в системном мышлении (лучше всего — мышлении моделированием, но можно и письмом, что хуже в силу меньших усилий на формализацию, ещё хуже — просто мышление рассуждением вслух, ибо у результатов мышления не будет задействована внешняя по отношению к ненадёжному мозгу память), как и беглость и безошибочность в любом другом деле (игре на музыкальных инструментах, разговоре на иностранном языке и т.д.), приобретается только путём долгих занятий, путём набора опыта работы.

В ходе работы по предусмотренному нашим руководством моделированиям и заданиям по написанию постов вы заполняете множество табличек и пишете множество текстов — и постепенно набираете какой-то опыт системного мышления. Продолжайте создавать таблички, похожие на приведённые в заданиях нашего руководства, модифицировать их, заполнять на материале ваших рабочих проектов — и **скоро вы не будете понимать, почему это было поначалу так трудно. Это и есть метанойя, когда вы не можете понять, почему раньше всё казалось сложным, трудным, а сейчас вдруг вы системно мыслите — и даже не замечаете этого! Нейронные сети в биологическом мозге обучаются медленно, но всё-таки обучаются!** На какое-то (не до уровня чемпионов, конечно) освоение системого мышления до хорошо заметной в работе квалификации нужно затратить на работу по нашим руководствам примерно год по паре часов в день. Это даже быстрее, чем освоение высшей математики, танцев,

[369] И опять напомним, это ведь важное слово: https://ru.wikipedia.org/wiki/Метонимия

карате!

Повторим оценки времени: тут примерно как с походами в тренировочный зал, если вы решили вдруг заняться своей физической подготовкой. За три месяца упорных занятий результаты будут едва заметны, за год — вы разительно будете отличаться от «людей с улицы», за десять лет — можно разговаривать о чемпионстве. То же с танцами. Системное мышление не трудней спорта или танцев, но и не легче. Так что — за три месяца вы поймёте, в чём его суть, за год научитесь бегло использовать, а лет через десять будете одним из лучших в мире, если продолжите им упорно заниматься. Увы, труд наставника по системному мышлению можно автоматизировать, а труд инженера-менеджера по освоению системного мышления автоматизировать нельзя, придётся работать!

Моделирование: проверка на ошибки выявления целевой системы

Вы уже заполняли эту табличку по целевым системам от трёх рабочих проектов, в которых вы участвуете. Выполните моделирование ещё раз, но с учётом материала текущего раздела.

Номер п/п	Название проекта	Целевая система	Функция/назначение	Надсистема	Сколько проверок (на физичность, на рассмотрение во время эксплуатации, на учёт разветвлённого графа создателей и т.д.) вы сделали при выявлении целевой системы. Посчитайте тут число проверок из чеклиста предыдущего раздела — сколько пунктов вы прошли (сделайте эти проверки!)	На сколько процентов вы уверены в вашей гипотезе о целевой системе после всех этих проверок

Именование системы

Мало выявить целевую систему, ей надо дать какое-то осмысленное имя, чтобы вас понимали быстро. Если я назвал целевую систему Муркой, то это может быть что-то живое и крупное (вряд ли таракан), и даже речь идёт о самке (кошки или человека — не так важно, но вряд ли это будет самец), при этом я буду подозревать, что это имя экземпляра, но не тип целевой системы. Но вот если я назвал систему #78F3, то мне придётся всем рассказывать — это я назвал тип или экземпляр, а также что это вообще такое: зверь, электронный гаджет, территория, или не система вообще, а только описание системы, например «документ #78F3».

Система обычно именуется по узкому типу её основной роли, выполняющей какую-то основную функцию/сервис. Так её именует команда, занимающаяся созданием и развитием надсистемы для этой системы, а если её именует производитель продукта, который не очень хорошо понимает, каким именно аффордансом выберут его продукт, то обычно он даёт самое общее название ожидаемой роли, для выполнения которой систему купят.

Если речь идёт о создателе и его сервисе (методе, работы по которому создатель будет выполнять для экземпляров предметов метода, которые ему будут давать другие создатели), то именовать этот создатель свой предмет метода будет в наиболее общем виде так, как это будет делать владелец заготовки целевой системы, для которой работает заказчик сервиса. А себя этот создатель будет именовать тоже по роли сервис-провайдера. Например, вас интересует сервис маникюра[370] — приведение ногтей в эстетичное состояние. Именовать вы свой сервис будете как «маникюр» (косметическая процедура, поведение провайдера маникюра», создатель «красивых ногтей» будет именовать себя как провайдер сервиса — «маникюрный салон», предмет сервиса тут — «ногти», переводимые из состояния «неухоженные» в «ухоженные», в сленге это будет или «идти на маникюр» (сеанс сервиса) или даже «идти на ногти» (именование не по сервису, а по предмету сервиса, но это всё-таки просторечное).

Так что идёт ли речь о создателе, или какой-то целевой системы — именуем по роли исполнителя функции/сервиса, которыми система меняет своё окружение (функция в надсистеме, сервис для окружения), а не конструкции (то есть не по отсылке ко времени создания, название ссылается на использование!). Указание конструкции иногда важно, но главным образом как уточнение названия по роли исполнителя функции/метода/сервиса/практики и добавляется только как указатель на какие-то архитектурные характеристики — то, что потенциальному потребителю системы (внешняя проектная роль, которая захочет использовать вашу целевую систему либо в составе своего инструментария где-то в графе создателей уже своей целевой системы, либо как подсистему в составе своей целевой системы) покажется важным, чтобы принять решение о том, что ваша система ему подходит. Но:

- Сначала система должна подойти по её роли: назначению/функции, основному методу её работы (первичное рассмотрение системы — как функционального объекта, то есть роли в надсистеме)
- И только потом — по конструкции, которая влияет на архитектурные характеристики (надёжность, ремонтопригодность, экономичность и т.д.).

Иными словами, система именуется как ролевой объект, если вы её используете. Но

[370] https://ru.wikipedia.org/wiki/Маникюр

если вы её производите, то в производимой системе вы тоже пытаетесь предугадать её функцию в ожидаемой вами надсистеме, указать её роль (по типу), хотя иногда допускается добавление уточнений про конструкцию, чтобы в названии отразить архитектурные характеристики (они зависят от конструкции!).

Если у вас в автомобиле коробочка, которая связывает разные электронные устройства (замок зажигания, спидометр и т.д.) с интернетом (какими-то серверами в облаке), то как её назвать? По типу, причём можно в основание типа положить тип-прототип — какой-то иной, хорошо знакомый тип систем, на который ваша система как-то похожа.

В доме устройство, которое связывает разную домашнюю электронику с интернетом называется роутер/router. Отлично, для автомобиля связь автомобильной электроники с интернетом будет делать автомобильный роутер! Всё, тип (маршрутизатор пакетов данных от каких-то локальных устройств в «большой интернет») определён, и вы дальше сэкономите много-много минут объяснений того, что это у вас за целевая система. Основное назначение: маршрутизация/routing::метод «пакетов данных»::предмет метода, уточнение — что эта маршрутизация будет делаться в автомобиле. Провал проекта будет, если вы назовёте этот автомобильный роутер «автомобильной электроникой», «интернет-аппаратурой автомобиля» или чем-нибудь другим, где не просматривается явно функция: что именно там с интернетом, автомобилем, электроникой — если нет глагола, непонятно что делает, нет традиционного имени типа системы, из которого понятно назначение, если в имени суперобобщения типа «электроника» или даже «интернет». Если это «роутер/маршрутизатор, вроде того, что у вас дома соединяет домашние компьютеры и телефоны с интернетом, только для автомобильных компьютеров», то вы за две минуты будете объяснять, чем занимаетесь и команде проекта, и внешним проектным ролям. В каждом разговоре будете экономить минут так пять-десять — на протяжении всего хода проекта. Выявление и точное называние целевой системы для этого и нужно: быстро и надёжно скоординировать коллективную работу по проекту.

По основному назначению/функции/сервису, как ролевой объект именуются не только целевые системы, но и все остальные (надсистема, системы в окружении, подсистемы), т.е. именование системы оказывается именованием её как функционального объекта, роли, «чёрного ящика» заданного назначения, а не прозрачного ящика с известной конструкцией или методом работы, и это именование связывается с поведением, игрой некоторой типовой роли. Вот эта типовая роль и будет именем системы. Так что именование системы зависит от ожидаемого окружения системы. Сервис-провайдера и инструмент тоже стремятся назвать как какое-то обобщение ожидаемой роли и функции его как системы в надсистеме.

Так, микроскоп предназначен для того, чтобы быть в роли инструмента для рассматривания мелких предметов: смотреть::метод микро/мелкие объекты::«предмет метода». Если иногда им заколачивают гвозди, то это не считают его основным назначением, хотя это иногда и будет его поведением/функцией в надсистеме. Это редкое назначение, особенность некоторых вариантов использования. Поэтому микроскоп нельзя называть «микроскоп с возможностью гвоздезабивания» или «молотковый микроскоп». Нет, только по основному назначению. Но если вы берёте микроскоп в вашу систему именно как гвоздезабиватель, то так и назовите!

Название «продавца» и название «покупателя» могут различаться: ожидаемая функция системы и включение в название системы как конструктивного объекта для выпячивания возможной роли в архитектуре клиентской системе обычно делаются продавцом и выражают ожидание возможного использования. Скажем: «водонапорный насос». А вот покупается функциональный объект, и он именуется по актуальному использованию — «насос заполнителя бака третьего контура пожаротушения на пятом этаже». Если покупаете микроскоп (имя, данное производителем производителя!) как гвоздезабиватель (имя, данное потребителем по актуальному использованию!), то не удивляйтесь двум именам.

Именуется при этом класс/тип систем, а не уникальный экземпляр системы. И это очень часто ролевой объект, «исполнитель метода работы с определёнными предметами метода» (сборщик собирает предметы сборки, погрузчик грузит грузы, нарезчик нарезает), но не только. Назначение «ножниц» (что-то резать, это производное от двух «ножей») хорошо известно, так что ножницы назовут, скорее всего, ножницами — но добавят уточнения по предмету метода резания ножницами-инструментом (например, «ножницы по металлу»).

Помним, что сверхобобщение при выборе термина для типа — это большая ошибка. Микроскоп нельзя называть «оптический прибор» (так, телескоп — это тоже оптический прибор, и будет путаница), ножницы лучше бы тоже уточнить — парикмахерские, или по металлу. **Уточнить/специализировать — это хорошо, обобщить/генерализовать — плохо.** Если вы назвали систему «фрукт», а не «апельсин», не удивляйтесь, если будут страшно ругаться, обнаружив, что вы продали не персик. Ибо персик — это «фрукт», и покупатели будут считать, что «фрукт» исполнит любую роль в этой категории. Если не любую — будут совершенно справедливо ругаться. Так что называйте, как всегда, точно, не обобщайте. Если купили «зверя», ожидая «тигра», то не удивляйтесь, если вам продана «мышь» — тоже зверь, что просили, то и получили!

Самая частая ошибка, которую делают при именовании — это перечисление всей

возможной функциональности длинным списком и ещё добавление конструкции. Вместо «автобус» это будет «система посадки, перевозки по автомобильным дорогам, оповещения об остановках и коммуникации водителя и пассажиров, высадки пассажиров и закрывания дверей в целях безопасности, усиленная галогеновыми фарами». Так нельзя: любое длинное перечисление функций в названии недопустимо. И иногда есть уже устоявшееся название, его и нужно использовать — тот же «автобус». Если это тип, то всем и так будет понятно, что делает такая система.

Другая не менее частая ошибка — это указание не на ролевой объект, т.е. на действия/функцию в надсистеме (роль, выполняющая метод: система «сварщик» для выполнения метода «сварка», система «зубодёр», если она выдирает зубы — вот так надо!), а сразу на конструкцию. Конструкция обычно относится к времени изготовления системы, «из чего и как мы эту систему делали». Вместо «ножницы по металлу» будет «система металлическая с винтиком», явная ошибка. Вместо «антистатик для пластмасс» (метод снятия электрического заряда «антистатик» и предмет метода — пластмасса) будет «система введения углеродных трубок в пластмассы» (из чего делали антистатик — из углеродных трубок, назначение трубок умолчано, ошибка раз, а вторая — говорим даже не о самих трубках, а о «системе введения», то есть инструменте-системе с сервисом «введение» для нанотрубок, то есть продаём не антистатик, а вроде как инструмент для введения какого-то непонятного субстрата «нанотрубки» в пластмассы — но зачем?!). Нет, название системы даётся по первичному/обычному/типовому назначению системы, по роли/функциональности в типовой надсистеме для такого сорта системы, какую вы именуете. И не путаем — продаём систему, сервис, инструмент для какого-то метода, что? Имя должно быть точным по типу!

Про нарушение «принципа почтальона» (прежде всего — указание слишком общего ролевого объекта, слишком общего действия/функции) мы уже не раз говорили, но повторим. «Система разрезания» вместо «ножницы по металлу» или «автотранспортное средство» вместо «автобуса», «фрукт» вместо «апельсина». Роль должна быть специфицирована точно, без лишнего обобщения. Вот прямо сейчас попробуйте придумать какую-то абсолютно другую систему, которая отлично подойдёт под имя, которое вы дали вашей целевой системе. Трудно придумывается? У вас хорошее имя! Легко придумывается? Для вашего тигра, которого назвали «зверь», вы легко придумали и мышь, и зебру? Тогда меняйте имя вашей системы на тигра (а то и амурского тигра, чтобы не путать с какими-нибудь другими тиграми).

Нельзя говорить «расчёты», нужно указывать, какие именно расчёты, что именно эти расчёты считают (не «расчёт», а «расчёт среднесрочных рисков» или «расчёт даты готовности заказа», или расчёт прочности — и не просто расчёт прочности, но с указанием предмета этого метода, например, расчёт прочности на разрыв ленты шахтного конвейера).

В названии целевой системы должна чётко угадываться предметная область из системного уровня надсистемы, никаких сверхобобщений до уровня «инженерия вообще», «IT-вообще», «менеджмент вообще».

Особенно отличаются игнорированием предметной области программисты: у них всё «расчёты», «сервисы», «базы данных» без указания на то, что они делают в прикладной предметной области (не в области программирования, а в той области,

для которой они программируют!). Нет слов из предметной области, в которой работают эти «расчёты» и «сервисы» — что-то названо не так, налицо сверхобобщение, ибо какая-нибудь «система обработки рисков» может работать, судя по названию, с рисками чего угодно, но в жизни это оказывается глубоко не так: риски в ней могут быть только те, для каких она разработана, но не другие. Уточните, проведите DDD[371], слово «domain» (предметная область) для прикладной предметной области там не случайно!

Ещё можно указать на неоправданное использование слова «система» в названиях: «система ножниц по металлу» лучше сразу заменять на «ножницы по металлу». Проверьте, нет ли слова «система» в названии вашей целевой системы — и немедленно его исключите, назовите вместо типа мета-мета-модели («система») именем из предметной области, с которой работает эта система.

Вопросы. Именование системы

Система медленно ползёт по грядкам, берёт овощи спецзахватом, отрезает хвостики и кладёт их в ящик. Ящик запечатывается и маркируется, укладывается на тележку. Как можно было бы назвать эту систему?
--- сельскохозяйственная система ### Нет, это слишком общо —вряд ли эта система занимается всем, чем занимаются в сельском хозяйстве, явно нарушен «принцип почтальона».

--- автоматизированная фермерская система ### Нет, это слишком общо — вряд ли эта система занимается всем тем, чем занимаются фермеры, нарушен «принцип почтальона».

--- агро-робот ### Нет, это слишком общо, функциональность непонятна, в агропромышленности слишком много видов деятельности/методов, чтобы робот делал их все.

--- система сбора овощей ### Нет, хотя, это названо по основной (обычно самой дорогой и труднодостижимой, если их много — и тут это точно не отрезание хвостиков) функции. Но всё-таки можно короче: слово «система» как очень общее название типа лучше из названия исключить.

+++ сборщик овощей ### Да, это названо по основной (и обычно самой дорогой и труднодостижимой — точно не запечатывание и маркировка ящиков) функции. Название лучше, чем «система сбора овощей» — короче на одно слово.

--- система сбора, упаковки, маркировки и погрузки овощей ### Нет, перечислены все функции, а нужно назвать только основную.

--- овощепогрузчик ### Нет, это не основная самая трудная функция. Сбор овощей много трудней их погрузки.

[371] https://en.wikipedia.org/wiki/Domain-driven_design

--- овощной робот ### Нет, это слишком общо, непонятно, что делается (функция/поведение) с овощами: переработка в консервы? Нарезка? Выбраковка гнилых овощей, которые уже кем-то собраны?

Система собирает по сети данные о днях рождения людей, отождествляет их с фотографиями, печатает индивидуальные поздравительные открытки и отправляет их обычной почтой. Оплачивает всё это благотворительный фонд, идея состоит в поднятии настроения людям, что должно спасать их от всех болезней. Как фонд мог бы назвать такую целевую систему?
--- система поднятия настроения ### Нет, это указывает лишь на сервис (что система делает с людьми с точки зрения её создателей-инженеров, их лучшее приближение для функции). Хороший кандидат на название, если бы мы не знали, что называет систему фонд, а не её создатели. Для фонда это система спасения от всех болезней, название по её основной функции. И ещё в названии слово «система», от него хорошо бы избавляться.

--- система спасения от всех болезней ### Нет, хотя это основное назначение системы с точки зрения фонда: спасать людей от всех болезней. Но можно короче: от термина «система» как слишком общего типа в названии стоит избавляться

--- система спасения от болезней через улучшение настроения ### Нет, лишнее уточнение про механизм работы. И слово «система» тут лишнее.

--- социальная поздравительная система ### Нет, это только про механизм работы для достижения основного назначения/выполнения основной функции системы с точки зрения фонда — спасения от всех болезней.

--- система сбора и распространения информации ### Нет, не отражена основная функция с точки зрения фонда — делается система не для сбора и распространения информации, а для спасения людей от всех болезней. Ну, и слово «система» тут явно лишнее (можно было бы назвать «сборщик и распространитель информации», хотя это всё ещё неправильный ответ).

--- поздравитель с днём рождения ### Нет, это просто роль исполнителя стратегии, которая была выбрана в качестве метода улучшения настроения. Основное тут — не поздравление с днём рождения, а спасение людей от всех болезней. Метод можно потом в ходе развития поменять (спасать от всех болезней не через улучшение настроения), а вот предмет метода — болезни (состояние: их нет), в названии системы не отражён.

--- система печати и рассылки открыток ### Нет, это указание на довольно низкоуровневые сервисы для достижения основной функции с точки зрения фонда: спасения людей от всех болезней.

+++ спасатель от всех болезней ### Да, это роль системы, названа по роли для основной функции — спасение.

--- благотворительная открыточная система ### Нет, это никак не указывает на то, что же именно делает система, какова её роль. И ещё слово «система», от него надо бы избавиться. Обычно «система самолёт» и «система кошка» не говорят, надо прямо назвать роль.

--- благотворительная система поздравлений с днём рождения ### Нет, это указание на вариант реализации инструмента для сервиса улучшения настроения, задействованного для достижения основной функции с точки зрения фонда: спасения людей от всех болезней. И ещё слово «система» тут явно лишнее.

На поверхность детали наносится напыление из трёх слоёв, дающее повышенную износоустойчивость и коррозионную стойкость детали. Как можно было бы назвать это напыление как систему?
--- трёхслойное напыление ### Нет, пропущена основная роль, указывающая на функцию, то есть «что система делает», а не «что система такое» — повышение износоустойчивости и коррозионной стойкости.

+++ упрочняющее и антикоррозионное покрытие ### Да, именно так: покрытие (роль) упрочняет и защищает от коррозии, это основные функции данного покрытия.

--- система износоустойчивости и коррозионной стойкости ### Нет, хотя функция указана верно, но в качестве роли лучше бы выбрать что-то более конкретное, чем «система», например, «покрытие» (хотя даже конструктивный вариант «напыление» тоже будет лучше, чем общее слово «система», оно «ни о чём»).

--- трёхслойная система износоустойчивости и коррозионной стойкости ### Нет, указание на конструкцию из трёх слоёв тут лишнее.

+++ упрочняющее и антикоррозионное напыление ### Да, отличное название (можно и «покрытие», но напыление дополнительно может указывать на тонкость покрытия, что может быть архитектурно важно).

--- поверхностная наносистема ### Нет, слишком общее название — основная функция не определена, поведение/назначение неясно, да ещё и слово «система», хотя и «нано».

--- улучшающая наносистема ### Нет, слишком общее название — основная функция не определена, поведение/назначение неясно. Ещё и слово «система» очень общее.

--- защитная наносистема ### Нет, слишком общее название — тут хорошо бы указать, от чего защита. И убрать слово «система».

Надсистема: её тоже нужно выявить!

Надсистема (иногда говорят using system, «использующая система») прежде всего должна проверяться на то, что целевая система является в момент эксплуатации (operation, работы, функционирования) неотъемлемой физической частью этой системы, то есть буквально входит в состав (composition) надсистемы. Увы, большинство ошибок происходят именно по невниманию к этому её признаку.

Системное разбиение (system breakdown structure) — это иерархия по отношению состава/сборки/часть-целое, а не по каким-то другим отношениям. На ошибку выделения надсистемы и целевой системы указывают обнаружение между ними отношений вызова подпрограммы, классификации и специализации, принадлежности в части имущества, назначения на роль, создания — ведь есть огромное количество других отношений, кроме предписанного для системного разбиения отношения состава (composition, is_part_of, включения как физической части).

Например, рассмотрим мужчину и женщину. Какая у них надсистема? Да, они как-то взаимодействуют, у них есть какая-то функциональность в составе надсистемы. Какой? В студенческих аудиториях эту загадку решают минут пять, а в более взрослых аудиториях инженеров-менеджеров на это тратят пятнадцать секунд: надсистемой тут является семья (студенты часто говорят «пара», подчёркивая возможность более слабой связи), а женщина и мужчина находятся друг у друга в операционном/эксплуатационном/рабочем окружении, это и есть «в семье».

Загадки выделения надсистемы обычно решаются путём выявления пропущенной, не названной, не выявленной системы в системном разбиении — и выделенной системе надо дать имя (часто имя типа такой системы уже есть, вроде той же «семьи», но просто сразу оно почему-то неочевидно).

Если над целевой системой ещё как-то думают, то надсистемы для неё (их может быть несколько!) обычно плохо различимы, и иногда для них нет устоявшихся названий. Эти «пропущенные» системы приходится выявлять (то есть описывать их роли, функции, границы к их окружению, делая ещё один шаг «наружу» от целевой системы) и как-то называть. Как называть? Точно так же, как любую другую систему, т.е. преимущественно (если нет какого-то явно устоявшегося

исторического названия) по роли этой надсистемы в её надсистеме и выполняемой этой ролью функции/методу/сервису, т.е. по роли в над-надсистеме по отношению к целевой и функции/методу/сервису с её предметами сервиса в над-над-системе. Просто рассматриваем сначала роли от границы любой системы вверх по системным уровням, а роли подсистем — потом.

Признаком того, что какая-то часть целевой системы всё-таки относится не к самой целевой системе, а к надсистеме (а команде проекта обычно приходится иметь дело с обеими этими системами) является то, что команда не уполномочена как-то самостоятельно в одностороннем порядке изменять надсистему или даже её описания/модели (концепцию использования, концепцию системы, архитектуру надсистемы). А вот целевую систему и её описания команда изменять может, вполне полномочна делать это самостоятельно!

Снова и снова мы видим, что разница между системами разной природы пролегает не по их свойствам как физических объектов, а по каким-то договорённостям людей: системы систем (SoS, определяются через разную принадлежность автономных систем, входящих в систему систем), службы (принадлежность систем по отношению создания: выполнение операций какого-то метода над чужим предметом метода), и в данном случае — полномочия по изменению конструкции целевой системы и надсистемы, тоже отношения собственности. Даже если у нас служба (server), которая что-то красит, то мы полномочны менять слой краски, но вряд ли полномочны менять ту систему, которую красим и те системы, которые потом смотрят на краску (если это художественная окраска) или воздействуют на краску (если это защитная краска от какой-нибудь агрессивной химической среды).

И тут можно отметить, что ситуация уже не такая жёсткая как ещё десяток лет назад. Раньше не предполагалось, что команда как-то может повлиять на надсистему — она брала её в проект как данное, и должна была просто стыковать свою целевую систему с имеющимся системным окружением. Сегодня это не так: команда проекта не уполномочена изменять системное окружение непосредственно, своим волевым решением. Но команда проекта может влиять на то, чтобы в целях согласования характеристик целевой и надсистемы эта надсистема была всё-таки изменена — самостоятельно её владельцами, или командой проекта по согласованию с владельцами надсистемы. Для этого команда проекта (внутренние проектные роли) активно работает с внешними проектными ролями/стейкхолдерами, влияет на них, и это происходит долго, ибо «непрерывное всё», развитие системы продолжается долго, это не «изготовили и забыли», речь может идти о многолетней совместной эволюции целевой системы и её надсистемы, смене многих версий целевой системы и надсистемы. Всё-таки целевая система — это просто подсистема надсистемы как системы, инженерно они должны быть согласованы, и иногда легче поменять что-то в надсистеме, чем в целевой системе.

При выявлении надсистемы нужно помнить, что в зарубежных текстах иногда надсистему называют «использующей системой», using system.

Это потому, что в её составе инженер использует целевую систему. Это using относится к моменту создания, а не к моменту эксплуатации. Хотя часто так говорить можно и о моменте работы/функционирования системы, речь об окружении, которое «использует», «задействует», «эксплуатирует» систему. Но «часто» — это не всегда. Так, если оговориться, что «женщина использует мужчину», то дальше можно ошибочно по чисто лингвистическому рассуждению сказать, что женщина — это использующая система. И отождествить с надсистемой. То есть мужчина получается в составе женщины (все молекулы мужчины входят в число молекул женщины), что есть нелепость. Поэтому аккуратней со значениями слова «использование», их два:

- Использование целевой системы инженером надсистемы этой целевой системы как части конструкции надсистемы в design-time. Часы::«целевая система» используются дизайнером интерьеров в «интерьере квартиры»::надсистема (окружение часов). Ни о каком «сценарии использования» тут речи быть не может, ибо тут не время использования часов, они ещё не работают, это время проектирования — для них только проектируют среду, в которой они будут работать, их размещают в этой среде. «Использование дизайнером» это совершенно не «использование в ходе эксплуатации».
- Использование кем-то или чем-то в окружении в run-time/operations (нужно дальше разбираться, что это вообще за «использование»). Часы используются жильцом квартиры, чтобы поглядеть время (и вот там речь пойдёт о сценарии использования, use case).

При определении надсистемы важно, чтобы это был ближайший системный уровень (принцип почтальона, адрес должен быть точен — нельзя отсылать в слишком

общий объект). Уровень надсистемы должен быть тот, на котором ожидается эмерджентность/системный эффект от работы целевой системы в её составе. «Я делаю цветок, который улучшит мир в целом» — это влажные мечты, профнепригодность. Вам надо найти надсистему, в которой цветок изменит состояние, даст какое-то свойство, которое было бы другим, если бы цветка не было. Вряд ли мир в целом станет другим, но если это ситуация «свидания» (можно её рассмотреть как «мероприятие», как надсистему для цветка), то цветок может улучшить настроение участников свидания. Поэтому если вы делаете цветы, которые будут использованы для свидания — так и скажите. Если для похорон — это, возможно, будут бумажные цветы. Или букетик цветов с чётным числом цветов. Не ошибитесь с надсистемой, она важна!

Верный способ выявить надсистему — это разобраться, с какими агентами (прежде всего, конечно, людьми) приходится разговаривать (помним, что разговариваем в лице этих людей с исполняемыми этими людьми ролями!), чтобы проект состоялся. Они обычно и есть владельцы надсистемы. Скажем, вы делаете «систему перевозки самолётов», используемую во время ремонтов самолёта. Целевая система — отремонтированный самолёт, который опять летает, перевозя грузы и пассажиров (эти грузы и пассажиры будут в надсистеме отремонтированного летающего самолёта в момент эксплуатации). Ваша «система перевозки самолётов» (плохо, что в названии слово «система», но хотя бы так, чем никак), но что для неё надсистема?! Надсистема — авиация страны, министерство промышленности? Нет, принцип почтальона говорит, что это «правда, но бесполезная правда», с равным успехом можно говорить и о «человечестве» как надсистеме, и даже о «всей вселенной»! В данном случае довольно просто выясняется, что «системой перевозки самолётов» интересуются главным образом люди из эксплуатационной службы одного из авиапредприятий.

«Эксплуатационная служба авиапредприятия»::организация (и даже не всё авиапредприятие!) и является надсистемой для системы перевозки самолётов, а целевая система там будет — самолёт, который надо в том числе перевозить. Система перевозки самолётов тут — один из создателей для самолёта, инструмент для эксплуатационной службы, служащий для выполнения работ сервиса перевозки. У команды эксплуатационной службы авиапредприятия (внешние проектные роли по отношению к системе перевозки самолётов) есть потребности — им нужно ремонтировать самолёты, которые не летают. И поэтому у них есть какая-то концепция использования наземной системы перевозки самолётов (заведомо не летающих). Всё, с этого момента (система перевозки самолётов как часть эксплуатационной службы авиапредприятия = система как часть надсистемы) можно начинать обсуждать проект по созданию системы перевозки самолётов. Ход на выявление владельцев надсистемы обычно крайне продуктивен: всё равно с этими владельцами нужно общаться, они важнейшие внешние роли в вашем проекте.

Ещё одна ошибка, когда надсистема и целевая система называются одинаково (мы уже говорили об этой ошибке, но её повторяют снова и снова, поэтому повторим ещё раз, другими словами). Разбиения типа «А состоит из А и Б» недопустимы, может быть только «А состоит из Б и В» — и скорее всего эта ошибка не в том, что само разбиение в жизни какое-то неправильное, а просто неправильно выбраны имена для систем в нём. Например, «ячейка состоит из ячейки и прокладки». Подробное обсуждение показывает, что речь идёт о ячейке, которая состоит из

корпуса и прокладки. «Жилая комната состоит из комнаты и интерьера». Подробное обсуждение показывает, что жилая комната состоит из помещения (строительной части, коробки без отделки) и интерьера (в который входят даже приклеенные к стенам помещения обои).

А ещё системное мышление состоит в том, что вы должны разобраться с устройством надсистемы, чтобы понять роль в ней целевой системы. На картинке в этом подразделе, взятом из работ Jan Dietz, это обозначено как reverse engineering для using system (a ontology — это разные описания надсистемы, получаемые путём обратной инженерии, выявления устройства надсистемы). Целевую систему вы будете получать прямой инженерией (но то, что это «прямая инженерия» обычно не говорят, говорят просто «инженерия»): описания целевой системы вы будете выявлять в части использования в надсистеме (концепция использования) и разрабатывать (концепция системы), чтобы целевая система смогла выполнить ожидаемую от неё функцию в надсистеме.

Как делать все эти описания (методы разработки концепции использования и концепции системы), разбирается подробно в руководстве по системной инженерии. В нашем руководстве по системному мышлению только указывается, что **вам обязательно нужно понять роль целевой системы в надсистеме и выполняемую этой ролью функцию/метод/сервис, для этого нужно обязательно выявить надсистему и как-то разобраться в её устройстве, и зачем в этом устройстве нужна целевая система.**

Если вы не разбираетесь сначала с надсистемой, не разбираетесь с вышестоящим системным уровнем, и только после этого идёте разбираться с подсистемами, то у вас не системное мышление. **Ход проектного внимания от целевой системы на надсистему и разбирательство с ближним системным окружением сначала, а с частями целевой системы потом — это главное в системном мышлении.**

Системный подход: для всех видов систем, не только для целевой

Системное мышление обладает свойством рекурсивности в его применении: оно похожим образом разворачивается для каждой из систем в обширных и глубоких системных разбиениях. Как-то специально выделять целевую систему среди многочисленных системных уровней какого-то системного разбиения нужно главным образом из-за необходимости помнить, что является коллективной целью, то есть для работы в каких-то командных проектах. Это новинка системного подхода в его втором поколении, деятельностное (субъективное/постпозитивистское, т.е. определяемое намерением какого-то агента что-то изменить в физическом мире) введение целевой системы/system-of-interest. **Целевая система всегда выделяется из мира кем-то и для чего-то (то есть где-то используется, а не просто «существует»), это выявление целевой системы не «объективно» и «правильно/истинно».**

Тем не менее, понятия системного мышления приложимы ко всем системам, не только к целевой и даже не только к «нашей». Вокруг любой системы можно выстроить иерархию по отношению композиции и построить граф создателей по отношению создания.

Так, свои надсистемы и подсистемы (над- и под- тут всегда отношения композиции, в системных разбиениях эти приставки не используются для указания вида и рода,

вариантов) есть и у целевой системы, и у надсистемы, и у подсистем, и у любых других систем окружения и даже систем создания в их обширных графах. Различать описания (ментальный объект) и системную документацию (носитель с описанием), равно как отличать описание системы от воплощения системы нужно для всех систем. Все системы прежде всего называют по их первичному назначению/роли в окружении и выполняемой функции/методу/сервису, очень редко обращаясь к конструкции. Во всех системах подсистемы взаимодействуют, чтобы они проявили эмерджентность/системный эффект, хотя в некоторых системах это взаимодействие устроено хитро — оно происходит через системы-создатели, наблюдающие эмерджентные свойства таких систем, например, клиентура как система взаимодействует с создателем продукта — и такие её характеристики, как ёмкость рынка, относятся ко всей клиентуре, а не к отдельным клиентам.

Систем какого-то вида может быть в проекте много, необязательно одна. Например, **надсистем у одной целевой системы может быть много. Целевая система вполне может входить в состав множества надсистем, у которых может быть у каждой своя группа внешних проектных ролей, занятых своими проектами, озабоченных своими разными потребностями для каждой из многочисленных надсистем. И систем создания у одной целевой системы также может быть множество.**

Скажем, одна компания проектирует здание, другая его строит, третья эксплуатирует, и эта же третья сносит, четвёртая строит заново, «восстанавливает исторический облик» — каждая компания будет отдельной системой в общем проекте создания и развития здания. Есть эмерджентность/системный эффект, возникающий от того, что все эти компании работают в каком-то смысле совместно (и часто эта совместность оговаривается в договорах между этими компаниями), и можно сказать, что всё это не отдельные системы создания, а подсистемы одной системы создания. Но так говорить — это совсем необязательно, так обычно не делают. Мы говорим чаще всего о множестве создателей в графе создателей, а не об одном создателе, внутри которого рассматриваем граф уже из подсистем этой объемлющей единственной системы создания, да ещё там эти подсистемы не просто совместно работают, а друг друга создают.

И мы рассматриваем обычно именно графы создателей, а не просто линейные последовательности-цепочки создателей, связанных отношением создания. Один создатель может создать и развивать несколько разных систем, в том числе несколько разных создателей, а одну систему, в том числе создателя, могут создавать несколько разных её создателей, предоставляющих для этого разные свои сервисы.

Даже в биологическом мире в создании большинства организмов принимает участие не только самка, но популяция из двух организмов: самец и самка, каждый предоставляет собственный сервис/метод для создания потомства.

Напомним тут и частую ошибку менеджеров: попытка выделить типовую целевую систему, которой они занимаются в самых разных непохожих друг на друга проектах. При этом возникают серьёзные осложнения: поскольку надсистемы оказываются у этих систем совсем разные, невозможно обобщить разные потребности самых разных внешних проектных ролей самых разных проектов создания самых разных систем так, как будто это всё внешние проектные роли с их набором потребностей для одной сверхобобщённой целевой системы. Нет, проекты

для разных видов целевых систем, которые будут использоваться в разном окружении и работать там, предоставляя разные сервисы/функции/методы для этого окружения, нужно разделять, не пытаться их искусственно объединить в мышлении. Если нет эмерджентности от объединения (новые свойства в момент использования! А не в момент создания), то и не надо объединять какие-то части в искусственно создаваемую «систему».

Если менеджеру разными путями попало десять проектов, то не нужно делать вид, что это один большой проект и есть системный эффект от объединения этих десяти проектов в один, этих десяти целевых систем в одну: мы уже говорили тут много раз, что отношения владения/собственности, отношения влияния и распоряжения ресурсами — это не отношения композиции/«часть-целое».

Наличие нескольких разных системных разбиений (как минимум, наличие в рассмотрении для якобы одной целевой системы различных надсистем с их различными внешними проектными ролями) в рассмотрении должно настораживать: скорее всего, речь идёт не об одном проекте, а о нескольких — и вы просто должны понимать, что в жизни одновременно проходит множество проектов, и эти проекты как-то влияют друг на друга.

Речь не идёт о том, что вы находите системные разбиения разные по их виду: функциональное, конструктивное, размещения, по источникам финансирования и т.д. Речь идёт о том, что получается несколько вариантов разных функциональных или конструктивных разбиений, одного типа — и трудно их объединить в одно (хотя, конечно, все эти разбиения входят в разбиение «надсистемы всех надсистем», Вселенной, но нас волнует как всегда не теоретическое/формальное объединение, а объединение для удобства задействования каких-то методов работы, практическое/деятельностное объединение).

Скажем, вы занимаетесь покраской сепулек и монтажом фенечек. Не пытайтесь объединить эти проекты, определить «сепуле-фенечковую систему». Вам придётся отдельно заниматься покраской сепулек, разбираться с их надсистемами, иметь для этого каких-то специалистов-маляров (или красильщиков, специфика сепулек в этом плане важна!) в нужных для покраски целевых сепулек создателях/оргзвеньях/предприятиях, иметь в этих создателях как провайдерах покраски сепулек ещё и инструменты покраски. И отдельно надо будет заниматься фенечками и их монтажом, разбираться с их надсистемами, иметь отдельный проект с отдельными специалистами-монтажниками фенечек, отдельными инструментами монтажа. Если у этих проектов один начальник, или даже ими занимается одна и та же команда, системы в этих проектах будут разные, ничего хорошего от объединения этих проектов в мышлении (путём выдумывания сепуле-фенечковой системы) не будет.

Не смешивайте эти системы, не смешивайте эти проекты. Работайте с ними отдельно. Скорее всего в этих проектах будут совсем разные ситуации, они потребуют разных инженерных, менеджерских, творческих решений, разного моделирования, взаимодействия с разными внешними проектными ролями. Каждый раз, когда происходит обобщение или объединение проектов и систем, нужно задавать вопрос: какие вопросы станет легче решать при этом обобщении или объединении? Вполне возможно, что никакие — и лучше разворачивать системное мышление индивидуально в разных проектах, не пытаться из них сделать

«суперпроект».

Итого: чтобы найти свою целевую систему среди чужих в системном разбиении нужно задаваться вопросами не только про неё, но и ответить на множество сопутствующих вопросов. Вопросы для себя, любимого:

- Что лично *твоя* целевая система?
- Что целевая система *твоего* клиента?
- Что целевая система *твоей* команды?
- Что *ты* можешь изменить своим решением, а на что только влияешь?
- Что может изменить своим решением *твой* клиент (там разные подроли! И у них могут быть разные мнения по поводу решения!), а на что только влиять?
- Какую систему ты называешь целевой для *себя*, когда думаешь? *Твой* клиент? *Твоя* команда?

Вопросы для себя, как члена команды:

- Когда *вы* (кто входит в состав этого «вы»? где границы этого «вы», определяющие кто и что там снаружи, а кто и что внутри?) общаетесь совместно с клиентом, командой, какую систему *ты* будешь называть целевой — свою собственную, или клиента, или команды? Помним при этом поговорку театралов: «как было на репетиции, так будет и на спектакле». Лучше бы в разных разговорах вы одни и те же системы называли одинаково.
- Команда чья — *твоя*, или твоего клиента? Команда твоя внутри компании «против всех и клиента», или вся компания как команда «против клиента», и нет твоей «команды против компании»?
- Когда в проекте разные люди говорят *«мы»* — кто входит в состав этого «мы»? Кто в этом «мы» окончательно определяет (не имеет полномочия утверждать чей-то выбор, а реально сам думает и определяет!) выбор целевой системы, определяет её границы с окружением? Кто озабочен надсистемой, то есть чьи потребности удовлетворяем?

Предписанных ответов тут нет, алгоритмов мышления на эти темы нет, но системное мышление предлагает задуматься над ответами на эти вопросы и как-то определиться. Это вопросы второго поколения системного мышления: важны не только отношения часть-целое в системном разбиении, в котором находится целевая система, но и системы-создатели, о них обязательно надо думать.

При этом обязательно нужно много общаться (системное мышление — коллективное мышление!) а также действовать, работать в проекте, принимать инженерные решения, а не только «понимать» как «наблюдатель/аналитик», не «просто думать». Надо провести какие-то переговоры с самыми разными внешними и внутренними проектными ролями, с членами команды и внешними людьми, каждый из которых может иметь какой-то особенный интерес, надо провести действия, которые ведут к достижению договорённостей по этим интересам, а также надо поменять ситуацию так, чтобы система стала успешной — эти интересы удовлетворились. Системное мышление — про договорённости (и, конечно, последующее их исполнение, воплощение в жизнь). Вам нужно достичь договорённостей в проекте, а системное мышление подсказывает, по каким вопросам важно эти договорённости достичь. Первая из них — это договориться о целевой системе, определить её границы, её роль и функции/сервисы в надсистеме.

Только после рассмотрения организационной ситуации вы можете принять решение: что считать *целевой системой*, а что считать *вашей системой* (необязательно эти системы совпадут!).

Последнее дело тут представлять себя «объективным» системным мыслителем на белом коне «над схваткой». Нет, вы должны чётко отдавать себе отчёт — какая ваша собственная проектная роль, или набор проектных ролей, насколько профессионально можете сыграть вашу роль, насколько велика вероятность, что вы сделаете в этой роли новичковые ошибки, насколько плохо сработаете из-за отсутствия мастерства и/или инструментария для используемых вами методов работы в проекте (не надо говорить, что вы не используете никаких методов! Если работаете, то используете — другое дело, что не задумывались, какие это методы, не думали, как их поменять).

Системное мышление тут поможет, но оно не содержит «объективных ответов». Системное мышление поможет только в том, что позволит больше времени провести в размышлениях о важном, не даст об этом важном забыть: оно поможет в управлении вниманием в большом проекте. И это, возможно (но не гарантированно!), убережёт от ошибок в сложных ситуациях, не даст увязнуть в неважных мелочах, поможет объединить труд в условиях глубокого разделения труда по самым разным проектным ролям, раскиданным по самым разным исполнителям этих ролей.

В мышлении вы должны чуть ли не буквально следовать системной мантре. Мы не будем тут её повторять, мы надеемся, что вы её сейчас уже сами сможете воспроизвести по памяти.

Вопросы. Системы создания

Программист Петя не может работать без ВКонтакте. Если он не напишет трех постов за день, у него наступает синдром абстиненции. ВКонтакте является:

--- Целевой системой программиста Пети ### Нет, Петя, как видно из задачи, не разрабатывает софт самого ВКонтакте — у него какая-то другая целевая система, над которой он работает.

--- Системой создания Пети как программиста ### Нет, ВКонтакте не разрабатывает Петю, не готовит его как программиста.

+++ Системой в окружении Пети ### Да, Петя взаимодействует с находящимся в его окружении ВКонтакте в тот момент, когда Петя работает. ВКонтакте тут как заправочная станция для автомобиля: находится в окружении автомобиля, автомобиль с ней взаимодействует, она не часть автомобиля, но автомобиль без этого взаимодействия не работает. И тут Петя не программист, а просто Петя.

--- Надсистемой программиста Пети ### Нет, Петя не часть ВКонтакте.

Программист Петя работает над новой компьютерной игрой. Каждую неделю он пишет отчёт о ходе своих работ и подаёт заполненную

ведомость затраченного им времени (таймшит) руководителю проекта Васе.

--- Отчёт документирует целевую систему, таймшит документирует систему создания ### Нет, отчёт написан тоже о работе, а не о самой игре. Так что отчёт документирует систему создания, как и таймшит.

--- Отчёт документирует систему создания, таймшит документирует целевую систему ### Нет, таймшит — это тоже о работе по созданию игры, а не о самой игре. Так что таймшит документирует систему создания, как и отчёт.

+++ Отчёт и таймшит документируют систему создания ### Да, и то, и другое — о работе, то есть о поведении программиста Пети как системы создания и развития новой компьютерной игры как целевой системы, которую пишет Петя.

--- Ни один из этих ответов не верен ### Нет, правильный ответ в списке есть: и отчёт, и таймшит документируют систему создания.

Руководитель проекта Вася работает над компьютерной игрой. Каждую неделю он пишет отчёт о ходе работ команды и посылает заполненную ведомость затраченного командой времени (таймшит) заказчику Гене.

--- Отчёт документирует целевую систему, таймшит документирует систему создания ### Нет, отчёт написан тоже о работе по созданию игры, а не о самой создаваемой игре. То есть отчёт документирует систему создания, как и таймшит.

--- Отчёт документирует систему создания, таймшит документирует целевую систему ### Нет, таймшит — это тоже о работе по созданию игры, а не о самой игре. Так что таймшит документирует систему создания, как и отчёт.

+++ Отчёт и таймшит документируют систему создания ### Да, и то, и другое — о работе по созданию целевой системы (компьютерной игры), то есть о создателе.

--- Ни один из этих ответов не верен ### Нет, правильный ответ в списке есть: отчёт и таймшит документируют создателя (они оба про работу по созданию системы, а не эксплуатации).

Застройщик сдал жилищный комплекс «Угорские триалы» без отделки и без разведённых по квартирам инженерных коммуникаций. Покупатель квартиры Иванов нанял бригаду отделочников, которые живут в квартире и делают ремонт. Семья Ивановых по отношению к квартире является:

+++ Создатель ### Да, ибо в момент создания квартиры они организуют бригаду отделочников. Иванов-наниматель-бригады определённо причастен к доделке квартиры (это неважно, что в другой своей роли он потом там будет жить).

--- Надсистема ### Нет, квартира не часть семьи (но часть домохозяйства как системы, состоящей из семьи, квартиры, интерьера и домашней утвари).

+++ Система в окружении ### Да, ибо в момент эксплуатации квартиры семья Ивановых и квартира будут входить в одну надсистему — «домохозяйство».

Застройщик сдал жилищный комплекс «Угорские триалы» без отделки и без разведённых по квартирам инженерных коммуникаций. Покупатель квартиры Иванов нанял бригаду отделочников, которые живут в квартире и делают ремонт. Бригада по отношению к квартире является:

+++ Создателем ### Да, раз бригада-как-ремонтник создаёт квартиру, ведёт её жизненный цикл. Тут два проекта, создания и проживания. Бригада-как-ремонтник в проекте создания квартиры.

--- Надсистемой ### Нет, квартира не часть бригады.

+++ Операционным окружением ### Да, пока бригада-как-жилец живёт в квартире и ведёт в ней хозяйство — то есть пока квартира эксплуатируется, а бригада взаимодействует с квартирой в ходе эксплуатации (operations). Это для Ивановых квартира не готова и не эксплуатируется. Для бригады-как-жильца квартира вполне готова и эксплуатируется как жильё. Тут просто два разных проекта: создания и проживания. Бригада-как-жилец тут в проекте проживания в квартире.

Застройщик сдал жилищный комплекс «Угорские триалы» без отделки и без разведённых по квартирам инженерных коммуникаций. Покупатель квартиры Иванов нанял бригаду отделочников, которые живут в квартире и делают ремонт. Ребёнок семьи Ивановых по отношению к квартире является:

--- Надсистемой ### Нет, квартира не часть ребёнка.

+++ Исполнителем проектной роли ### Да, ребёнок — это вполне определённая роль со вполне определёнными интересами (например, обычно ему нужна в квартире отдельная «детская» комната).

--- Надсистемой или исполнителем проектной роли, в зависимости от возраста ребёнка ### Нет, квартира не может быть частью ребёнка, независимо от возраста.

--- Надсистемой или исполнителем проектной роли, в зависимости от отношения к детям в семье покупателя ### Нет, квартира не может быть частью ребёнка, независимо от отношения к детям в семье покупателя.

+++ Системой в окружении ### Да, в ходе эксплуатации квартиры в составе домохозяйства квартира взаимодействует с ребёнком.

--- Ни один из этих ответов не верен ### Нет, правильные ответы в списке есть — ребёнок одновременно рассматривается как проектная роль и как система в окружении квартиры.

Задания: вы — сервис-провайдер

Поставьте отметки о выполнении:

1. Написан пост, в котором вами изложена системная мантра, конкретизировав все типы мета-мета-модели в этой мантре типами из вашего рабочего проекта. Вы проверили по тексту изложения системной мантры в одном из мест нашего руководства, что ничего в этой мантре не пропустили.
2. Написан пост по итогам прохождения материала и занятий моделированием в текущем разделе руководства. В посте отражено, что вы изменили в своей работе после принятия наставлений раздела всерьёз.
3. Написан пост про себя как сервис-провайдера. В посте перечислено, какие сервисы вы предоставляете окружающему вас миру, каковы предметы этих сервисов.
4. Написан пост о том, как далеко вы находитесь от целевой системы в вашем проекте в графе создателей, насколько трудно сохранять нацеленность вашего проекта на успех целевой системы. В каждом переходе по отношению создания («создатель-1 создаёт и/или развивает создателя-2») в графе создателей от целевой системы и по отношениям часть-целое в системном разбиении есть соблазн забыть, для чего существует весь этот граф создателей и все эти подсистемы. Опрошены ваши коллеги по проекту про их понимание ситуации, их нацеленность на успех целевой системы, на рост выпуска, к материалу поста добавлено их мнение.
5. Имена систем, которыми вы заняты в вашем рабочем проекте (как минимум, те имена, которые зависят именно от вас), поправлены во всех документах. Написан пост, как вам это помогло в коммуникации с другими участниками вашего рабочего проекта.

10. Как описывать системы

Фундаментальность/трансдисциплинарность системного мышления

Только после того, как удалось выявить/discover целевую систему в её окружении (первый шаг внимания в системном мышлении всегда наружу от границы целевой системы!) как «чёрный ящик», можно заняться тем, что заглянуть внутрь этого «чёрного ящика» и посмотреть, какие там должны быть подсистемы, чтобы их взаимодействие давало функцию системы по отношению к её окружению. Только второй шаг мышления может быть внутрь границ целевой системы, первый шаг всегда — от границ целевой системы к окружению! Если ты не знаешь, что требуется от системы в целом, описание частей системы и их взаимодействия тебе

не поможет. Если ты не знаешь, время тебе надо замерять своим датчиком, или давление воздуха, или температуру воды — никакое обсуждение внутренностей датчика тебе не будет в помощь. Но нам же известна роль «датчик»? Разве этого недостаточно? Нет, это слишком общий тип, в предыдущем разделе мы уже разбирали, что слишком общие имена не помогают, нужно разбираться, что именно надо замерять, с какой точностью, сколько могло бы стоить измерение и сколько времени оно должно занимать, в каком формате и куда передавать данные измерений. Это всё про датчик как чёрный ящик. И только после этих уточнений можно обсудить, из каких функциональных частей надо делать этот датчик, чтобы он выдавал потребную функцию, а потом искать аффордансы в конструктивных объектах, чтобы воплотить задумку функционального устройства системы.

Нам потребуются следующие понятия:

- **разбиение/breakdown системы** (и их основных будут четыре, в следующем же подразделе мы их опишем). Важно, что разбиение как иерархия отношений часть-целое/композиции принадлежит физическому миру, оно реально существует. Если у вас есть шестерёнки механических часов, висящих на стенке в вашей комнате, то эти шестерёнки относятся к часам как части к целому. И это отношения между реальными объектами, это не описание отношений! Разбиение/breakdown мы отличаем от описания разбиения — это информация (модель) о частях и целых. В жизни, конечно, очень часто под «разбиением» имеют ввиду не собственно разбиение, а описание разбиения, а ещё чаще — документ с описанием разбиения. Но одно дело часы, разбивающиеся на корпус, шестерёнки, циферблат, стрелки, пружинки, а другое дело — компьютерный файл, содержащий описание этого разбиения. Разбиение — в самих часах, а в файле — описание/модель/информация разбиения.
- **описание системы** — в простейшем случае это будет описание разбиения, но описания могут быть самые разные. Например, функциональное (ролевое) разбиение даёт функциональные части системы (подсистемы). Используя знание об этих частях, мы можем описать какие-нибудь потоки (энергии, вещества, информации) между частями на одном или даже нескольких системных уровнях, получив «принципиальную схему». Или можем описать какие-то физические характеристики системы в целом, зависящие от времени и набора, и состояния подсистем (так называемое 1D-моделирование). Главное тут запомнить, что кроме описаний разбиений есть ещё множество других описаний.

И вот тут мы столкнёмся второй раз с уже упоминавшейся фундаментальностью/трансдисциплинарностью системного подхода: в связанных с системой проектах присутствует множество самых разных проектных ролей, которые работают с самыми разными предметами методов разными методами, и каждая из них, ведомая своими ролевыми интересами/предпочтениями и изобретательно их реализующая, предложит свой вариант описания системы, выигрышный для реализации предпочтений именно этой роли в ущерб реализации предпочтений других ролей.

В том числе **роли будут предлагать своё видение разбиения воплощения системы на части. Система будет одна и та же (о ней договорятся, когда будут делать концепцию использования), деление на части от разных ролей — разное. А потом эти части (даже если договорятся о том, какие**

они) можно будет ещё и описывать самыми разными способами. И, конечно, в ходе обсуждения концепции системы — договариваться о том, чтобы все эти разные описания системы (включая разные описания разбиений) соответствовали одной системе, а не разным.

Кто-то::роль интересуется **функциями частей** в ходе эксплуатации::время и скажет, что заварочный чайник состоит из функциональных частей ёмкости, литьевого носика, заливочного отверстия в ёмкости, ручки на ёмкости, крышки и ручки на крышке, а ещё паровыпускного отверстия в крышке (иначе при закрывании мокрой крышки будет из носика выплёскиваться вода), это всё «для чего нужно». Другой::роль говорит про то, что заварочный чайник состоит всего из двух **деталей** (то есть не подлежащих далее разборке частей) в **конструкции** чайника, которые нужно изготовить, ибо его предметы интереса/важные характеристики лежат во времени изготовления чайника, предметом интереса тут конструкция, «из чего изготовить». Третий::роль говорит, что крышка и чайник должны храниться рядом, лучше бы крышка прямо на чайнике, предметы интереса в **размещении**, «где во Вселенной находятся части работающего чайника». Четвёртый говорит, что в чайнике главное — это его **стоимость**, поэтому лучше бы он был не из золота и не слишком причудливой формы, чтобы формовать его было подешевле, а потом он должен не ломаться и не терять тепло, чтобы не было затрат на ремонт и эксплуатация чайника была подешевле (и тут часть описаний относится ко времени эксплуатации, а часть — ко времени изготовления).

Фундаментальность/трансдисциплинарность тут в том, что системный подход заранее соглашается со множественностью описаний систем на каждом системном уровне и готов работать с этими описаниями, приходящими из разных прикладных/частных/предметных методов (материаловедение, культура чаепития, производство посуды и т.д.) с их разными знаниями/дисциплинами/теориями/алгоритмами и инструментами поддержки этих знаний. Системное мышление «по ту сторону дисциплин»/«в основе мышления о дисциплинах» всех этих методов работы всех этих самых разных проектных ролей, которые задействуют самые разные прикладные методы:

- Системное мышление учитывает **разные виды описаний, которые мы делаем для систем на каждом системном уровне**, потому как из-за эмерджентности нужно на каждом новом системном уровне описывать новые свойства. Мы уже говорили, что описание работы клеток в мышцах танцора абсолютно не такое, как описание его перформанса, описание работы аккумуляторной батареи в электромобиле абсолютно не такое, как описание движения этого электромобиля из Москвы во Владивосток, описание методов мышления людей абсолютно не такое, как описание методов работы нейронов мозга людей. Системное мышление как использование фундаментальных/предметонезависимых/трансдисциплинарных понятий системного подхода учитывает все эти прикладные описания, оно общее для рассуждений о них, оно трансдисциплинарно по отношению к описаниям отдельных прикладных методов. Его используем всегда, независимо от прикладной предметной области.
- Системное мышление учитывает **разные виды описаний, которые мы делаем на одном системном уровне**, потому как разные проектные роли имеют разные интересы к системам на этом уровне и для их удовлетворения им требуется построить какие-то прикладные модели этих систем,

помогающие им прогнозировать интересующее их поведение системы. Одним проектным ролям хочется знать, как именно система работает внутри себя, чтобы выполнять роль/функцию в надсистеме, а другим проектным ролям хочется знать, из чего система сделана/сконструирована, третьим интересно, где во Вселенной все эти части системы расположены, четвёртым интересны финансовые вопросы по поводу системы, пятым интересны работы по созданию и развитию системы (то есть интересно поведение создателей системы), и так далее — по всем предметам интереса всех ролей.

Системные описания выполняются самыми разными методами и помогают самым разным прикладным ролям договориться по системам, методам работ по их созданию, по самим работам по этим методам и ресурсам для этих работ, далее по эксплуатации, далее продолжать договариваться по вопросам развития системы.

Системные описания не только выполняются разными методами этих описаний, но и поддерживают работу по самым разным прикладным методам за счёт того, что системное мышление как метод мышления со своим набором понятий находится «по ту сторону» (транс-) от этого множества описаний, системное мышление никуда не исчезает, когда обсуждаем вопросы других дисциплин, оно пронизывает все рассуждения в проекте — и тем самым позволяет увязать все эти системные описания в непротиворечивое целое. «Увязать все эти описания» — это означает, что разные прикладные роли (прикладных видов инженерии, менеджмента), объединяемые системным мышлением, **договариваются** о том, какую надо делать систему, а также о том, насколько успешной она получается, обоснованы ли принимаемые решения.

В условиях современного разделения труда в сложных проектах визионеры принимают свои рациональные решения, разработчики — свои решения, архитекторы и «безопасники» — свои, инженеры внутренней производственной платформы — свои, менеджеры тоже принимают примерно такой же набор решений с примерно таким же разделением на роли, только по отношению к организациям в графе создателей. Системное мышление — это часть интеллект-стека фундаментальных методов мышления, так что оно не «выключается», когда идут прикладные рассуждения всех этих ролей. Системное мышление помогает удерживать внимание всех ролей на важных объектах проекта, по поводу которых надо будет договариваться. Системное мышление выпячивает противоречия в описаниях и ставит вопрос о достижении договорённостей по этим коллизиям перед всеми участвующими в проекте ролями. Описания системного мышления не заменяют прикладных описаний, они определяют типы для прикладных описаний самых разных других дисциплин, используемых в проекте. Каждая роль не может сказать, какие ещё надо делать описания, а затем что с этими описаниями делать в части решения противоречий. Каждая роль занимается только своими описаниями, а системное мышление рассказывает о том, что описаний заведомо много, подсказывает, какие из них самые важные и их нельзя не делать, а также заставляет всех договариваться по поводу этих описаний.

Четыре основных вида описания системы на базе четырёх способов разбиения системы

Системное мышление предусматривает множество описаний системы, и в этом оно ничем не отличается от общефилософской/онтологической идеи о множественности описаний какой-то ситуации разными агентами, преследующими

разные цели. Но **системное мышление конкретизирует, какие именно описания надо обязательно сделать**, какие описания будут основными — хотя системное мышление признаёт, что разных видов описаний надо делать больше, чем основных.

На сегодня системное мышление выделяет **четыре основных вида описаний, которые отражают четыре основных способа деления системы на части. Для каждого получаемого разными способами разбиения на части набора частей системы делают разные описания/view, современное системное мышление делает обязательными четыре таких вида описаний и в тренде пятое. Для каждого вида частей делается, конечно, множество разных описаний**:

1. **Функциональные:** на основе частей, полученных функциональной декомпозицией (разбиением на функциональные части/роли по отношениям часть-целое, именно такое функциональное разбиение/breakdown чаще всего называют системным разбиением, тут особое использование слова «системный», указывающее на важность выполнения функции системы в её влиянии на объекты окружения, для интеллектуальных агентов-создателей — это влияние работ метода/сервиса на внешние по отношению к создателю предметы метода). Функциональное описание делается для времени работы/эксплуатации/функционирования/операций системы (описание «как работает», времени использования).
2. **Конструктивные,** иногда говорят структурные, не менее часто — продуктовые, чуть реже — продуктные, изредка — материальные. Эти описания делаются для частей, выделяемых как конструктивные/материальные объекты путём структурной/модульной/продуктовой декомпозиции (часто также говорят не о разбиении/breakdown/декомпозиции/decomposition, а об обратном разбиению действии — синтезе/сборке из частей, устойчивое словосочетание тут — **модульный синтез**, ибо конструктивную часть часто называют модулем). Конструктивные описания делаются для времени создания системы, это описание «из чего и как собрано».
3. **Пространственные/размещения:** как мест в пространстве (обычно именно в пространстве, но мы бы рекомендовали говорить сразу о пространстве-времени), где размещаются/locate/allocate части системы (где во Вселенной находятся пространственные части системы, полученные пространственным разбиением). Иногда такие описания называют «компоновкой». Эти описания позволяют однозначно договориться о том, какой объект обсуждается: в 4D экстенсионализме важно, что как бы ни были разны описания, если они относятся к одному и тому же месту/размещению в пространстве-времени, то это описания одного и того же объекта.
4. **Стоимостные/затратные** (на что потребуется потратить деньги и иные ресурсы, cost). Тут сразу идёт речь о total cost of ownership/совокупной/полной стоимости владения[372]. Это проявление уже второго поколения системного мышления, которое занялось не только целевой системой в её окружении, как это было в первом поколении, но и создателями, организованными в граф по отношениям создания и развития друг друга и в конечном итоге создания и развития целевой системы.

[372] https://en.wikipedia.org/wiki/Total_cost_of_ownership

Стоимостные описания затрагивают как целевую систему в момент эксплуатации (сколько стоит эксплуатировать систему), так и затраты на создателей во время создания целевой системы (сколько стоит её изготовить: или купить у создателей целиком, или закупить сырьё и заплатить создателям за работы по созданию).

5. **Работы:** тут только тренд, ещё нет отражённого в стандартах и моделерах консенсуса, но это довольно мощный тренд. Окончательно смотрим на граф создателей и работы этих создателей — и пишем описание этих работ, это обычно называют work breakdown structure, структура разбиения работ. Обратите внимание, что это именно работы (как они понимаются даже не в процессном управлении, а в проектном управлении).

Это только основные описания, в проектах подобных описаний много больше. Например, описание работ в ряду основных описаний пока только «кандидат», а описания методов работ в основных описаниях пока нет, но мы вполне можем ожидать их появления — и в руководстве по методологии мы приводим стандарты описания методов работ, прежде всего OMG Essence. Конечно, если у вас есть какие-то работы проекта, то перед тем, как описывать работы, вы должны понимать, какими методами эти работы ведутся — начиная с общего описания метода создания и развития системы. Руководство по методологии как раз наставляет по этому вопросу.

Основные, равно как и все остальные виды описаний системы в жизни в чистом виде практически никогда не встречаются, чаще всего они гибридны, и это понятно: все эти описания должны как-то биться друг с другом, быть согласованными между собой, не противоречить друг другу — и для этого гибридные описания как раз хорошо приспособлены.

Четыре основных (по факту — обязательных/нормативных/предписанных) вида описания системы за последние тридцать лет в системном подходе стали основными в большинстве его вариантов. Четвёртое, стоимостное разбиение в числе обязательных появилось только недавно, это понимание пришло в инженерных проектах последнего десятка лет. Пятое (работы) только обсуждается, а методы работы создателей описываются повсеместно, но вот в стандартах пока представлены недостаточно, чтобы их считать обязательными.

Все эти разные виды основных описаний опираются на описание системы в терминах частей, полученных разными видами разбиений/декомпозиции. Ошибка — считать, что системные описания это и есть описания разбиений, то есть описание деревьев разных (функциональных, конструктивных, мест, стоимостных) объектов, выстроенных по отношениям «часть-целое».

Описания разбиений (деревья) — это не единственные системные описания! Системные описания тут ровно в той мере, в которой обсуждаются аспекты систем в их разбиении (то, что описывается самыми разными описаниями, берётся из соответствующих разбиений). Например, можно говорить о функциональной декомпозиции системы, но можно обсуждать и функциональную модель как результат 1D-моделирования (в одноразмерных моделях что-то «течёт», поэтому функциональные параметры определяются вдоль этих условных линий потока — электрические, гидравлические цепи, теплоэнергопереносы по трубопроводам, потоки грузов в логистических/поставочных/supply цепях и т.д.). 1D-моделеры часто называют «физическими моделерами», они как раз могут считать токи и

напряжения в электрических цепях, напоры и скорости потока в гидравлических системах. Но они, конечно, сложнее представляют системы, чем просто «разбиения».

Разберёмся подробней с breakdowns/разбиениями. Сами эти разбиения, конечно, в реальном мире — все их объекты в реальности, но мы можем говорить ещё и об описаниях разбиений. В быту часто вот эта разница между разбиением «в жизни» как реальными физическими объектами и описанием разбиения как информацией, методом как поведением агента с мастерством и инструментами «в жизни» и описанием метода как информацией о поведении (теорией/дисциплиной/знаниями/объяснениями, в том числе описаниями инструментов), онтологией (какие объекты в жизни, сами объекты) и онтологическими описаниями (как мы описываем объекты в жизни, информация об объектах) утрачивается, и часто говорят об описаниях, используя основной термин «про реальность»: разбиение, метод, онтология там, где надо бы говорить описание разбиения, описание метода, описание онтологии. Говорят «разбиение», без слова «описание», но референт/денотат — не в физическом мире, а какой-нибудь документ с описанием разбиения. Как понять, что имеется в виду — реальный мир, или описание того, что в реальном мире? Только из контекста, а когда уж совсем непонятно, то правильно будет — переспросить.

Вот модификация диаграммы из стандарта IEC 81346-1:2022, иллюстрирующая первые три основных вида описаний, базирующихся на первых трёх основных видах разбиений (на ней ещё нет стоимостного описания, можете представить его как какой-то финансовый расчёт, например, «совокупной стоимости владения», total cost of ownership[373]):

[373] https://ru.wikipedia.org/wiki/Совокупная_стоимость_владения

На рисунке изображены три проектные роли, которых интересуют разные описания системы, они изображены фигурками человечков. Помним, что роли — это только роли: их может играть и один человек, и целые группы людей с их инструментами и оборудованием. На рисунке три разных человека играют три разные роли (но рисунок отражает только три аспекта как пример, даже в IEC 81346-1:2022 этих аспектов пять, ещё «тип» как произвольная группировка по какому-то свойству и «другое», куда входит всё остальное — логистика, стоимостное деление, административное подчинение. Наше стоимостное разбиение по этому стандарту будет в этом «другом» разбиении, other aspects).

Каждая из проектных ролей видит систему разбитой на определённые виды/типы взаимодействующих частей, условно изображённые цветами. **Минимально в системном подходе нужно выделить вниманием четыре основных вида/типа частей системы, и на базе этих частей уже строить системные описания**:

1. **Функциональные/ролевые части** (functional elements, но мы переводим elements как «части», ибо элементы не подразумевают дальнейшей делимости, а мы всегда о ней помним; role elements[374]), обсуждение взаимодействия которых позволяет отвечать на вопрос о том, как работает система, чтобы она выполняла свою функцию/роль в её надсистеме. Это

[374] При этом компонентой иногда называют функциональную часть, а иногда конструктивную (например, отдельно закупаемый продукт согласно IEC 81346-1:2022). Будьте внимательны: в разных предметных областях это слово употребляется по-разному. А ещё в современном русском языке компонента/компонент примерно одинаково по частоте употребления как в женском роде, так и в мужском, никакой нормы на этот счёт нет.

обязательно будет обсуждение времени эксплуатации/работы/операций/использования/службы. На картинке нарисована принципиальная электрическая схема, на которой приведено описание функциональных/ролевых частей системы и их **соединений** (connectors). Эти функциональные/ролевые части выполняют свою функцию в системе через соединения с другими функциональными частями, меняя своё состояние в ходе работы/функционирования целой системы от этого взаимодействия и меняя состояния других частей.

2. **Конструктивы/продукты/модули/«материальные части»** (modules, products, components как заказные единицы, сборочные единицы, логистические единицы, конструкты) — этот вид частей позволяет отвечать на вопрос, из чего собрать и как соединить через **интерфейсы** (interface) конструктивы/модули системы, отвечают на вопрос «как собрать систему», обсуждается время создания. Время эксплуатации, «как работать» по этим разбиениям и созданным на их основе другим описаниям обсуждать нельзя!
3. **«Пространственные объекты», места** (locations), или **размещения** (allocations), которые позволяют отвечать на вопрос, где в пространстве на заданный момент времени можно найти части системы, как система скомпонована из своих частей в пространстве. Чаще всего термин «место» даётся для run-time, а термин «размещение» даётся для пространства, определяемого в design-time. Место/размещение тем самым важно для совмещения ролевых и физических объектов в run-time: если оба находятся в одном и том же месте в одно и то же время, то это один и тот же объект, «конструктив реализует/воплощает роль». На картинке для этого разбиения изображены отсеки, в которых ведётся монтаж системы и в которых она затем работает, но из рассмотрений только мест/размещений непонятно — из чего система собрана и как она работает.
4. **Стоимостные/затратные части** (costs), позволяющие обсуждать издержки/затраты на систему, причём не только на время разработки, но и на время всего создания и развития системы: на материалы, на работы по созданию, на работы и материалы времени эксплуатации, ремонта, модернизации, на работы по уничтожению выводимой из эксплуатации системы, причём с учётом развития системы — речь идёт о многих версиях.

Напомним, про пятое разбиение —работ (work breakdown structure, WBS), но это ещё не полностью закрепилось в инженерной культуре, это только «кандидат в обязательные разбиения». Для каждой целевой системы есть создатель, который ведёт с ней работы (в ходе непрерывного ввода в эксплуатацию разных версий развиваемой системы замысливает, формулирует концепцию использования, проектирует с принятием решений по концепции системы и архитектурных решений, изготавливает с учётом решений по «методам изготовления»/«технологии производства», обосновывает успешность, а затем монтирует в состав системы очередную часть с расширенными возможностями, потом вводит в эксплуатацию и помогает эксплуатировать). Эти все работы нужно как-то учитывать, их тоже нужно как-то проектировать, распределять между подрядчиками. И это делается в каждом проекте. **Поэтому разбиение работ по созданию и развитию системы сегодня часто относят к обязательным системным описаниям, но мы пока не стали их включать. Тут даже неважен тот факт, что описание работ относится к описанию не системы, а создателя системы — стоимостное описание тоже в существенной мере**

затрагивает и стоимость сервисов создателя, так что это просто продолжение тренда на полноценную реализацию идей системного мышления второго поколения, «система не только мыслится всегда как часть надсистемы, но всегда помнится о том, что у неё есть создатели». А ещё помним, что вслед за описанием работ и разбиением работ придётся говорить и об описаниях методов работ и разбиении методов.

Часто считают, что разбиение работ по созданию и развитию системы (а также разбиение методов работы) как-то можно сделать up-front, имея в виду проектное управление и WBS оттуда, но это не так — подробности в руководствах по методологии, системной инженерии, системному менеджменту. В любом случае, разбиение работ — это уже описание не столько целевой системы, сколько системы создания (работает система создания, эту работу и описываем), хотя каждая работа имеет чёткую привязку к методу работы и предмету этого метода: какой-то важной части целевой системы, которую надо привести в нужное состояние. А ещё ведь работы по созданию инструментария, по созданию создателей — их надо бы тоже учесть по всему графу создателей! Но подождём, пока это пятое описание не станет окончательно общепринятым. А сейчас это только тренд.

Даже стоимостные части, которые появились относительно недавно как полноценный объект описания системы, уже не совсем про целевую систему (ибо затрагивают как стоимость материала/сырья, так и стоимость работы создателя, а также расходы на поддержание работоспособности целевой системы во время эксплуатации — и надо ещё договориться, как это всё рассчитывать).

Всё это выделенные в соответствии с разными ролевыми интересами части одной и той же целой системы, изображённой кубиком с цветными гранями — и каждая роль получает возможность обсуждать свой предмет интереса/важную характеристику по удобно устроенному для этого описанию:

- Интересующийся «как работает» получает обычно принципиальную/функциональную/потоковую схему с соединениями функциональных/ролевых частей (хотя сейчас вместо диаграммной нотации всё чаще предпочитают использовать иные представления, например, описание физической системы в виде системы дифференциальных уравнений на каком-то языке моделирования[375]),
- интересующийся «из каких продуктов собрать» получает описание конструкции: материальных продуктов/модулей и их интерфейсов друг ко другу,
- интересующийся «где найти, куда положить/доставить/установить» получает описание мест, где расположена система, компоновку,
- не изображённый на картинке интересующийся «сколько это стоит» получает описание того, во сколько обойдётся работа с системой. В последние несколько лет принято считать описания совокупной стоимости владения необходимыми, хотя в стандарт эти описания ещё не попали.
- И ещё есть тоже не изображённый на картинке интересующийся WBS/« какие с этим проводятся работы». Про методы работы мы и не говорим, но это по факту тоже описывают.
- … и таких интересующихся разными аспектами/вариантами разбиений на части много, это изображён был только самый минимум описаний. И описания

[375] https://ailev.livejournal.com/1549559.html

ещё и гибридны!

Системные уровни определяются по главному в системном мышлении разбиению — функциональному! Поэтому функциональное разбиение часто называют системным разбиением/system breakdown structure. А остальные разбиения? Они «просто» разбиения/breakdowns.

В принципе, эта терминология «системного разбиения» (как только функционального) не всегда соблюдается. Сама система чаще всего одновременно и функциональная часть надсистемы, и конструктивная часть этой же надсистемы (функциональные части в девяти случаях из десяти реализуются/воплощаются конструктивной частью, проблемы будут с одним случаем из десяти, мы дальше с этим разберёмся подробней), и занимает в этой надсистеме какое-то компоновочное место, и требует каких-то на неё затрат и выполнения над собой каких-то работ. В какой-то мере любое из основных разбиений системно. Но в инженерии обычно если и говорят о системном разбиении и системных уровнях, то это именно функциональное разбиение, а не конструктивное, пространственное или стоимостное.

Мы сознательно в предыдущем абзаце говорим о том, что система — часть надсистемы, а не разбиваем систему на части-подсистемы. **Основной ход системного мышления — от системы как части к надсистеме как целому.** Но рассуждение такое можно провести на любом системном уровне (рекурсивно), в том числе и для подсистем, и под-подсистем, и надсистем, и над-надсистем.

Когда говорим о частях системы, не лишне будет уточнить, это части по какому из разбиений: из основных (функциональное, конструктивное, размещения, стоимостное) или каких-то других (работ по созданию, очередности изготовления, последовательности сборки, экологичности и т.д. — их множество самых разных!). Если в описании несколько видов частей (гибридное описание, они встречаются чаще всего), то надо, чтобы вы в этих гибридах понимали, какие разбиения там описаны, а каких разбиений не хватает.

Функциональный анализ и модульный синтез

Каждая трудовая/профессиональная/проектная роль в силу специфики своих предметов интереса видит частное описание системы как «прозрачного ящика», состоящее из интересующих её видов частей, позволяющих обсудить её предметы интереса. «Операционный менеджер»::роль, например, по поводу системы интересуется не тем, как система работает (функциональным описанием), но «из чего собрано» (конструктивы) и логистикой этих конструктивных частей к месту сборки, а после сборки — к месту продажи. У операционных менеджеров нет предмета интереса «функциональность системы», но у них интерес к конструкции системы, они будут управлять конфигурацией, «чтобы ничего не потерялось, всё собралось и ничего не перепуталось». Менеджмент/management — это обычно не про функциональность, это про логистику! Это про «в срок, вовремя, ничего не потеряно, всё передано из рук в руки, выполнены все предложенные инженерами работы». Предметом операционного менеджмента как метода работы операционного менеджера будут конструктивные части, претерпевающие изменения их состояний в ходе работ создателей.

А вот инженер-разработчик, думающий над расчётом режима

работы/функционирования/операций какой-то системы, думает сначала над «как работает» (**функциональный анализ** на основе функциональной декомпозиции) система, чтобы из взаимодействия функциональных частей-подсистем появлялась/emerge нужная функция целой системы, а потом он обязательно будет решать «из чего собрать» (**модульный синтез/сборка,** подбор аффордансов-конструктивов из конструктивного разбиения для функциональных частей системы). Модульный синтез у инженера-разработчика будет не сразу, но после того, когда разработчик в ходе функционального анализа поймёт, какие именно там операционные/рабочие режимы времени эксплуатации, какие характеристики потребуются от подсистем, чтобы подобрать необходимые аффордансы конструктивов/модулей.

«Функциональный синтез» говорить было не принято. Говорили именно про функциональную декомпозицию и функциональный анализ (анализ — это тоже «дробление»). В анализе берём функциональный объект (роль) в надсистеме, и ищем его декомпозицию/разбиение на подсистемы, тоже функциональные объекты. А для конструктива как материального объекта — берём самые разные материальные объекты-конструктивы (материальность тут подчёркивает существование и вне времени эксплуатации системы, но и во время создания, и даже вне проекта — «самих по себе») и ищем в них аффордансы/подходяшки, чтобы собрать из этих аффордансов-конструктивов такой надконструктив, который сможет «сыграть требуемую роль»/«воплотить функциональный объект» во время использования. То есть вроде как конструктивы синтезируем/собираем из материальных частей, а функциональные объекты, наоборот, только дробим — это логическое описание метода.

Но в последнее время начинаем говорить и о функциональном синтезе, и о декомпозиции модулей. Стремительно происходит разделение труда: то, что раньше было просто «инженер» и затем «опытный инженер» сегодня разделяется на прикладного методолога (занимается функциями/методами), архитектора (который занимается ограничениями на модульный синтез: пытается найти оптимум для межмодульных зависимостей) и проектировщика (который пытается для предложенных методологом функций найти конструктивы, учитывая зависимости). Подробней о том, чем занимаются прикладные методологи (помним, что эта роль только-только начала выделяться, чаще всего её склеивают с ролью «опытного разработчика») — в руководстве по методологии. Как устроено общее разделение труда в инженерной разработке и как взаимодействует проектировщик и архитектор — об этом в руководстве по системной инженерии.

В жизни всё это происходит итеративно и с разными догадками, довольно долгое время, ибо оптимизация же, итеративный процесс, да ещё и на многих проектах, многих версиях системы. Особо заметим, что и функциональный и конструктивный объекты — это виды физических объектов. Разве что функциональный объект физический тогда, когда его роль играет конструктивный материальный объект (то есть материал для физичности функционального объекта берётся не «ниоткуда», а именно из материального объекта, который поставлен играть роль, выполнять назначенную функцию).

Обратите внимание, что в инженерии обязательно присутствуют

- и функциональный/системный анализ (системы обычно описывают прежде всего по их ролям, функциям — поэтому функциональный и системный

анализ синонимичны), плюс методологическая работа по тому, чтобы разобраться, какие методы сегодня SoTA в том или ином классе систем, это стык работы методолога и проектировщика,
- и модульный синтез, который идёт на стыке работы проектровщика и архитектора, но конструкция тоже, конечно, анализируется.

У нас руководство по полному системному мышлению, в котором анализу и синтезу уделяется равное внимание. Это не руководство только по системному анализу, весьма небольшой и ограниченной части системного мышления!

Те, кто занимается только анализом (функциональным/системным или каким-то другим) — это мечтатели, которые не имеют шанса изменить мир. Мир-то нужно менять физически. Для этого нужно, чтобы какие-то конструктивы/продукты/модули начали воплощать функциональные части системы, то есть физически были собраны так, чтобы начать взаимодействовать. Мало понимать, как система работает, думать о системе, обсуждать систему, анализировать её — система должна быть воплощена в физическом мире, для этого надо выполнить модульный синтез, то есть сконструировать/спроектировать конструкцию системы из конструктивов, изготовить конструктивы, собрать из них систему, отладить её работу, дальше развивать эту систему, выпускать множество постоянно улучшаемых версий системы. Это всё работа «синтетиков», а не «аналитиков». Инженер — это тот кто думает и делает, а не только аналитик, который только думает и с работами по изменению мира никак не связан.

Это ровно то же рассуждение, что про исполнителей человеческих ролей в спектаклях. Если только писать пьесы, прописывать хорошо роли и их поведение, но не думать про воплощение этих пьес актёрами (в том числе роботами, в том числе животными!), то можно смело считать, что это просто художественная литература, а не драматургия. Актёры там будут летать, растворяться в воздухе, реплики давать телепатически: если не доходить мыслью до реальности, то можно всё! Драматург и режиссёр пьесы обычно представляют, что их пьеса будет жить в реальности, играться на весьма ограниченной в пространстве сцене весьма ограниченное время, а зрителей в зрительном зале будет тоже ограниченное количество — и зрителям никак не передашь мысли и переживания персонажей, как это можно сделать в художественной литературе, просто дав описания. Нет, надо всё выразить средствами актёрской игры, на это и драматургия, написание сценариев пьес, а не просто «писательство» как таковое.

Инженерное мышление не должно быть только аналитично, умозрительно, оно должно быть адекватно, то есть не терять связи с физическим миром, не витать только в облаках, выходить на изменение мира к лучшему, к созданию и развитию систем, мало только объяснять мир, анализировать системы!

Поэтому системный мыслитель не ограничивается только временем функционирования, но занят ещё и временем разработки. Он обязательно выделяет не менее четырёх областей/зон интереса (и это только начало, обычно этих зон/областей интереса больше, и в каждой по множеству предметов интереса):

- функциональную/ролевую,
- конструктивную/материальную/модульную,
- пространственную (совмещения функциональной и конструктивной в рамках 4D экстенсионализма, отношение реализации проявляется в том, что

функциональные и конструктивные части занимают одно и то же пространство-время, поэтому пространство-время вводится явно),
- стоимостную/затратную/экономическую/ресурсную.

Анализ/декомпозиция всегда только часть мышления! Опасайтесь проекта, где непонятно кто принимает решения по синтезу! Синтез меняет мир, анализ не меняет мира! Только решения по синтезу меняют мир!

Кто имеет право указать, что в ракете будет три ступени? Что в предприятии будет три подразделения? Что Вася Пупкин будет играть роль Принца Гамлета, а Даша Неврубкина не будет? Это точно не аналитики! Это инженеры, которые меняют мир, а не только понимают мир!

«Аналитик» тут — роль, а не должность. Агент (человек или даже целая организация) на должности «аналитик» вполне может выполнять и роль аналитика, и роль синтетика/конструктора, то есть принимать решения и по функциональной декомпозиции, функциональном анализу, и по модульному синтезу. Но лучше бы назвать такую должность с принимающим решения по поводу изменения мира человеком иначе, «инженер»: участники проекта не ожидают от «аналитиков» полномочий по изменению мира и поэтому не склонны обсуждать с людьми на должности аналитиков решения по модульному синтезу.

Обычно такие решения по назначению конструктивных частей системы на ролевые/функциональные части делает инженер-проектировщик, опираясь при этом на рекомендации по методам, которые даёт методолог и принципы разбиения системы на конструктивные части и описание связи этих конструктивных частей, которые готовит инженер-архитектор (это основная часть работ по созданию концепции системы — назначение конструктивов-аффордансов на роли подсистем). Аналитик тем самым — только часть инженера-методолога, часть инженера-проектировщика и часть инженера-архитектора. Отдельный аналитик сегодня считается больше «испорченным телефоном» между внешним миром и инженером, который принимает решение. Как вы яхту назовёте, так она и поплывёт. Переобзовите аналитика инженером, и вы увидите, как он начнёт брать на себя ответственность за принимаемые решения (а не только за аналитическое «понимание» и «найденные проблемы»), а окружающие его люди будут признавать эти решения, а не считать, что это только «рекомендации аналитика» для какого-то инженера. Подробней об этом рассказывается в руководствах по методологии, системной инженерии, системному менеджменту.

«Инженер предприятия»/менеджер::роль для «инженерии предприятия»/менеджмента::метод. Орг-методологи, орг-проектировщики и орг-архитекторы/«архитекторы предприятия» понимают, что они по сути своей — инженеры организаций::система (иногда так и говорят: организационных систем). Иногда орг-проектировщики и орг-архитекторы (орг-методологи пока ещё довольно редко встречаются, эта роль полностью ещё не выделилась из роли орг-проектировщика) прямо называют себя инженерами предприятия, чтобы отличить себя от «просто менеджеров», имея ввиду, что они не «операционные менеджеры» и занимаются не методами operations time предприятия, то есть что они не «операторы предприятия» и не занимаются администрированием, но создают и развивают предприятие. Помним, что MBA — это степень master of business administration, менеджмент в разговоре сводят к администрированию. Орг-

разработчики и орг-проектировщики подчёркивают при этом, что они в инженерии предприятия пользуются стандартами системной инженерии, относясь к организации как системе.

Лидеры (те, кто проводят организационные изменения, аналоги инженеров-технологов в машиностроении, то есть «специалисты по изменениям физического мира, изготовлению воплощения системы») берут у орг-проектировщиков и орг-архитекторов концепцию организации (специализация концепции системы для орг-системы: решения по реализации/воплощению оргролей оргзвеньями с учётом орг-архитектурных решений) и дальше «изготавливают» само предприятие по этим описаниям, разъясняя оргроли из функционального описания организации агентам на каждой должностной позиции в оргструктуре как конструктивном разбиении организации. Лидеры «катализируют сотрудничество» путём создания атмосферы, где каждый агент-сотрудник качественно исполняет свои роли, где бы ни находилось его рабочее место (описание размещений) и с кем бы ни приходилось этим агентам-сотрудникам общаться в условиях неизбежности продуктивных конфликтов между интересами играемых агентами-сотрудниками ролей.

Конечно, это всё подробно рассказывается в наших остальных руководствах по программе рабочего развития, тут же нам надо только подчеркнуть, что **концепцию организации лидеры берут у орг-проектировщиков и орг-архитекторов, то есть у «синтетиков», а не аналитиков, а если концепции организации нет, то лидерам буквально нечего делать — если нет оргпроекта, непонятно, что объяснять сотрудникам в части разделения труда в проекте**. Орг-аналитика выполняется по необходимости самими людьми, выполняющими инженерные роли орг-проектировщиков и орг-архитекторов. Какой-нибудь «начальник отдела», организующий свой отдел, выполняет роли орг-проектировщика и лидера (вместе эти роли можно объединить в роли «менеджера оргразвития»), а также орг-архитектора этого отдела, и ещё может выполнять роли в проекте прикладной инженерии «его системы» (для него это будет «наша система», иногда даже совпадающая с целевой), например, архитектора системы — чем меньше предприятие, тем больше ролей играет один и тот же агент.

И, конечно, «функциональный анализ» (с неизбежным синтезом) и «модульный синтез» (с неизбежным анализом) — это только два аспекта в описании системы. В инженерии (целевой системы, личности, предприятия, сообщества) нужно при анализе и синтезе разбираться со всеми основными (четырьмя или даже пятью) аспектами, и даже большим числом аспектов — ибо кроме основных могут быть и другие. Системное мышление как раз про то, как найти субоптимальное решение синтеза (как функцию воплотить конструкцией, как-то скомпонованной в пространстве и с затратами не выше тех, которые готовы платить внешние проектные роли, а также работами, которые возможно выполнить с имеющимися ресурсами в обозримые сроки), устраивающее все проектные роли.

Оптимума в синтезе всё равно не получите, субоптимальных (то есть заведомо хуже оптимальных, «наименее плохих из всех нам известных», а не «лучших из известных хороших») решений всегда множество примерно одного качества на базе SoTA идей (помним про неустроенности от конфликтов между системами разных системных уровней), вы должны догадаться хотя бы о некоторых субоптимальных решениях, выбрать одно из них. Рациональный выбор проектного решения среди предложенных в ходе синтеза часто тоже называют «анализом», при этом есть ресурсное ограничение на этот выбор решения для синтеза — избегаем analysis

paralysis[376].

Системный/модульный синтез — это всегда многокритериальная оптимизация, которая идёт сразу на многих системных уровнях, в основе её лежат **догадки/гипотезы/изобретения**, которые прошли многочисленные обоснования. **Если вам удалось более глубоко согласовать все описания по всем основным видам разбиений проектируемой системы, и даже добавить какие-то дополнительные виды, большой шанс, что у вас будет (временный, только на текущий момент) успех по сравнению со всеми конкурентами, которые учтут описание по меньшему числу аспектов.**

Одна система, но множество описаний, множество имён: это нормально!

Четверное (это только основные четыре! Их много, много больше!), многократное по разным аспектам восприятие целостной системы нормально, так и должно быть! Один раз рассмотрели систему как состоящую из функциональных частей с их поведением во время использования, другой раз рассмотрели систему как конструктивную с поведением заготовок-конструктивов в ходе их изготовления и сборки, третий раз посмотрели на систему как состоящую из компоновочных мест, далее поглядели, что там с совокупной стоимостью владения, что там за методы работы, а затем и сами работы, и сколько они займут чьего времени, и так далее — и ещё удерживать все эти рассмотрения одновременно, чтобы они были согласованы друг с другом.

Единым в мыслях у разных проектных ролей, имеющих разные интересы, может быть только единонемыслие, поэтому единства разбиения на части, единства рассмотрения системы в проекте не будет.

Реальное единство всех этих описаний даёт только 4D экстенсионализм: «конструктивные части»/конструктивы/продукты/изделия/модули/аффордансы (из чего собрано) занимают в пространстве-времени то же самое место, что занимают «функциональные части»/роли, которые конструктивы реализуют/воплощают. Само это «занимаемое место» — это и есть «размещение», а ещё система что-то стоит (желательно меньше, чем внешние роли, например, заказчик, оценивают наносимую ей непоправимую пользу).

Отношение между ролью и конструктивом — реализация/realization/воплощение. В речи это отношение часто выражается как «играть роль», то есть выполнять метод/функцию как поведение роли. **И конструктивное описание, и функциональное описание — это всё описания одной и той же системы как одного и того же физического объекта в пространстве-времени. Никаких отождествлений конструктивного и функционального описаний через отнесение к каким-то типам, через «определения», как в логике или математике! Все отождествления — только через ход к физической реальности, к физическому объекту в конкретном месте пространства-времени!**

Четыре основных способа выделения частей одной и той же системы дают четыре основных способа организации внимания, выделения главного и отбрасывания второстепенного, и это только основные способы — их ведь обычно больше! Далее **проектные роли должны договориться о том, что все четыре основных**

[376] https://en.wikipedia.org/wiki/Analysis_paralysis

описания, а также остальные (не относимые к основным, но нужные для работы) не противоречат друг другу, описывают одну и ту же систему.

У немецких инженеров-электротехников авторами стандарта системных описаний IEC 81346 был подсмотрен приём: чтобы сразу было видно, о каком виде частей системы говорится, их именуют с использованием специальных префиксов:

- функциональный/ролевой префикс "=",
- модульный/продуктовый префикс "-",
- префикс размещения "+".

А имена объектов какого-то разбиения по какому-то одному отношению даются на каждом уровне иерархии этого разбиения (помним: разбиение — это иерархия по отношению часть-целое). Но в каждом разбиении отношение «часть-целое» строится по одному какому-то типу объектов в разбиении.

Например, =S12=16 означает, что речь идёт о функциональной части 16, входящей в состав (отношение composition/is_part_of) функциональной части S12. Число уровней разбиения в имени не ограничено. Функциональное/ролевое имя инженеры часто называют **тегом** (tag). Проектирование, конечно, ведётся в типах/классах: это имя типов/классов частей системы, а не конкретных частей конкретной воплощённой/реализованной физической системы в конкретный момент времени (экземпляров материального воплощения объектов того типа/класса, который был в проекте/design).

Если имя -M87-K5, то речь идёт о продукте/модуле/конструктиве/изделии K5, который входит в состав надконструктива M87. Это тоже не конкретный физический объект с серийным номером, а тип/класс этих объектов, вид продуктов определённой марки.

Стандарт IEC 81346 закрепил такое именование, чтобы давать короткие **коды** вместо «нормальных» длинных имён. Действительно, если в каком-нибудь Boeing 787 есть 6 миллионов индивидуальных деталей (деталь — конструктив, который не собирается из его подконструктивов), то это сразу означает, что у них будет сразу по нескольку имён, добавятся функциональные имена и имена мест. И тогда лучше бы держать эти имена короткими, и каждое имя чтобы было понятно, к какому типу частей из какого разбиения, и к какому целому для этих частей из какого разбиения в составе самолёта относится.

Стандарт IEC 81346-1:2022 закрепляет современный инженерный консенсус про три основных вида разбиения системы на части. Всего там вводится пять разбиений, кроме функционального, продуктового и размещения даётся ещё «тип» как произвольная группировка по какому-то свойству и «другое», куда входит всё остальное — логистика, стоимостное деление, административное подчинение. Наше стоимостное разбиение по этому стандарту будет в этом «другом» разбиении, other aspects.

Если поглядеть в сегодняшние инженерные проекты, то именование::метод (тонких различий между именованием/naming::метод, идентификацией/identification::метод и десигнацией/означкованием/designation::метод с немного разными предметами всех этих методов мы не будем тут рассматривать) частей системы и всей системы целиком именно так и проходит, а ещё к таким именам и кодам добавляют и другие характеристики: напряжение или давление, типовой размер и даже цену (хотя цена — это не полная замена совокупной стоимости владения, и она не имя, и не

код, но без этого описания про систему уже говорить не принято).

Мы уже знакомились с ситуацией, когда Принц Гамлет и Василий Пупкин могут быть одним агентом-человеком, но называться по-разному в зависимости от того, что нам от него нужно: понять, какая реплика будет следующая в его пьесе (это нужно понимать про роль/функционал, т.е. про Принца Гамлета), когда он планирует выучить новую роль в новом спектакле (это нужно понимать про конструктив, т.е. Василия Пупкина), или выяснить, есть ли у него дети (это нужно понимать про конструктив, т.е. Василия Пупкина, если речь не идёт о «детях из пьесы»). То же можно сказать и о неодушевлённой системе, например, так по-разному названные системы могут оказаться одним и тем же прибором:

- «измеритель давления в пятом контуре охлаждения»::«ролевой/функциональный объект»,
- «манометр KLM-23 завода “Давижмимонтажавтоматика”»::«конструктивный объект»/продукт/модуль,
- датчик в «пятом ящике на третьей полке склада номер 4»::размещение,
- «$300»::стоимость.

В жизни вы вряд ли увидите около таких имён указание на тип объекта из мета-мета-модели, а тут мы указали типы для того, чтобы пояснить разность типов объектов, которые обозначены такими именами, в жизни придётся каждый раз понимать, какой тип объекта назван тем или иным именем.

Разные имена, указывающие на самые разные типы объектов, свидетельствуют о том, что мы будем задействовать самые разные методы работы с этим предметом метода, поэтому для нас один и тот же прибор выступает в разных ипостасях, так что удобно для разных методов работы называть его разными именами. Это нормально, этого в проектах не избежать!

Заставить все роли в проекте использовать одно и то же имя (в том числе один и тот же классификатор, который помогает дать уникальный код), одно и то же разбиение системы на части — невозможно, непродуктивно, это ошибка. Признание ролевого мышления, ведомого ролевым интересом к предметам метода роли, требует множества имён для объектов, множества классификаторов для частей системы.

Но вы должны давать эти имена осмысленно, это не просто свобода именования, когда каждый пишет, как он дышит, ибо «я художник, я так вижу». Это всё культурно-обусловленные имена, «культурно» тут надо читать как культуру из длинного синонимического ряда: метод/способ/культура/стиль/практика работы. Вид труда, инженерии, занятий, а также стратегия и даже сервис — это тоже отсылка к методам, знаниям/теориям/объяснениям/алгоритмам этих методов и инструментарию этих методов. Выбор «культуры» из этого синонимического ряда означает лишь, что метод разделяется довольно большим числом людей, он общепризнан. Так что имена надо не выдумывать произвольно, а давать с опорой на уже накопленное цивилизацией знание об похожих предметах метода. Но поскольку к одной системе можно применить как предмету методов работы много разных методов, то и типов у системы будет много и соответствующих этим типам имён.

Имя «Принц Гамлет — Василий Пупкин» вполне культурно (то есть будет понятно многим людям без дополнительных разъяснений), хотя в нём использовано сразу

два имени отдельных ипостасей актёра: действующего лица и исполнителя роли. Так и имена систем в промышленности часто могут состоять из имён нескольких ипостасей, например, «=S12=16 –M87-K5 $300» (составное имя одной системы, состоящее из трёх частных имён этой системы). Такие имена служат для удовлетворения интересов разных проектных ролей, частные имена перечисляются через пробел. Мы условно цену посчитали тоже именем, хотя это и не совсем так, но интуитивно указание финансовой характеристики около имени не вызывает ощущения чего-то необычного, хотим именно это подчеркнуть.

Обычно такое составное **«системное имя»/«system designator»** можно использовать только в одном конкретном проекте так же, как составное имя «Принц Гамлет — Василий Пупкин» можно использовать только в одном спектакле, в другом спектакле распределение ролей будет другое. Все эти составные имена говорят о каком-то определённом месте в пространстве каждое, так что всегда можно разобраться о том, какая же система названа таким составным именем, конкретизируя ситуацию в физическом мире. Тут бесполезно разбираться, требуя определений для всех составляющих имени, а потом разные объекты в проекте «подводить под определение»! Нет, надо сразу конкретизировать/заземлять, от типов/классов и определений переходить к объектам в физическом мире (или хотя бы примерам объектов в физическом мире).

Вы предупреждены: споры об определениях (часто называются «спор о терминах») бессмысленны, а системное мышление — не математика, задаётся не «строгими определениями» и «однозначными именами»! Бесполезно уточнять определения «как в математике», «как в словаре», «как в науке» и требовать одинаковых имён для

- горячей системы (функциональное имя),
- вчера купленной (продуктовое/конструктивное имя) системы
- дальней системы (пространственное имя),
- дорогой или дешёвой (стоимостное имя) системы.

Ситуацию всегда нужно уточнять, но не через определения, а только через понимание, где расположены эти системы в физическом пространстве-времени: если в разных местах-временах, то это разные объекты, если в одном месте-времени, то это по-разному описанный один и тот же объект, одна и та же система. Иметь много описаний, а не одно, много имён, а не одно — норма в системном мышлении, но системное мышление также предлагает 4D экстенсионализм и заземление/grounding как методы борьбы с путаницей во множественности имён одной системы.

Суть системного подхода в том, чтобы ни в один момент времени не забывать про систему в целом — воплощение/realization системы, которое просто по-разному представляется разными проектными ролями разбитым на части того или иного типа, в зависимости от методов, которыми будут действовать эти проектные роли.

Беглости такого одновременного удержания во внимании разных представлений разных ролей для удовлетворения их разных интересов в одном и том же объекте, беглости работы со множеством описаний одной системы, в том числе и множеством вариантов деления системы на части, нужно некоторое время учиться, хотя сама идея множественности имён для множественности типов (классификацией во множестве классификаторов, удобных для разных методов работы разных ролей) понимается обычно с первого предъявления. Но одно дело — понять, другое дело —

самому свободно владеть таким мышлением, удерживающим внимание на одних и тех же объектах как предметах разных методов работы и поэтому имеющих ещё и разные имена.

Хорошо тут помогает записывать эти объекты внимания: никто не требует удерживать эту системную множественность в голове, записывайте! В системном мышлении обычно «все ходы записаны», это сильно упрощает жизнь. Компьютер не забывчив, записи в нём доступны многим агентам (не только людям!), выполняющим самыми разными методами работы с самыми разными предметами этих методов самые разные роли в проекте, поэтому **системные разбиения всегда документируют, не принято договариваться про системные разбиения «с голоса». Пишите ваши разбиения!**

Альтернативные варианты основных видов разбиения системы на части

Функциональные объекты, конструктивные элементы, размещения, части совокупной стоимости владения — это не единственный способ увидеть в системе части. Этих способов огромное количество.

Исторически в разных школах мысли части системы выделялись очень по-разному:

- ISO 15926 — **два** основных вида частей: функциональные объекты, физические объекты. Остальные могут вводиться по потребности.
- IEC 81346-1:2022 — «по меньшей мере» **пять** (функция, продукт и компонент как заказной продукт, место, тип как произвольная группировка по каким-то свойствам, и «другое» — куда можно включить всё, что душе угодно, например ход на описания создателей: административные, логистические аспекты, связанные с создателями в их графе, а не аспекты самой системы). В нашем руководстве мы опираемся главным образом на этот вариант, добавляя к нему недавно появившуюся стоимость/cost как «другое» и не выделяя особо «группировку по типу» — она по факту неотличима от «другого» (и стандарт предлагает делать для «типа» и «другого» соглашение, чтобы команда понимала, что было выбрано).
- Книга Косякова, Свита, Сеймура «Системная инженерия. Принципы и практика»[377] — **два** вида: функциональный элемент, компонента (конструктивный объект).
- СМД-методология — **пять** (процессы, элементы и связи, внешние функции, морфология, материал)[378]
- … и так далее[379], в среднем 3-7 основных видов частей, ипостасей системы. Впрочем, «ипостаси» тоже называются везде по-разному: структурные аспекты, типы/виды описаний, которые выполняются на заданном типе частей как предметы разных методов в разных школах системного мышления. И везде указывается, что это «по меньшей мере», то есть признаётся, что речь идёт только об основных вариантах, которые нельзя не рассмотреть. А разбиений может быть много, они делаются по потребности, разные в разных

[377] https://www.ozon.ru/context/detail/id/28160736/
[378] http://webcache.googleusercontent.com/search?q=cache:CLJYHzTDPj0J:www.mmk-documentum.ru/glossary/6
[379] https://en.wikipedia.org/wiki/View_model — тут примеры разных вариантов для системной архитектуры предприятия

проектах. Но основные разбиения делаются во всех проектах.

В нашем руководстве мы принимаем вслед за IEC 81346-1:2022, что системы имеют три основные ипостаси/аспекта — функциональную, конструктивную и размещения, добавляем четвёртую ипостась/аспект: стоимостную, это уже де-факто основное описание в подходе **цифровых двойников/digital twins**[380], и дальше продолжаем «и эти четыре — минимальное число аспектов/ипостасей/«вариантов разбиения» как основа для разных вариантов описаний, базирующихся на разных вариантах разбиений.

Также помним, что кандидат на следующее, пятое, основное разбиение, которое ещё не попало в инженерные стандарты как «обязательное», но всё чаще и чаще встречается в реальных проектах — это work breakdown structure из проектного управления. Оно нужно, чтобы было легко потом соединить разбиение продуктов и разбиение работ и проверять: каждая работа выполняется с каким-то продуктом, и нет продуктов без работ и работ без продуктов. Трудности с его введением:

- Проектное управление очень популярно в инженерии, но оно основано на старой концепции «жизненного цикла» (проект подразумевает однократное прохождение жизненного цикла системы, сегодня это заменено на создание MVP и затем непрерывное развитие системы). Проектное управление требует upfront планирования, для большинства проектов это невозможно. Подробности — в руководствах по методологии и системной инженерии.
- Есть огромная путаница методов работ и самих работ. Методы работы в инженерии стали предметом методов ситуационной инженерии методов и были выпущены стандарты инженерии методов, которые посвящены были как раз «модели жизненного цикла». Развитие этого направления привело к тому, что работами стали кейсы, которые бьются на подкейсы, но главное — не требуется их планировать с самого начала. Это означает, что предъявить в какой-то момент WBS (work breakdown structure) в проекте нельзя, если это не проект довольно узкого класса — чаще всего речь тут идёт о строительных проектах. А понимание того, что разбиение методов более базовое (планирование работ ведётся на основе того, что понятно, какими методами ведутся работы), чем разбиение работ — оно ещё не широко распространено.

В презентации 8D digital twin experience от SNC-Lavalin с июньской конференции 2020 года по digital twins приводится 8 аспектов[381], D там — это dimension, «измерение» (ещё одно слово для аспекта/ипостаси, подчёркивается тот факт, что система существует как объект во многих измерениях/dimensions). Всё тут ровно то же самое, что было ещё десяток лет назад в работах по управлению жизненным циклом продуктов (PLM), но сегодня уже разговор про digital twins, а не PLM, ибо акцент уже не только на создании системы, но и на её эксплуатации, и поэтому данные инженерии/engineering data дополняются данными по эксплуатирующимся активам/asset data, а конечная цель — автономная (без человека, где-нибудь в абсолютно безлюдной местности) эксплуатация на основе «единственного источника правды», single source of truth, то есть цифрового двойника как модели системы, поддерживающей согласованные между собой различные описания как целевой системы в её операционном/рабочем окружении, так и создающей системы. Обратите внимание, что до сих пор это «второе поколение системного подхода»,

[380] https://ailev.livejournal.com/1570630.html
[381] https://www.youtube.com/watch?v=OjH1OPezfak

потому как не предусмотрено множество версий целевой системы, то есть не только однократное прохождение «жизненного цикла системы», но непрерывное развитие системы:

- 1D метаданные, документация (тексты). Помним, что в машиностроении под 1D-моделированием понимают физическое моделирование функциональных характеристик системы, но в мире digital twins под 1D-ипостасью системы понимают совсем другое.
- 2D чертежи (там, где «обычные двумерные электронные чертежи» ещё есть, «родные файлы» из CAD/САПР — помним, что digital twins процветают на стройках, а там очень медленно развиваются методы проектирования/design и их инструментарий, 2D представления ещё встречаются повсеместно)
- 3D информационные модели (это представление физической формы/геометрии/компоновки системы с указанием материалов и многих других характеристик)
- 4D «видеоролик» стройки/сборки в развёртке во времени, то есть к 3D добавляется план/schedule сооружения (стройки и монтажа) или сборки
- 5D это ресурсы, material status/cost, то есть добавляется стоимость
- 6D вот тут появляется operation, real time data (и мы перешли из enabling realm/времени создания в operation realm/время эксплуатации, и с этого места принято говорить, что речь идёт не о «системе поддержки жизненного цикла»/PLM, а «цифровом двойнике»/digital twin уже созданной физической системы)
- 7D это live streaming, передача видеопотока работающей целевой системы и её окружения (с видеокамер, со спутника — отдельно от real time data с датчиков, ибо это огромные объёмы данных).
- 8D это уже аналитика, предсказания систем машинного обучения (ML predictive data)

Хорошо видно, что в основе тут не столько динамическое определение методов работы, а затем планирование, сколько сразу планирование работ по созданию системы — предполагается, что это планирование возможно. Всё это делается главным образом для стройки: «один раз спроектировал, один раз построил, модернизация проекта/design и модернизация здания находятся далеко за рамками проекта».

Но тренд очевиден даже в этом узком классе проектов: ипостасей/аспектов, задающих разное разбиение системы на части, всегда больше четырёх предложенных нами (в том числе произвольные группировки по типам и ещё «другое», если не хватило типов), да и сами эти четыре варианта разбиения могут быть сложно устроены внутри себя, это не «одна ипостась — одно описание — один документ». И сами эти ипостаси/аспекты в чистом виде обычно не встречаются, они часто тесно сплетены друг с другом («гибридны»). Системное мышление рекомендует распутывать гибридные описания, восстанавливать начальные предметы интереса/важные характеристики, определять предметы методов работы разных ролей проекта. И, конечно, использовать в этих гибридных описаниях множество имён для одних и тех же систем — как раз для того, чтобы не запутать, а распутать гибридные описания.

Часто гибрид появляется из-за того, что какое-нибудь продуктовое разбиение на очередном системном уровне делают вдруг функциональным, «перескакивая между временами» — разбиение в development-time вдруг продолжают как разбиение в

operations-time. Это нужно отслеживать, ибо «автоматический перескок» может вести к ошибкам, мы их рассмотрим в нашем руководстве чуть позже.

Такого сорта «перескоков» с типа на тип частей системы множество, они ведут к разным вариантам ошибок: разбиение размещения (например, разбивку дома по этажам) вдруг превращают на следующем уровне в продуктовое/конструктивное разбиение (разбивка по единицам оборудования на этаже). Или инженерный документ вдруг превращается в сводно-заказную спецификацию по закупке, и на первое место выходят закупочные цены. Можно ли такое делать? **Можно всё, если вы понимаете, что вы делаете. Если не очень понимаете, то такое лучше не делать — и в любом случае надо понимать, какую ипостась системы вы обсуждаете, какого типа рассматриваемый вами объект-система: функциональный? конструктивный? размещения? стоимостной? иной?**

Для разных ролей в проекте система будет представляться в своих разных аспектах-ипостасях частично, следуя методу того или иного варианта разбиения, но при этом в системном трансдисциплинарном мышлении она будет оставаться целостной/холистичной/целокупностью всех своих ипостасей/аспектов — в терминах типов объектов и отношений этих аспектов (роль и функция, конструктив, воплощение и композиция и т.д.). **Части системы выделяются в физической системе вниманием, внимание ведётся знанием о типах трансдисциплинарной мета-мета-модели и мета-модели предметной области (можно ещё выделить метаУ-модель как в учебниках и метаС-модель как в регламентах). Разность системных разбиений для одной системы — это просто результат разных критериев выделения частей вниманием! А потом на базе удерживаемых вниманием частей системы мы строим различные аспектные и (даже чаще) гибридные описания системы.**

Несовпадение функционального и конструктивного разбиений системы

Разбиения системы по разным видам частей, которые будут предметами методов для разных проектных ролей оказываются тоже разными, ибо «система в глазах смотрящего», а вернее, в глазах не просто смотрящего, но работающего по методу. Каждое разбиение с его способом выделения из мира объектов внимания удобно для обсуждения каких-то одних предметов интереса в работе роли по её методу, и неудобно для обсуждения других.

В рабочих проектах границы систем обычно выбирают таким образом, чтобы они

как функциональная/ролевая часть, продуктовая/конструктивная часть и размещение занимали одно и то же место в пространстве-времени, как изображено на картинке целым кубиком. Но если начинать одновременно разбираться с тем, как такая система работает (её роли, функциональные части и их функции), из чего её сделать (конструктивы/модули, продуктовые/конструктивные части), как она скомпонована (места/размещения), и какие на всё это затраты (стоимостные части), то окажется, что иногда функциональные части, продуктовые части, места в ней полностью совпадают (их и назовут «подсистемы» — подсистемы ведь тоже системы), но иногда такого совпадения не будет — и это намеренно, это нормально!

Так, ТРИЗ (теория решения изобретательских задач) повышает идеальность изобретаемых в ней систем тем, что несколько разных функций (помним: функция — это поведение функциональной/ролевой части системы в ходе работы её надсистемы, то есть функция системы — это функция системы в надсистеме, функция подсистемы — это функция подсистемы в системе) выполняются в ней меньшим числом конструктивов — в идеальной системе все функции выполняются нулевым числом конструктивов (конструкции вообще нет, а функции роли выполняются!). ТРИЗ прямо такую ситуацию называет важной, кладёт её в основу изобретательства. Сходу понятно, что один конструктив может выполнять как аффорданс множество функций, тем самым участвуя в реализации самых разных ролей из функционального разбиения: « — Вот какая польза от этой картины на стене? - От этой картины на стене очень большая польза. Она дырку на обоях загораживает»[382].

Польза/назначение/функция/метод (это всё поведения, ведущие к результатам —

[382] Э.Успенский, «Дядя Фёдор, пёс и кот», https://youtu.be/qPeUokgTRik

переходу каких-то предметов метода в какие-то целевые состояния) могут выполниться самыми разными «конструктивами-в-ролях» — забивать можно и молотком, и камнем. Но и молоток, и камень могут выполнять роль как «забивала», так и пресс-папье, причём в одном и том же проекте. Скажем, плотник забивает молотком-забивалом гвоздь, а затем кладёт этот же молоток, но уже как молоток-пресс-папье на бумажный эскиз собираемой конструкции, чтобы этот эскиз не унесло ветром. Функций две, молоток один — хотя эти роли он выполняет в разное время. Но выполнение разных ролей одним конструктивом в один момент времени тоже вполне частый случай.

Нельзя гарантировать, что системное/функциональное разбиение, а также продуктовое/конструктивное разбиение и пространственное разбиение совпадают. В девяти случаях из десяти они могут совпасть, но у вас будут огромные проблемы в проекте с оставшимся одним случаем из десяти, если вы не будете его учитывать!

Нельзя путать роли/«функциональные части» и конструктивы/продукты. Нельзя путать ролевого Принца Гамлета и конструктивного Василия Пупкина, даже если во время спектакля это одно и то же, они совпадают 1:1. Если вы будете за кулисами говорить Василию Пупкину «Ваше Высочество», а на сцене говорить Принцу Гамлету «как там твои жена и дети?», то окружающие на вас будут смотреть странно, а Василий Пупкин сильно озадачится в обоих случаях.

Мы уже рассматривали заварочный чайник, в котором всего два продукта/детали/конструктива (корпус и крышка) и довольно много функциональных частей (ёмкость, носик, заливочное отверстие в ёмкости, ручка ёмкости, крышка, ручка крышки, паровыпускное отверстие).

Кажется, что трудно перепутать в чайнике функциональные и конструктивные части? Увы, легко! Например, к вам попал список функциональных частей, и вы хотите всё это изготовить. Вы можете попросить изготовить ручку крышки один завод, а крышку — другой завод, если не разобрались. Посмотрите на картинку чайника — это же нелепо, правда? Но если вы никогда не видели чайника, а работаете со списками объектов проекта (не указано каких) и планируете поставки — такого сорта ошибки у вас будут непременно.

Рассмотрим на примере ножниц, что бывает, если люди путают части функциональные (ролевые времени эксплуатации, работающие с окружением по их назначению/функции/методу) и конструктивные/продуктовые/модульные (времени создания, над которыми работают системы создания для их проектирования и

изготовления).

Для ножниц инженеры придумывают разные формы ручки и ножевого блока, думая о ручке и ножевом блоке как о функциональных частях. Они обсуждают ручки и ножевой блок как физически существующие (за ручку держат, проектируются её длина и диаметры колец для пальца и гладкость, а также на парикмахерских ножницах размеры упора для мизинца[383], ножевым блоком режут — проектируются его длина, прочность и острота, угол раскрытия). Если менеджеры отдела закупок будут воспринимать ножницы состоящими из таких функциональных частей как из продуктов/изделий/модулей, то они попытаются заказать работы по изготовлению ручек и ножевого блока раздельно: ручки фирме, понимающей в эргономике, а режущий блок заводу, который умеет делать ножи («ножницы» как раз происходят от слова «нож»!). Инженеры от этого будут в шоке, поскольку они для целей изготовления и сборки начинают думать про ножницы как состоящие из модулей/рабочих продуктов: двух цельных кусков металла специальной формы, скреплённых винтиком. Заказать можно только конструктивы/продукты/изделия/материалы, а ручка и ножевой блок у ножниц только ролевые/функциональные элементы — их роль никто не играет, если нет назначенных на эти роли конструктивов/продуктов/модулей! Конструктивами в ножницах будут только две половинки ножниц (а ещё винтик). Если делать ручки и ножевой блок отдельно как конструктивы, а потом их как-то скреплять, то это будет плохое и ненадёжное инженерное решение. Вот ножницы на момент «как делать» (результат модульного синтеза: конструктивы, которые выполнят функции ножевого блока и ручек):

Менеджеры в ситуациях несовпадения функциональных и конструктивных частей сначала внимают доводам инженеров, но потом... смотрят на ножницы в сборе, видят в действии (эксплуатации, operations) ручку и ножевой блок — и опять пытаются с ними сделать что-то по отдельности не в момент «игры/спектакля/работы» (когда ножницы в сборе и используются — ножевой блок режет, а за ручку их в этот момент держат), а в момент изготовления — менеджера же не волнует время использования, он себя ощущает ответственным за изготовление! Вот почему от менеджеров требуется удерживать внимание не только на создателях, но и на целевой системе в момент её использования!

Чаще всего менеджер пытается разделить работы по сборке ручки и ножевого блока, хотя при соединении половинок ножниц винтиком разделить работы по

[383] https://mustang-professional.ru/parikmaherskie-nozhnitsyi-s-uporom-zachem-nuzhen-upor

ручке и ножевому блоку принципиально невозможно. Промежуточная ошибка тут и в том, что во внимании менеджера (обычно это операционный менеджер) только «ресурсы и работы», но не методы работы. А ведь работы нужны только для того, чтобы был выполнен какой-то метод работы, а метод работы должен привести какие-то предметы в нужные состояния. Инженеры это продумывают, менеджеры — нет, отсюда и ошибки, и образ менеджера в глазах инженера как полного идиота.

Или менеджер может сделать каталог ручек и каталог ножевых блоков и потом пытаться заставить инженеров выпускать эти якобы «детали ножниц» в виде запчастей. Список ошибок и заблуждений тут бесконечен, и эти ошибки не прекратятся: менеджеры постоянно находят ролевые «ножевой блок» и «ручки» в инженерной документации и пытаются поступать с ними как со сборочными/конструктивными единицами.

Правда в том, что в ножницах разные проектные роли усматривают два разных разбиения: одна — функциональной декомпозиции («аналитическая»), а вторая — конструктивного/модульного синтеза/сборки («синтетическая»):

Обратите внимание на префиксы перед одними и теми же именами, они обозначают по IEC 81346-1:2022 как раз разные типы, «=ножницы» тут типа функционального объекта, а «-ножницы» — конструктивный объект. Если имена совпадают, то невозможно по одному только имени понять, о функциональных или конструктивных ножницах мы говорим, обсуждаем ли время эксплуатации или время изготовления: требуется или задать префикс, или определять это из контекста, а если контекста не хватает, то обязательно переспрашивать.

Целевая система едина как функциональная часть и она же конструктивная (полностью совпадают), но вот дальше «функциональная структура»/«системное разбиение» «модульная структура»/«продуктовая декомпозиция»/«конструктивное разбиение» могут существенно различаться.

В инженерных языках моделирования концепции «железной» системы как описания воплощения конструктивными частями ролевых/функциональных частей (раньше говорили «архитектуры», до появления у архитектуры своего собственного предмета работы по оптимизации связности конструктивов/модулей с целью получения приемлемых архитектурных характеристик системы), равно как и в языках моделирования предприятия, функциональные и конструктивные части имеют различные значки, обозначающие разные типы этих объектов. Если не понимать разницу между объектами разных типов, то использование языков

моделирования становится ошибочным и приносит вред, а не пользу. Многие ошибки менеджеров как раз идут от попыток принимать менеджерские (логистические, отвязанные от функциональности) решения по поводу функциональных частей, когда по факту в их власти принятие решений только по поводу конструктивных частей (они планируют ресурсы создателей на создание и развитие конструктивных частей).

Если в классическом проектном управлении сделать up-front (то есть заранее спланированные, когда уже известны все детали изготовления) графики воплощения/реализации/изготовления конструктивов/модулей (скажем, финализации описаний модулей-контейнеров в программной инженерии) и реализации функциональных частей (в программировании они, например, могут называться «фичи», features) — то они будут абсолютно различными, но оба могут иметь смысл! **Главное при этом — это решения создателей системы, какие части конструкции реализуют какие необходимые для работы системы функции/методы работы, начиная тем самым играть какие-то роли (то есть воплощая функциональные объекты). Обычно для этого нужно описать и функции (поведение), и функциональные части (ролевые части, которые выполняют поведение/функцию), и конструктивные части/модули/продукты/изделия, и где это всё размещено (компоновка), и стоимостные части (как распределена совокупная стоимость владения, а заодно и стоимость создания).**

В сложных случаях моделируют отдельно и функции (поведение), и функциональные части (ролевые объекты, которые как-то себя ведут). В более простых случаях обходятся либо функциями/поведением, либо функциональными/ролевыми частями — но никогда только конструктивными частями/продуктами/модулями, ибо предмет интереса «как работает» обязательно должен обсуждаться, его нельзя пропускать. **«Как работает», функциональность (какую пользу приносит система надсистеме, какой сервис делает система для систем в окружении) как область интересов — это главная область/зона интересов в современном системном мышлении, главный предмет интереса, важнейшая характеристика системы!**

Важность функционального разбиения подчёркивается в самых современных исследованиях, где системное мышление используют для объяснения таких явлений, как эволюция и жизнь. Функциональный аспект вводят через понятие аффорданса/affordance[384], подразумевающее единство агента и используемого им для своих целей (выполнения какой-то функции) какого-то подходящего для выполнения нужной функции конструктивного объекта из многих объектов окружающей его среды. Агент понимает, какую функцию может выполнить внешний конструктивный объект, и слово affordance подчёркивает «необъективность» функциональности, её задание зависит от агента, который вменяет функционально-нейтральному физическому/конструктивному предмету окружения какую-то нужную агенту функцию в целевой системе, агент тем самым делает **изобретение**, то есть угадывает, какие конструктивы могут выполнить нужную функцию — это и есть «изобрести», для известной функции подобрать конструкцию, модульный

[384] https://en.wikipedia.org/wiki/Affordance, см. также https://arxiv.org/abs/2110.14602, https://www.activeinference.org/

синтез — это и есть «изобретение».

Моделирование: функциональные и конструктивные разбиения

Приведите пять примеров не совпадающих (как у ножниц) функциональных и конструктивных разбиений каких-то систем из ваших проектов:

Номер п/п	Система	Функциональные части (один или два уровня разбиения)	Продуктовые части (один или два уровня разбиения)
Пример.	Ножницы	=Ножевой блок, =Ручка	-Половинка 1, -Половинка 2, -винтик
1.			
2.			
3.			
4.			
5.			

Разработка концепции системы: функциональный анализ и модульный синтез

Основные решения по устройству системы называют концепцией системы. Обычно концепция системы обрастает деталями, пока точность описания системы не оказывается достаточной для её изготовления на производственной платформе. Помним, что концепция системы «живая», она меняется в ходе развития системы по мере выпуска новых и новых версий системы, то есть в современном мире система не просто создаётся и эксплуатируется в неизменном после её создания виде, но система постоянно модифицируется даже после начала её эксплуатации. Это отражается в том, что вместо «жизненного цикла системы, от рождения до смерти» говорят сейчас о «создании и развитии системы» (создание MVP системы и последующий выпуск множества версий системы с новыми и новыми функциями/возможностями/сервисами, которые дают новые и новые фичи/features). В любом случае, концепция системы разрабатывается начально для выпуска MVP (minimal viable product, минимально жизнеспособная версия системы, то есть версия системы, способная принести хоть какую-то пользу при её эксплуатации), а затем продолжает дорабатываться.

Основное содержание концепции системы заключается в описании совмещения функционального («как работает») и конструктивного («из чего сделано») разбиения, с учётом пространственной компоновки и результирующей совокупной стоимости владения. Но это только основное содержание, а вот дальше напомним, что кандидатом на включение в основное содержание концепции системы будет ещё описание разбиения работ. Кроме этого, рассматривается описание не только работ, но и методов работы — это мы уже выходим не на собственно концепцию системы, а на концепцию системы с учётом создателей из графа создателей. Тем самым концепция системы так или иначе

начинает затрагивать создателей, и всё больше приобретать черты **концепции создания системы** времени эксплуатации (то есть описание создателей системы, а не самой системы), а не просто концепции системы (описание системы в момент её эксплуатации, но затрагивающее какие-то связи с созданием — прежде всего выбор аффордансов, «изобретение»). Но этого разделения на концепцию системы и концепцию создания системы в современной инженерии пока нет, хотя иногда и встречается (скажем, в строительстве есть ПОС, проект организации строительства — это оно и есть, хотя ПОС обычно много детальней концепции).

Границы того, что включают в концепцию системы, определены очень нечётко (подробней о том, как создавать концепцию системы, будет в руководстве по системной инженерии), но консенсус в том, что самое основное в этой концепции — это показ совмещения

- функционального разбиения с учётом его прохождения через ограничения по линии требуемой функциональности (поэтому кроме разбиений обычно включают функциональные описания — какие-то инженерные расчёты характеристик, 1D моделирование)
- конструктивного разбиения с учётом его прохождения через ограничения в наличии аффордансов (поэтому тут часто исследуют наличие заказных конструктивов/компонент/модулей, возможности изготовления элементов конструкции, характеристики конструктивов),
- компоновки с учётом прохождения пространственных ограничений,
- стоимостного разбиения с учётом прохождения ограничений по затратам.

В отечественной литературе концепция использования и концепция системы часто совмещаются под названием **технико-экономического проекта** (а если система, например, блок атомной электростанции, то название может быть другим — ОБИН, обоснование инвестиций, или аван-проект). Инженеры до сих пор спорят, технический ли это проект без экономической части, или технико-экономический. Но наличие обязательности стоимостного описания среди основных системных описаний говорит, что обязательно это технико-экономический проект, чисто технический проект не имеет смысла.

После обрастания технико-экономического проекта подробностями переходят к собственно созданию более детальных проектов — сначала **эскизного** (и иногда даже эскизный проект путают с менее детальной концепцией системы), потом **рабочего** (с точностью, достаточной для изготовления). В русском языке это всё проект/design, поэтому часта путаница с проектом/project. Так что фраза «проект создания системы» двусмысленная по-русски: то ли это проект/project::работа по созданию системы, то ли это проект/design::описание системы. Как это различать? Как всегда: обращать внимание на тип понятия, обозначаемого термином «проект» и уточнять тип из контекста или прямо переспрашивать. Если сами говорите или пишете, старайтесь избегать ошибок из-за этой встроенной многозначности. «Проект скворечника по проекту скворечника» — такое слышишь часто, не забывайте проставить тип к каждому «проекту», это разные типы! И помним, что под словом «проект» часто имеют в виду ещё и третье значение — организация проекта.

Но вернёмся к концепции системы. Она начинается с того, что в ней согласовываются описания четырёх (а часто больше) основных ипостасей системы, описанных через части системы, которые мы находим в основных

разбиениях/breakdowns. Так что для начала надо понять, как проходит вот это совмещение разбиений, для начала сосредоточимся только на двух из них — функциональном/ролевом и конструктивном/продуктовом/модульном.

В литературе для разных разбиений часто используют просто термин «**структура**», но нужно помнить, что это не любая «структура» как какой-то граф объектов и отношений, а именно иерархия по какому-то виду отношения часть-целое как основе разбиения/breakdown. Тем самым «структура» оказывается иерархией по отношению «часть-целое» (is_part_of, composition, состава) объектов функциональных, конструктивных, пространственных, стоимостных и им подобных (не только же основные виды частей системы используются в проектах, системы представляются не только в их основных ипостасях, но и во многих других!). Все другие отношения (специализации, классификации, предшествования во времени, взаимовлияния и т.д.) в этой «структуре» разбиений не учитываются.

Если иерархия показана как разбиение/breakdown, то структура разбиваемого объекта-системы — это именно структура частей-целых! И эти разные виды частей-целых показывают не объекты физической разборки-сборки, но просто указывают на выделение вниманием разного вида частей в физической системе. За одним исключением: конструктивные/продуктовые части, входящие в состав системы (affordances/аффордансы/«подходяшки») не только выделяются вниманием в ходе их взаимодействия при эксплуатации, когда они воплощают какие-то функциональные части, но и указывают на реальную физическую разборку-сборку, ибо они рассматриваются во время создания, где есть и разборка-сборка в ходе изготовления/воплощения системы.

Вот рисунок из стандарта IEC 81346-1:2022 (а этот стандарт взял этот рисунок из более раннего стандарта IEC 1392/09):

Это **базовая схема работы по созданию концепции системы (и она же — базовая схема изобретательства). Тут не учтены компоновка и стоимость, но при реальной работе они тоже обязательно присутствуют, как и варианты разных других специфичных для того или иного проекта разбиений. Но главное тут — переход от функциональной декомпозиции к модульной/продуктовой композиции, от анализа к синтезу. То, что это «переход» и там есть «сначала» и «потом», задаётся линией времени (time line).**

Рассмотрим целевую систему, которую в начальный момент мы рассматриваем как функциональный объект А (подсистема::роль А в надсистеме для А). Первое, что происходит в ходе размышления о будущей системе — это функциональная декомпозиция: функциональные части разбиваются на более мелкие (на рисунке это В и другие четыре безымянные) и делается попытка подобрать для них конструктивные/материальные объекты как аффордансы, которые могут выполнить во время эксплуатации готовой системы функции этих функциональных частей. Вот этот подбор (догадка о том, какой из миллионов материальных/конструктивных

объектов мог бы выполнить нужную роль, причём возможно такой объект надо будет ещё и изготовить, то есть речь может идти не об уже существующем объекте, а только о том, который надо будет ещё специально создать с нужными свойствами) — это и есть **модульный синтез**, а если задача представляется нетривиальной (то есть это не задача, а проблема: непонятно, что делать, если аффорданс сразу не нашёлся), то это называется **изобретательством**.

Типовое поведение молотка — «бить по предмету, передавать энергию бойка этому предмету». Это ожидаемое поведение отлично подходит для функции «забивание» с предметом функции «гвозди» (синоним: метод с его предметом метода, но терминология «метода работ» обычно применяется к одушевлённым предметам или роботам, а вот для простых агентов вроде молотка чаще говорят о функциях).

Изобретательство тут — увидеть, что не только молоток, но и микроскоп может дать ожидаемую функцию «забивать гвозди». И у молотка и у микроскопа есть ожидаемое поведение «бить по предмету, передавать энергию твёрдой массы этому предмету». И даже «камень» тоже может быть использован для воплощения роли «забивала» гвоздей, предложение задействовать для забивания гвоздей камень — тоже изобретение. Изобретатель за неимением молотка может выбрать в качестве аффорданса камень, ибо молотка нет, микроскоп может быть большой и неудобный, а камень легко подобрать по размеру и форме — соображения тут могут быть разные, и отнюдь не только соображения низкой цены камня.

Это ключевой момент: чтобы изобретение было возможно, на текущем уровне развития цивилизации должны существовать известные аффордансы для нужной функции. Если изобретение невозможно, то системы не будет. Хороший тут пример — изобретение вычислительной машины. Функционально с ней всё было понятно ещё Чарльзу Бэббиджу и Аде Лавлейс в начале 19 века. Но вот аффордансы для потребных функций логических операций работали ненадёжно: механика сбоила (это пробовал сам Бэббидж), пневматика не работала, электрические реле тоже сбоили, гидравлика тоже не работала! В итоге сработала электроника, после изобретения: было предложено использовать для создания универсального вычислителя логические вентили на радиолампах-триодах, задействованных не в режиме усилителя (для чего они первоначально и использовались), а в ключевом режиме. Эти лампы были доступны и относительно дёшевы, но изобретены они сами были только 1906 году, а изобретение компьютера на базе радиоламп состоялось только в 1945 году, но примерно половину времени компьютер не работал из-за ненадёжности радиоламп, супер-надёжные радиолампы появились только в 1948 году. Подробней эта история рассказана в руководстве по методологии, она и имеет отношение к стратегированию: вы не должны полагаться на то, что сделаете изобретение в проекте. Наоборот, вы можете начинать проект новой системы, если сделали изобретение! Если вы не понимаете, какова концепция системы хотя бы в общих чертах, если не понимаете, какие есть аффордансы — не начинайте проекта! Возможно, вы будете гениальным изобретателем, но возможно, это вы пытаетесь изобрести летательный аппарат тяжелее воздуха ещё до того момента, как появились лёгкие двигатели внутреннего сгорания (недаром братья Райт, изобретатели самолёта, были владельцы мотоциклетной мастерской). Концепция системы как раз говорит о том, что изобретение сделано, и можно приступать к детальному проектированию.

На рисунке из стандарта IEC 81346 видно, что на первом шаге декомпозиции подобрать конструктивы/продукты/модули с подходящими ожидаемыми функциями

для воплощения/реализации подсистем удаётся только для двух подсистем из пяти (и показано это в левой части как подсистемы первого уровня с жёлтой и голубенькой гранью одновременно: жёлтая грань — описан функциональный аспект, то есть понятно какую выполняют функцию в надсистеме с именем А, а голубенькая грань — понятно, из каких конструктивов в правой части диаграммы эта подсистема будет будут сделана). На следующем шаге декомпозиции на подсистемы это уже удаётся для всех подсистем, в том числе для подсистем подсистемы В. Эта часть функциональной декомпозиции, когда мы пытаемся систему в её функциональной ипостаси разбить не на любые подсистемы, а на такие, какие потом удастся воплотить хоть какими-то аффордансами, иногда называется **функциональным анализом.** Анализ (предполагающий разъединение на части, декомпозицию) — это нормально, если вы не забываете про его подчинённость синтезу (сборке из частей). Анализ плох только если он сам по себе, в отрыве от синтеза, не подчинён целям синтеза. Рисунок из стандарта IEC 81346 как раз показывает, как функциональный анализ/декомпозиция поддерживает конструктивный/модульный синтез/сборку из аффордансов — трудно сразу предложить такой одноуровневый набор конструктивов, который реализует функциональность системы А, это много легче сделать, сначала разбив функциональность по отдельным подсистемам, а затем и конструктивы тоже объединять в целую систему в несколько шагов, через «сборочные единицы»/узлы[385] в какой-то их иерархии, а не одноуровнево.

После (помним стрелку логического времени на диаграмме) функциональной декомпозиции мы должны из всех конструктивов собрать/синтезировать целевую систему-конструктив (систему, рассматриваемую как конструктив), то есть конечный продукт/изделие, который будет выполнять начально заявленную функцию во время использования/эксплуатации/работы системы.

Для конструктива/модуля/узла В’ на рисунке, реализующего функциональную часть В это получается только в два шага — сначала собираются два промежуточных конструктива, и только потом они объединяются. На следующем шаге конструктив/модуль/узел В’ включается в состав конструктива А’, который реализует изначально задуманную систему А (когда говорим тут «система», то имеем ввиду её главную ипостась: функциональную).

Эту часть создания концепции системы, то есть составления целой системы в её конструктивной ипостаси (конструктив/модуль/продукт/изделие/узел) из её более мелких конструктивных частей называют **модульный синтез**. Иногда метод модульного синтеза (синтеза продукта из конструктивов) подают как метод «разузловки», то есть разборки на узлы, но нет — логически «разбираются», то есть декомпозируются именно функциональные части (подсистемы), а для конструктивных частей логически идёт именно сборка в конечный конструктив, то есть движение по диаграмме идёт от целой системы-роли к середине как набору подсистем, реализующих мелкие функции, а затем эти подсистемы воплощаются мелкими модулями/узлами, которые затем объединяются в конечную систему-конструктив (продукт/изделие). Так что речь тут идёт по факту о «сузловке», хотя этот термин никогда не используется. Почему же «разузловка», а не «синтез узлов»? Описываемый нами метод создания концепции системы был осознан инженерами не так давно, а нынешняя производственная терминология появилась

[385] https://ru.wikipedia.org/wiki/Узел_(машиностроение)

много раньше. Инженерам казалось, что они «разбирают в уме» уже готовую конструкцию, но нет — узлы как раз собирают, это «сборочные единицы».

Концепция системы иногда очевидна, ничего изобретать не надо — вы просто знаете уже сделанное раньше изобретение. То, что для вычислительной машины нужно использовать электронику — это сегодня очевидно. Но, конечно, иногда приходится и изобретать (чаще всего, повторимся, не в ходе выполнения какого-то проекта, а перед его появлением — изобретения лежат в основе инженерных проектов, а не являются результатом проектных работ!).

В ТРИЗ[386] как теории изобретательства важно, что небольшое количество конструктивов будет выполнять большое количество функций (т.е. степень идеальности системы растёт: идеальная система — это та, функции которой выполняются, а конструкции у неё нет, нет материального воплощения, модулей/аффордансов в системе ноль, изготавливать ничего не надо). Так что в ТРИЗ заведомо соотношение между функциональными и конструктивными элементами не 1:1 (как и в наших примерах заварного чайника и ножниц). **Желательно научиться много функций навешивать на один конструктив/модуль/продукт/изделие, уменьшая их количество в системе.**

Лучше всего это удаётся с компьютерами и роботами — поэтому общий тренд инженерии — в создании универсальных модулей, которые легко перепрограммировать для новых функций. В биологической эволюции природа пришла к более-менее универсальной и сложно устроенной клетке, самые разные функции которой (функции клеток печени, клеток мозга, клеток лёгких) получаются небольшим перепрограммированием. В техноэволюции смотрите на смартфон, компьютер и быстро развивающиеся идеи универсальных создателей, которые сегодня чаще всего проектируются как (само)программируемые интеллектуальные антропоморфные роботы.

В изобретательской деятельности мы решаем программу многоуровневой оптимизации, снимая конфликты между интересами, которые могут быть по поводу разных системных уровней. **John Doyle с сотрудниками указывал, что в природе всегда есть противоречия (из них он особо выделял противоречие между скоростью и точностью: бывают быстрые неточные части системы и медленные точные части) и это противоречие может быть снято использованием многомасштабных конструкций с многоуровневыми обратными связями, в один уровень такое противоречие не снимешь. Поэтому если нужно сделать быструю и точную систему, она по необходимости становится сложной — и там много изобретательской работы, но эта работа поддержана математикой, им разработанной[387].**

Обычно из имеющихся в наличии или легко изготавливающихся конструктивов трудно подобрать такие, которые при сборке их в готовую систему-конструктив реализуют все необходимые функциональные части, дающие во время эксплуатации систему со всеми необходимыми функциями. Для этого приходится нарушать логический порядок создания концепции системы, то есть менять уже разработанный вариант функционального разбиения на новый, в котором проще

[386] https://ru.wikipedia.org/wiki/Теория_решения_изобретательских_задач
[387] https://ailev.livejournal.com/1622346.html

выполнить модульный синтез. Тем самым в жизни логический однократный метод создания концепции системы (сначала функциональная декомпозиция, потом модульный синтез) по факту оказывается итерационным: идут попытки функциональной декомпозиции, но в случае неудачи модульного синтеза на основе текущего функционального разбиения возвращаются к анализу — и дальше пробуют провести модульный синтез, то есть сделать изобретение, подобрать аффордансы, опираясь уже на новый вариант функционального разбиения. **Опора на другой набор функциональных объектов по факту эквивалентно предложению другого способа работы устройства, другого принципа работы.** И тогда для реализации других функций можно подобрать другие конструктивы, более доступные.

Скажем, универсальный компьютер можно реализовывать как набор функциональных подсистем квантового компьютера или классического логического компьютера. Аффордансы (конструктивы) для хорошо и надёжно работающего квантового компьютера неизвестны. А вот аффордансы для классического компьютера — известны, это сегодня транзисторы, которые можно изготавливать массово в электронных чипах-микросхемах. Так что сначала мы задаём какую-то декомпозицию компьютера на подсистемы, а потом проверяем: есть ли дешёвые аффордансы (и есть ли они вообще сегодня), и если нет — пытаемся поменять функциональную декомпозицию. Ну, или можно надеяться на то, что мы в какой-о момент сделаем изобретение — и найдём аффордансы для текущей функциональной декомпозиции, но тут можно промахнуться на сотню лет (как при изобретении компьютеров или самолётов).

Конструктивы/узлы/модули/продукты/изделия приходится разрабатывать и изготавливать, если их нельзя купить готовые. **Купить — это простейший способ изготовления конструктивов!** В IEC 81346-1:2022 даже предлагают «продукт», если предполагается его закупка, называть «компонентом», настолько это частый случай. Терминология тут может быть самая разная, в другой литературе компонентом (или компонентой, в русском языке допустимы и мужской и женский род для этого слова) называют как раз функциональную часть, конструктивную часть будут называть «комплектующее», а предмет закупки будут так и называть — «предмет закупки».

Вопросы. Функциональные, конструктивные, пространственные, стоимостные части - 1

Инженер Лида долго разрабатывала принципиальную схему изделия и попыталась представить её коллегам и руководству в качестве начальной концепции системы. У неё не получилось. Чего не хватило?
--- другие инженеры просто придирались, принципиальной схемы достаточно ### Нет, принципиальная схема расскажет о функциональных частях системы и их связях, а нужно ещё рассказать о конструкции системы: какие конструктивы воплощают какие роли. А ещё нужно что-то сказать о компоновке/размещении, оценить совокупную стоимость владения.

--- нужно было ещё продемонстрировать расчёты режимов по принципиальной схеме, показывающие реализуемость (feasibility) концепции использования ### Нет, для концепции системы нужно ещё

продемонстрировать связанную с принципиальной схемой (трассированную к принципиальной схеме) конструкцию и добавить информацию о размещении частей системы, а также оценить совокупную стоимость владения.

--- нужно было добавить список модулей ### Конечно, нужно! Но ещё не хватает размещений — например, нужно приложить компоновку. И ещё что-то сказать про экономические характеристики (совокупную стоимость владения).

--- нужно было добавить конструкцию системы, а также продемонстрировать расчёты режимов по принципиальной схеме ### Нет. Мы узнаем многое про функциональные части системы и их связи, конструкцию и её интерфейсы, но до концепции системы всё ещё не хватает информации о физической компоновке — размещениях, и хорошо бы что-то добавить про стоимость системы.

+++ нужно было добавить модули, добавить компоновочный эскиз и согласовать их друг с другом, дать оценку полной стоимости ### Да, функциональные части и их связи, модули, а также их размещения и ожидаемая стоимость — это и есть начальная концепция системы.

--- нужно было добавить продуктное разбиение, а также продемонстрировать расчёты режимов по принципиальной схеме, обосновывающие достижимость концепции использования с учётом характеристик задействуемых модулей, добавить компоновочную диаграмму и согласовать их друг с другом, приложить технологические рекомендации для предприятия-производителя, а также оценку длительности работ ### Нет, всё равно недостаточно, хотя и много лишнего для начальной концепции системы: про стоимость всё-таки ничего не сказано, а это одно из основных описаний, в начальную концепцию системы должно войти.

Первичное рассмотрение целевой системы в составе надсистемы проводится как:
+++ Функциональной части ### Да, система в составе надсистемы рассматривается по её роли в окружении, а роль выполняет какую-то функцию/метод/сервис в ходе эксплуатации/использования системы.

--- Конструктивной части ### Нет, конструктивное рассмотрение вторично, хотя целевая система одновременно представляет собой совпадающие функциональную и конструктивную часть надсистемы как разные ипостаси одного и того же физического объекта. Первичное рассмотрение — как функциональной/ролевой части в какой-то надсистеме.

--- Физического объекта ### Нет, рассмотрение состава надсистемы как состоящей из физических объектов само по себе не системное, это просто удобный способ договариваться о том, что разные системные объекты

(функциональные, конструктивные, пространственные) описывают одну и ту же систему, воплощённую в физическом мире.

--- Стоимостного объекта ### Нет, рассмотрение состава системы как набора каких-то имеющих свою цену объектов может быть, например, в целях бухгалтерского или управленческого учёта, для оценок рынка маркетологами, но это не первичное рассмотрение. Первичное рассмотрение — это система как ролевой объект: какую роль она играет в надсистеме, выполняя какую-то функцию/метод/сервис этой роли.

Инженер Марфа обнаружила, что в переданной ей документации на принципиальной схеме 14 функциональных частей, а на печатной плате 11 модулей, и всё работает. Как это получилось?
--- перепутали или принципиальную схему, или печатную плату ### Нет, ничего не перепутали, это обычная ситуация: один модуль играет несколько ролей, то есть выполняет несколько функций. Более того, инженеры к такой ситуации стремятся.

+++ какие-то модули могут выполнять несколько функций, нет повода для волнений ### Да, это обычная ситуация, волноваться не надо. Если модуль выполняет несколько функций, то это просто означает, что один модуль исполняет роль сразу нескольких ролевых/функциональных частей. Идеальная система — это в которой функций (полезных для надсистемы поведений системы как функционального объекта/части надсистемы) бесконечно много, а модулей и вовсе нет (даже волшебная палочка в этом плане неидеальна: функций у неё бесконечно много, но они выполняются не нулём, но всё-таки одной конструктивной частью — самой палочкой). Самая лучшая победа в войне — это когда ты побеждаешь без единого выстрела.

--- нужно найти и исправить ошибку в принципиальной схеме, там могут быть лишние функциональные части, а печатную плату не нужно переделывать, ибо она работает ### Нет, различное число функциональных и конструктивных частей не является ошибкой. Если модуль выполняет несколько функций — это нормально.

--- нужно найти ошибку и переделать печатную плату, ибо «всё работает» — это наверняка неправда. Скорее всего, не все функции выполняются. Нужно тестировать работу платы более тщательно ### Нет, различное число конструктивных и функциональных частей системы не является ошибкой. Если модуль выполняет несколько функций — это абсолютно нормально.

Инженер Олег разработал принципиальную схему изделия, провёл расчёты и обнаружил, что на рынке невозможно купить модули, реализующие функции этой схемы. Он может далее:
+++ переработать принципиальную схему, чтобы она соответствовала доступным модулям ### Да, это наиболее предпочтительный вариант,

лучший способ создания модулей — это купить готовые модули и не связываться с разработкой.

+++ спроектировать отсутствующие на рынке модули и изготовить их ### Да, такое часто бывает, хотя купить модули обычно предпочтительней — но если их нет, а принципиальную схему поменять нельзя без ухудшения характеристик целевой системы, то делать нечего: нужно проектировать и изготавливать недостающие модули.

+++ найти наиболее дешёвый и быстрый вариант сочетания переработки принципиальной схемы и изготовления недостающих модулей ### Да, обычно так и поступают: итеративно проводят функциональный анализ/декомпозицию и модульный синтез, чтобы получить и хорошие функциональные характеристики системы, и приемлемую стоимость/время изготовления модулей.

--- доложить начальству о нереализуемости проекта ### Нет, ещё рано сдаваться: нужно пробовать переработать принципиальную схему, нужно пробовать разработать и изготовить недостающие модули.

Инженер Лена и снабженец Богдан долго спорили: выбранные Богданом к закупке покупные изделия никак не соответствовали принципиальной схеме продукта, подготовленной Леной. Тем временем уже пора начинать закупку. Что делать?
+++ отложить закупку, Богдан не прав, и надо будет провести новую закупку ### Да, в любом случае нужно закупку отложить — и уже потом разбираться, что можно сделать ещё. Если провести закупку, то собранные в систему покупные изделия не выдадут нужные функции, поскольку они не соответствуют тому, что предусмотрено в принципиальной схеме продукта.

--- Лена должна доработать принципиальную схему, и нет слова «нельзя», есть слово «надо». Закупку продолжать. ### Нет, обычно исправить принципиальную схему, чтобы она строго соответствовала выбранным наугад покупным изделиям, довольно трудно, и не факт, что это закончится успехом. Закупленные изделия останутся невостребованными, систему-продукт с нужными функциями из них не построишь, придётся тратить деньги второй раз: закупать изделия, соответствующие принципиальной схеме продукта.

+++ обоим наступить на горло собственной песне, и договориться: изменить и закупку, и принципиальную схему ### Да, закупка и принципиальная схема должны быть согласованы, скорее всего, придётся менять и закупку, и принципиальную схему — функциональное и конструктивное разбиения системы обязаны быть между собой согласованы, но принципиальная схема всё-таки логически должна быть первой.

--- продолжать спорить, пока начальник не уволит обоих за неграмотность ### Нет, не нужно проявлять безграмотность и спорить в этом случае. Функциональное и конструктивное разбиения системы обязаны быть между собой согласованы.

+++ пригласить конструктора Тимура, который им расскажет про ограничения по компоновке, что затянет достижение договорённости ещё на пару дней ### Да, причём это самый правильный вариант. Если модули (даже согласованные с принципиальной схемой) не будут соответствовать ограничениям компоновки — можно потерять и деньги на закупленные модули, и время на удовлетворение неучтённых компоновочных интересов конструктора. Это потерянное время будет много больше двух дней. Отговорки, что это время будет терять не «наша команда», а «их команда», которая вдруг столкнётся с неучётом компоновочных интересов при сборке надсистемы, и «они что-нибудь там придумают» — это неправильные отговорки.

«Для наиболее эффективного разделения частот перед усилителем применяется преобразование эллиптического фильтра Кауэра.» О каких описаниях системы идет речь?
+++ Функциональных ### Да, обсуждается принцип работы системы, математические преобразования обычно описывают именно функциональные элементы.

--- Конструктивных ### Нет, не обсуждается, какими именно конструктивными единицами/модулями/продуктами/изделиями (лампами, транзисторами, микропрограммами в контроллере) реализовано преобразование сигнала, что из чего и как собрано, речь не идёт о конструкции. Обсуждается «как работает» (функциональное описание), а не «из чего сделано» (конструктивное описание).

--- О размещении ### Нет, тут ничего не говорится о размещении каких-то конструктивных единиц в физическом мире (а слова «перед усилителем» описывают путь сигнала через функциональные части/единицы).

--- О стоимости ### Нет, о совокупной стоимости владения мы не узнали ничего.

--- О топологии ### Нет, топология системы тут вообще ни при чём.

"Предлагается установить на крышах домов солнечные батареи. Для хранения накопленной в течение дня энергии предлагается включить в цепь в качестве накопителей энергии не только стационарные аккумуляторы, но и электромобили, находящиеся во владении жильцов домов». Что такое в этом тексте «стационарные аккумуляторы» и «электромобили»?
--- функциональные части ### Нет, функциональная часть тут «накопитель энергии», названная по выполняемой функции.

Стационарные аккумуляторы и электромобили выполняют роль этой функциональной части, но сами они конструктивные части. Конструктивные части как аффордансы выполняют роль функциональной/ролевой части — именно это и написано в тексте.

+++ модули ### Да, электромобили и стационарные аккумуляторы — это конструктивы/продукты/модули, выполняющие роль «накопитель энергии».

--- размещения ### Нет, ничего не говорится, где именно будут размещены аккумуляторы и электромобили. Размещение («на крыше») указано только для солнечных батарей.

--- какие-то части затрат на систему ### Нет, это не описание совокупной стоимости владения.

Вопросы. Функциональные, конструктивные, пространственные, стоимостные части — 2

Каталог онлайн-магазина поделён на товарные категории («Напольные покрытия»), далее на товарные группы («Ламинат»), в которых перечислены товары («Ламинат 31 кл Kronospan Kronofix Дуб Ройял 2,47 м.кв. 7 мм»). Если рассматривать ассортимент магазина (совокупность товаров, доступную к продаже через данный онлайн-магазин) как целевую систему, то товарные группы в каталоге будут:
--- Конструктивными объектами ### Нет, в каталоге находятся только описания конструктивов, сами конструктивы (ассортимент) лежат на складе.

--- Функциональными объектами ### Нет. Мало того, что в каталоге находятся разбитые на части описания, а не сам ассортимент, в задаче на уровне товарной группы речь уже не идёт о функциях, то есть о том, для чего используются товары в момент использования. Там описаны конструктивные объекты.

+++ Описанием конструктивов ### Да, товарная группа в каталоге описывает относящиеся к ней товары как лежащие на складе конструктивы, составляющие ассортимент магазина.

--- Описанием функциональных объектов ### Нет, товарная группа в каталоге не описывает функциональности товаров, только сами товары как конструктивные/материальные объекты.

Инженер сомневается, как ему было бы правильно назвать разрабатываемую им подсистему жилого дома: «напорная система» или «насосный агрегат» (состоящий из двух частей: насоса и мотора). Где может быть ошибка?
+++ Насосный агрегат — это конструктив/материальный объект, указание на закупаемый модуль, так называть подсистему нельзя. ### Да, нельзя. Основное название должно быть «напорная система», по

первичной функции — в задаче подчёркивается, что инженером разрабатывается не конечная система (которую иногда удобней представлять как конструктив и называть так, если это именно конструктив, который делается на продажу в состав неизвестной системы, но и тут лучше как-то ориентироваться на название по предполагаемой функции в окружении), а подсистема, назначение которой известно.

--- Напорная система — это описание функции, неясно как она реализуется (например, это может оказаться конструктивно и водонапорным столбом), так называть подсистему нельзя — никто не поймёт. ### Нет, называть подсистему лучше «напорная система», так как речь идёт об объяснении назначения подсистемы в ходе её работы/функционирования в составе жилого дома. И все поймут.

Маркетолог завода-изготовителя сомневается — как назвать продукт в каталоге завода: «система напора» или «насосный агрегат».
--- Надо назвать «система напора», а «насосный агрегат» — это конструктивный/материальный объект, такие названия использовать в каталоге производителя оборудования нельзя ### Нет, в каталогах продуктов используются именно названия конструктивов/продуктов/изделий/модулей. По описанию функции покупатель может не понять, что он заказывает как элемент конструкции (скажем, это только насос или насос с мотором — насосный агрегат), а производитель может только догадываться о функции, которую будет выполнять его система (например, насос или насосный агрегат могут использоваться для откачивания воды из какого-то водохранилища и тем самым наоборот, уменьшения напора).

+++ Надо назвать «насосный агрегат», а «система напора» — это описание функции, неясно как она реализуется (например, она может реализоваться конструктивно и водонапорным столбом), такие названия использовать в каталоге нельзя ### Да, для каталога продуктов чаще всего выбираются продуктовые названия, названия конструктивов. Покупатель понимает уникальную функцию своей подсистемы, и для её воплощения подбирает какие-то конструктивы как аффордансы. К производителю он обращается за этими конструктивами (причём сервис этого продукта, когда его начнут эксплуатировать, всё равно угадывается в слове «насосный»).

--- Всё равно, можно использовать любое из них ### Нет, не всё равно. При заказе продукта надо понимать, что в него входит, а что не входит. В насосный агрегат входит мотор, а не только насос. Если же заказывать «систему напора» — это может оказаться и просто насос, и насосный агрегат, и водонапорный столб. Так что необходимо использовать название конструктива, а необходимая функциональность конструктива в момент воплощения им какой-то роли в системе заказчика этому заказчику и так известна.

Молодой инженер Михаил заметил, что в проектной документации на установку есть два наименования одной и той же подсистемы — «насосный агрегат» в закупочной ведомости и «система напора» на функциональной диаграмме. Старый инженер Фёдор сказал, что это нормально. Что должен сделать Михаил?
--- Немедленно доложить начальству и рассчитывать на премию за замеченную ошибку ### Нет, разные проектные роли могут по-разному называть подсистемы и даже у одной подсистемы может быть несколько имён. Тут функциональное и конструктивное рассмотрение используют разные названия: одно удобно для объяснение назначения системы, а другое для объяснения её материального состава.

+++ Вспомнить, что у системы может быть множество имён, и больше не позориться ### Да, имена подсистемы как функциональной части и выполняющего её роль модуля как конструктивной части могут различаться. Но это один и тот же объект, одна и та же система, что легко выясняется, если понимать, что они занимают одно и то же место/объём в пространстве-времени. У системы может быть множество описаний, и в этих описаниях она может иметь множество имён, это ничему не противоречит (наоборот: это соответствует системному подходу). Главное — это понимание, когда какое описание и имя использовать.

Системный администратор при подключении к корпоративной сети ноутбука нового сотрудника не нашёл свободных разъёмов в свитче, расположенном рядом с рабочим местом. Для решения проблемы он убрал старую модель свитча с 12 портами и поставил новую, на 18 портов, при этом сетевые адреса всех подключённых к свитчу рабочих станций и ноутбуков изменились. Какой объект системы изменил системный администратор?
--- Функциональный объект ### Нет, никакие функции частей корпоративной сети тут не изменялись, адреса тут не функциональных, а материальных частей.

+++ Конструктивный объект ### Да, поменялся продукт/модуль/конструктив: старый экземпляр свитча был заменён на новый, функциональность свитча не изменилась.

--- Экземпляр свитча ### Нет, обычно про замену экземпляра говорят, когда замена идёт на что-то однотипное, например, экземпляры продукта одной марки/типа/вида/«заводской модели». Тут поменялся не просто экземпляр конструктива, но и тип этого конструктива, так что речь идёт не просто о смене экземпляра (хотя формально это, конечно, смена экземпляра одного типа конструктива на экземпляр другого типа конструктива — но в системном мышлении правит не формальная верность, а верность для какого-то действия. На производстве про подобные действия не говорят, «замена экземпляра», на это и ориентируемся).

Инженер Петя долго искал трансформатор нужной мощности, но не смог найти подходящий, т.к. его не устраивали массо-габаритные характеристики моделей, представленных на рынке. Какие описания Петя уже разработал на данный момент:
+++ Функциональное ### Да, раз нужная мощность ему уже известна. Он понимает, что будет происходить в момент эксплуатации трансформатора.

--- Конструктивное ### Нет, он как раз его и разрабатывает, раз ищет нужный конструктив/продукт/изделие трансформатора.

+++ Пространственное ### Да, раз он уже понимает, какие габариты ему подходят, т.е. где он будет этот трансформатор устанавливать.

Моделирование: изобретение (поиск аффорданса)

Опишите в таблице воплощение/реализацию трёх функциональных частей вашей системы продуктовыми её частями для целевой системы из вашего проекта и дайте не меньше двух альтернативных вариантов, которые вы рассматривали.

Номер п/п	Функциональные части	Конструктивные части, реализующие функциональные части	Альтернативные варианты реализации, которые вы рассматривали
Пример	источник электроэнергии	аккумуляторная батарея	Солевые батарейки, сетевой блок питания
Пример	Охлаждение процессора	Вентилятор	Радиатор, тепловой насос
1.			
2.			
3.			

Альфы и артефакты/«рабочие продукты»

Стандарт OMG Essence[388] предлагает для контроля за изменением состояния проекта особый вид функциональных объектов — **альфа** (ALPHA). Если мы хотим как-то связно мыслить о том, насколько продвинулись создатели в создании и развитии системы, то согласно стандарту OMG Essence нам нужно удерживать

[388] http://www.omg.org/spec/Essence/

внимание в проекте не на любых произвольных объектах, а на альфах, которые они определяют как предметы метода.

OMG Essence — это стандарт описания методов разработки, один из последних стандартов так называемой инженерии методов[389], а потом и ситуационной/ситуативной/situational инженерии методов[390]. Основное отличие OMG Essence от предыдущих стандартов инженерии методов (их было множество, все они безнадёжно устарели) в том, что:

- Он создан не для удобства разработчиков методов (method engineers), а для удобства применяющих описанные в нём методы. Описания методов по этому стандарту удобно использовать в работе, а описания методов по прошлым стандартам было удобно разрабатывать, но не очень удобно использовать в работе по методу. Ключевым было то, что стандарт предлагает отслеживать выполнение метода через чеклисты, контролирующие прохождение предметами метода какого-то ожидаемого набора состояний. Применяющему метод сразу понятно, что делать: нужно проводить предметы метода по состояниям согласно пунктам чеклиста.
- В состав стандарта вошёл не только набор типов предметной области «методология» (language в терминах OMG Essence), но и пример самого верхнего уровня описания современной разработки софта с использованием этих типов (kernel в терминах OMG Essence). Предыдущие стандарты инженерии методов ограничивались только набором типов (language), но не давали пример использования этих типов (kernel).

В нашем руководстве мы рекомендуем использовать OMG Essence, но берём из него только language, а не описанный в нём пример kernel — ровно как этого и требует стандарт, ибо для каждой новой предметной области разработки надо бы разрабатывать свой kernel, а не брать описание метода разработки софта из стандарта. Подробней об этом будет рассказано в руководстве по методологии.

Альфой стандарт называет важный предмет метода, меняющий своё состояние в ходе проекта. Альфа нужна для того, чтобы отмоделировать явно этот важный предмет метода как объект коллективного внимания, требующий отслеживания своего состояния в ходе проекта. Метод характеризуется своими знаниями/теориями/объяснениями/алгоритмами, в которых как раз описываются предметы метода, которые по ходу применения/задействования метода в ходе работы мастерства и инструментария агента, выполняющего метод, будут менять своё состояние.

Альфа была названа авторами стандарта OMG Essence как «ALPHA», ибо нужно было хоть какое-то слово для предмета метода. Поэтому было подобрано короткое слово для объекта «нечто важное для проекта, что меняет своё состояние», нужно было, чтобы у слова не было какого-то бытового значения, не хотели путаницы. Много позже момента, когда в проекте стандарта OMG Essence появился термин «альфа», к нему разработчики стандарта придумали расшифровку термина ALPHA как аббревиатуры, но эта расшифровка абсолютно неважна, мы её даже не будем тут приводить (просто придумана, чтобы хоть как-то оправдать выбор термина, отсылка к «здоровью проекта», тоже произвольно взятой метафоре).

---

[389] https://en.wikipedia.org/wiki/Method_engineering

[390] https://ailev.livejournal.com/750878.html

Альфа — это предмет метода, состояния которого важны для проекта настолько, что требуется обязательно их коллективно отслеживать в ходе всего проекта. Сам проект в целом при этом считается работой команды проекта по совокупности всех используемых в нём методов. Этот совокупный метод называют **методом/процессом/практикой инженерии** или **инженерным методом** (engineering process), иногда сюда не включают эксплуатацию и даже изготовление и говорят о **методе разработки/development process**. Для нас это «масло масляное», потому что инженерия сама по себе и есть метод, но в языке часто работу по методу и сам метод путают, поэтому когда хотят подчеркнуть, что имеется в виду не работа по методу, а сам метод, вставляют дополнительно какой-то из синонимов метода или указывают на то, что речь идёт о специфике/специализации метода, добавляя «вид». Так «труд» — это метод, но иногда используется как «работа» в бытовом его употреблении. Когда хотят подчеркнуть, что это всё-таки метод, говорят «вид труда» или «метод труда». Это чаще всего делается с терминами, для которых нельзя сказать «метод работы», например «труд работы», «инженерия работы», «деятельность работы», «сервис работы». С «процесс работы», «практики работы», «культура работы», «стиль работы», «стратегия работы» такое не происходит.

Предмет метода может быть как абстрактным объектом (описанием), так и физическим функциональным (метод работает с функциональными объектами времени работы агента, выполняющего метод, а они существуют только тогда, когда их роль выполняется конструктивами), поэтому стандарт OMG Essence для материальных объектов, обобщающих понятия документа (материальный носитель информации) с описаниями объекта и конструктива для функционального объекта предлагает тип **рабочий продукт/артефакт** (work product, artifact, т.е. предмет искусственного происхождения). «Рабочий продукт»/артефакт — это то, над чем работаем в физическом мире, что можно обнаружить в окружающем мире. Хотя физические реализации некоторых альф трудно представить «продуктами» (например, альфа «внешние проектные роли» реализуется живыми людьми, трудно думать о них как о «рабочих продуктах» или «конструктивах»), смысл сохраняется: даже если «продуктом» выступают живые люди, вам нужно над ними работать, менять их состояния в ходе проекта. Оргзвенья (команды и коллективы как команды команд, в которых есть хотя бы какой-то один интеллектуальный агент — человек, AI-агент или робот, или даже организация как внутренние оргзвенья) — это тоже конструктивы.

Ещё будьте внимательны к словам: продукт/изделие/система и «рабочий продукт»/артефакт всё-таки разные понятия: продукты обычно физичны, а рабочие продукты, несмотря на физичность документов, всё-таки могут быть описаниями. Если вы в продуктовом магазине вместо бутылки молока как продукта вдруг получите листочек с лабораторными характеристиками этой партии молока, то вы удивитесь, но для проекта создания молока это будет вполне себе рабочий продукт. «Рабочий продукт» — это всё-таки не «продукт».

Особенность альф в том, что они отслеживают состояния предметов составных методов с разными предметными областями их частных методов, а также разными онтологическими типами, и делают это крайне неформально. Так, альфа целевой системы (например, здания) в самом простом случае может иметь

- состояние «задумано», когда у него ещё нет даже подробного описания, возможно даже нет концепции использования, ещё не собрана команда,

формально проект здания ещё не начат, ресурсы на разработку не выделены!
- затем будет состояние «проект/design разработан» — здания как физического объекта нет, но есть только описание здания, достаточно детальное для его строительства,
- затем может быть состояние «построено», то есть здание готово (намеренно опускаем прохождение инженерных обоснований, продажи площадей здания и т.д.)
- затем состояние «эксплуатируется/используется»
- затем состояние «модернизируется». Не уточняем там подсостояния предметов методов проекта капитального ремонта, косметического ремонта и т.д., важно просто подчеркнуть, что после того, как здание перешло в эксплуатацию, с ним может ещё что-то происходить, а потом будет опять возврат к состоянию эксплуатации/использования — альфа отражает состояния предмета метода создания системы не просто как «цепочка переходов между состояниями», это полноценный граф состояний, с циклами, соответствует концепции «непрерывного всего» (прежде всего — непрерывного ввода в эксплуатацию, не только создание MVP системы, но и развития системы).

В альфах названия состояний предмета метода даются через названия более мелких методов по изменению состояния, но эти методы даны как результаты прохождения работ по этих методам — переходными глаголами прошедшего времени («дверь открыта», «дверь закрыта», где открытие и закрытие — методы работы с дверью).

Например, Василий Пупкин в ходе включения в команду проекта каким-то методом (чаще всего этот метод в части привлечения людей называют onboarding[391]) должен принять роль финансиста заказчика (должна появиться оргвозможность финансиста заказчика) и в ответ на вопрос «быть или не быть?» сказать «быть!» и поставить подписи на документах. Альфа «финансист заказчика» нужна для отслеживания состояний объекта «финансист заказчика» во время всего проекта:

- вот мы узнали о том, что в проекте «необходимость финансиста заказчика признана командой», первое состояние всегда самое сложное, чаще всего это «вот ничего не было, а теперь мы знаем, что должно бы быть»
- вот «сформулировано описание внешней проектной роли финансиста заказчика», по факту это описание метода работы финансирования проекта заказчиком,
- вот «оргзвено, выполняющее роль финансиста заказчика выбрано», а в жизни это будет Вася Пупкин (когда проектируются состояния, ещё нельзя написать, что это будет выбран Вася Пупкин на должности зам.начальника финансовой службы заказчика, поэтому говорим о типе объекта)
- вот «исполнитель роли финансиста заказчика занял роль», это состояние Васи Пупкина, которому сообщили о том, что он теперь финансист заказчика, он знает о том, что ему надо делать, а также согласен работать по указанному в описании роли методу финансирования проекта со стороны заказчика.
- вот «финансист заказчика сотрудничает» (тут «сотрудничает» означает важное состояние после выполнения практик лидерства — нет переходного глагола прошедшего времени, как обычно), то есть Вася Пупкин уже работает

[391] https://en.wikipedia.org/wiki/Onboarding

над проектом — говорит своё «быть!» и ставит подписи, не задерживая никого в проекте.
- вот «все работы финансиста заказчика закончены», Вася Пупкин покинул проект.

Команда проекта потихоньку двигает альфу «финансист заказчика» по этим состояниям, применяя главным образом метод onboarding (это неважно, что роль внешняя, а не внутренняя — внешние проектные роли тоже должны как-то знакомиться с принятыми в проекте способами работы), ибо сам Василий Пупкин без работы команды по методу onboarding пойдёт пить пиво и не будет заниматься проектом, забудет о нём. Команда помнит, что ей нужно работать над тем, что и внутренние и внешние проектные роли кем-то будут играться, и проводит работы для этого — а поскольку работы нельзя проводить «никаким способом», то она оговаривает этот метод/способ и отслеживает состояния предметов этого метода/способа в явном виде: моделирует происходящее в проекте как набор альф.

Что тут будет рабочими продуктами? В том-то и дело, что состояние альфы определяется/отслеживается (в стандарте OMG Essence термин «**свидетельствуется/evidence**») многими рабочими продуктами, то есть документами с описанием ролей, пока нет выбранного исполнителя роли, далее самим Васей Пупкиным. Рабочие продукты, по которым мы выявляем/свидетельствуем состояние альфы придётся описать, когда описываем состояния предмета метода — но обычно альфы соответствуют метаУ-модели (как в учебниках предметной области), а рабочие продукты существенно зависят от ситуации предприятия, это уже описание метаС-модели (как в стандартах предприятия).

Описание метода как операций, переводящих альфы/предметы метода из состояния в состояние по мере выполнения работ метода, хорошо дробится: метод приведения в какое-то состояние может быть полноценно расписан как имеющий свои какие-то альфы. Подробно всё это будет описано в руководстве по методологии, будут предложены формы табличного моделирования методов работы в проекте (рабочих процессов, практик, культуры работы и т.д. — число синонимов тут внушительное) и задействования самых разных информационных систем для этого моделирования.

Рабочие продукты/артефакты, свидетельствующие состояния альф проекта, во всех проектах разные, а методы работы и, соответственно, предметы этих методов работы с их состояниями (альфы) остаются более-менее одинаковыми, за мелкими вариациями — и это сильно экономит мышление. Внешняя роль «финансист заказчика», которая ответственна за «выделение и исполнение бюджета закупки целевой системы»::метод, будет практически во всех проектах, но вот в одном проекте своё состояние по отношению к выполнению этой роли будет менять Анна Павловна, в другом Вася Пупкин, в третьем Виталий Викторович, и все они будут работать на разных должностях в организациях заказчика. Мышление команды про них всех будет устроено одинаково, в функциональных терминах метода и его альф/«предмета метода», а не в терминах конструктивов — этих разных людей-исполнителей роли. Более того, даже если «описание роли» будет свидетельствоваться в одном проекте рабочим продуктом «инструкция финансиста заказчика»::документ, а в другом проекте таким описанием выступит «стандарт методов работы внешних рабочих ролей»::документ, то менять в мышлении надо будет не так много.

Как узнать, в каком состоянии «гвоздезабиватель» в проекте? Поглядеть на «рабочий продукт»/артефакт, выполняющий его роль: сначала это будет проектная документация, а потом (после воплощения камнем или микроскопом) и сам физический объект, воплощающий гвоздезабиватель. Альфа «гвоздезабиватель» может проходить в проекте много разных состояний:

- потребность определена, концепция использования понятна (рабочие продукты тут будут в проектной документации, и нужно будет указать конкретный документ, где мы можем это прочесть — «с воздуха, со слуха» мы ничему в проектах не верим! Не записано — значит, нету! Мозг — это плохая память для проекта, надо использовать более надёжные носители информации)
- быть с определённой функциональностью, но ещё не выбранным конструктивом (рабочий продукт тут по-прежнему — проектная документация, но это, возможно, уже какие-то другие документы),
- быть выбранным и ещё не закупленным (и это уже опять какие-то другие документы, например, электронные документы в закупочной информационной системе),
- быть закупленным, (рабочий продукт — сам гвоздезабиватель в виде молотка или микроскопа, лежит на складе)
- быть выданным в работу (рабочий продукт — сам гвоздезабиватель),
- быть проверенным в работе (рабочий продукт — сам гвоздезабиватель).

Вот это будет во внимании участников проекта, эти изменения состояния будут отслеживаться, а состояния альфы документироваться в рабочих продуктах проекта.

Альфы совершенно необязательно характеризуют изменения состояний именно целевой системы в ходе проекта. Нет, они могут свидетельствовать и об изменениях состояний систем окружения, равно как и состояний создателей в графе создателей. Системное мышление — оно применимо для всех систем проекта (но не в отрыве от проекта! Если вы ничего не хотите сделать, то есть у вас нет проекта, то и системное мышление ничем вам не поможет), целевая система тут просто система в начале коллективных координат мышления, главный объект внимания всех участников проекта. Но внимание участников проекта удерживается и на других системах, альфы помогают отмоделировать эти системы, их описания — чтобы отслеживать изменение их состояния в ходе проекта.

Примеры самых важных альф (по сути дела — это указание на kernel в терминах стандарта OMG Essence, мы тут предлагаем более универсальный вариант, чем в самом стандарте, и в нём меньшее число альф):

- воплощение системы (обычно это целевая система, подчёркивается её физичность, отслеживается степень готовности/воплощения её в физическом мире, при этом альфа — это хитрый онтологический объект, она допускает моделирование воплощения системы, существующего ещё в виде сырья или несобранных комплектующих, непроверенное, ещё не введённое в эксплуатацию, но в том числе и указание воплощения системы как уже эксплуатирующегося),
- описание системы (в жизни изменения описания системы можно будет отслеживать по разным рабочим продуктам-документам, включая записи в базах данных),

- функция/метод работ (way of working), функциональное описание поведения системы (если неодушевлённая система — назовут «функция», если интеллектуальный или коллективный создатель — «метод/способ работы»), это время эксплуатации системы.
- работы системы (это относится ко времени эксплуатации воплощённой системы, ибо работы системы создателя — это другая альфа! Состояние работ свидетельствуется какой-нибудь диаграммой Ганта или списком работ из информационной системы управления проектом, issue tracker).

С этими четырьмя видами альф работают в предприятии как проекте создания и развития каких-то систем (помним, что «сервисные предприятия» тоже имеют целевые системы, разве что они не их собственные) самые разные методы над самыми разными предметами методов:

- методы коммерции/продвижения (основной предмет этих методов: клиентура::«создаваемая и развиваемая система» — как совокупность клиентов предприятия),
- методы бизнеса (основной предмет методов: инвестура::«создаваемая и развиваемая система» — как совокупность инвесторов предприятия),
- и методы инженерии систем соответствующего уровня (не все из которых даже называются классическими «инженерными системами», например, «мастерство»),
- методы системного менеджмента (основной предмет метода — организация), в том числе как менеджмента организационного (инженерия команды, настраиваемой на какой-то метод работы), так и операционного (загрузка ресурсов на работу и проверка того, что работы реально ведутся).

Агенты в организации, исполняющие роли для всех требующихся в проекте методов работы, двигают своими работами эти основные альфы по их состояниям в ходе проекта.

Альфы как онтологический объект очень сложны, в том числе по причине того, что методы работают и с описаниями, привязка к физическому миру затруднена (даже документирование разных состояний предмета какого-то составного метода будет в разных документах, их сложно отслеживать), объекты в разных состояниях могут называться по-разному и даже считаться разными объектами (яйца, гусеницы, куколки, бабочки — это состояния всё той же бабочки, но альфа их может отслеживать ещё до яиц — как гаметы и зиготы родителей!). Онтологи тут хватаются за голову, но в целом такой неформальный подход к отслеживанию смены состояний важных объектов проекта вполне себя оправдывает. Альфы для того и нужны, чтобы моделировать сложный граф переходов состояний предметов сложных составных методов — включая моделирование состояний предметов метода такого сверхсложного метода, как «процесс инженерии» в целом или даже «процесс работы» предприятия в целом.

В стандарте OMG Essence альфы и свидетельствующие их рабочие продукты/артефакты изображаются разными значками: альфа похожа на греческую α (но это ничего не означает по факту, просто пиктограмма «альфы», обозначение «важного предмета метода, изменения состояний которого надо отслеживать в проекте»), а рабочий продукт/артефакт — на лист бумаги с загнутым уголком (лист бумаги — это стилизованное изображение документа, но в жизни это чаще всего электронный носитель информации).

Стандарт разрабатывался главным образом для разработчиков информационных систем, поэтому для альф-описаний (например, программного кода как описания программы) свидетельствующие/презентирующие/представляющие их рабочие продукты — это документы, в том числе электронные, но вот для других альф (скажем, команды как предметы методов менеджмента) — это могут быть вполне живые люди-исполнители ролей, «оргзвенья». Графическое обозначение их по стандарту и их будет изображать «листочками бумаги», указание на материальность/конструктивность, оторванность от «содержания на листочке», упор на логистический аспект при создании, абстрагирование от функции при использовании. Сами же альфы — это содержательно (а не ресурсно, как рабочие продукты) рассматриваемые предметы метода, проходящие какие-то состояния после выполнения работ по каким-то операциям/приёмам метода (более мелким методам. Мы можем говорить тут о функциональной декомпозиции для методов, понимая, что метод — это всё-таки поведение, а декомпозиция на части-целые делается с ролями как функциональными объектами, демонстрирующими поведение).

Альфы документируются в рабочих продуктах проекта, чтобы договариваться о состоянии важных объектов внимания в проекте, они отвечают на вопрос «в каком состоянии важные объекты проекта».

Основное умение умного человека — это умение определить набор предметов метода, состояние которых важно отслеживать в проекте и документировать эту модель ситуации. Этот набор предметов метода задаётся типами из знаний/объяснений/теорий/алгоритмов мета-модели (предметных областей) или даже мета-мета-модели (общей картины мира для многих предметных областей) ситуации проекта. Эти типы объектов надо или уже иметь «в голове», или уметь быстро «нагуглить», или отмоделировать как-то иначе — а затем найти соответствующие этим типам объекты в жизни, в ситуации проекта. А поскольку эти объекты как предметы метода в проекте проходят через различные состояния, то не просто найти такой объект, но суметь его найти в тех состояниях, которые предусмотрены методом.

В жизни нет ни одного слова из текста наших руководств (в руководствах — мета-мета-модель, там нет слов из предметных областей рабочих проектов!), то есть в тексте руководств нет ни одного слова из окружающей вас жизни, примеры тут не в счёт: в них никогда нет совпадения с вашими рабочими проектами! В руководствах главным образом говорится про предметы методов и операции смены их состояния (частные методы для этих состояний как темпоральных частей объектов полного метода — мы очень аккуратно стараемся говорить о «частях метода», ибо для методов понятия частей и целых в современной онтологии определены очень плохо), то есть говорится про альфы как важные для отслеживания в ходе проекта объекты мета-мета-модели.

В жизни, в рабочих проектах находятся специфичные для каждой конкретной компании в каждый момент времени рабочие продукты (они тоже могут меняться для какой-то компании с течением времени!), для которых довольно легко понять типы свидетельствуемых ими альф из метаУ-модели «учебника прикладного метода», классифицированные типами мета-мета-модели из фундаментальных методов мышления, описанных в наших руководствах. Если это рабочие продукты — документы, то понимаем типы метаУ-модели в описаниях из этих документов. Типы самих артефактов-документов как носителей информации тут не слишком важны, хотя для типографий, курьерских служб, архивов они могут быть тоже важными, а стандарты предприятия с изложением метаС-модели даже могут их предписывать именно с целью облегчения нахождения среди многочисленных документов проекта).

Это ключевое место для понимания системного мышления, ключевое для понимания способа его использования — как описанные в нашем руководстве методы мышления (в основе которых какая-то теория/знания/объяснения с предметами мышления) связаны с успешностью задействования методов инженерии как изменений реального мира: как успешность изменения мира зависит от моделирования важных черт этого мира, важных свойств этого мира, важных объектов в этом мире. Не надо тратить время на моделирование неважного, но зато удерживаем во внимании в течении всего хода проекта:

- Модель важных объектов мира (альфы, функциональные объекты, например «физическое тело» из учебника физики, меняющее свои состояния в ходе проекта «бросок мяча» или «падение гири»). Альфа говорит о потенциальных состояниях физического тела, когда его бросают, работа с ней позволяет обсудить траекторию. В учебнике физики всё про альфы, про физические тела — и ничего про мячи или гири.
- Реальный мир, данный нам в измерениях и объектах воздействия, конструктивные объекты, «рабочие продукты», воплощающие альфы — сам материальный мяч или гиря.

Вы знаете, что нужно найти объект какого-то типа мета-мета-модели (скажем, «нашу систему»), и ищете объект какого-то типа из предметной области (скажем, это будет «мастерство собранности»::«наша система»), и ещё нужно найти сам этот объект («Васино мастерство собранности»::«мастерство собранности»). Если не смогли найти, то вы что-то не поняли — додумывайте, выясняйте, экспериментируйте, проявляйте активную жизненную позицию. Системное мышление показывает вам, на что обратить внимание — даёт типы мета-мета-модели, альфы. Но дальше от вас ожидается творчество: догадки, действия (и мыслительные, и в физическом мире) по их обоснованию/assurance, то есть моделирование, в том числе обязательное документирование модели.

Есть и обратный вариант: вы находите, что в проекте важный объект X::«тип мета-модели, из domain какого-то метода в проекте». И вы присваиваете ему тип мета-мета-модели интеллект-стека фундаментальных методов мышления. Дальше к объекту X оказывается применим весь интеллект-стек фундаментальных методов мышления, вы ведёте дальше мышление не с неизвестным X, а с хорошо знакомым объектом из наших руководств (точно так же, как если выяснится, что вам нужно определить траекторию движения титановой детали сложной формы, вы отождествляете её с типом «физическое тело» из учебника механики, затем вычисляете траекторию её движения, хотя в учебнике механики ничего не сказано

про траектории движения замысловатых титановых деталей). Ключевое тут — вот эта операция классификации, присвоения типа из более абстрактных онтологических уровней по отношению «классификация».

После того, как вы узнали для X с известным типом мета-модели тип мета-мета-модели (для этого и нужен интеллект!) вы понимаете, что спрашивать дальше, над чем размышлять дальше, что надо делать дальше в физическом мире, какие возможны при этом новичковые ошибки и как их избежать.

В любом случае, вы деятельны/агентны: работаете (в том числе) с системными описаниями вашего проекта/моделями мира и с самим миром. Работаете — это применяете разные методы работы, меняя состояния предметов мира как предметов этих методов. Альфы нужны именно для отслеживания смены состояний в проекте, их нужно использовать для моделирования ситуации (конкретного набора состояния самых разных предметов самых разных методов, которые применяют в проекте самые разные роли), чтобы коллективно обсуждать ситуацию проекта, коллективно отслеживать ход проекта.

Вопросы. Рабочие продукты

Что из нижеперечисленного — документация?
+++ Отчёт «Результаты испытаний сепулькария от 13 мартобря 2026г.» на 14 страницах ### Да, это документ (указано число страниц! Это не просто содержание отчёта!).

+++ База данных с концепцией системы фазовыпрямителя ### Да, база данных с моделью — это документация, она где-то хранится на носителе информации.

--- Концепция использования фазовыпрямителя ### Нет, это описание. Такую альфу нужно специально документировать каким-то рабочим продуктом: документом-стандартом, или документом-спецификацией, или набором записей в базе данных, или электронным документом в отдельном файле.

--- Архитектура фазоуловителя ### Нет, сама архитектура — это описание принятых архитектурных решений, она должна быть отражена в документах (например, в виде ADR, architecture decision records), которые могут существовать как файлы, или как записи в базе данных, но сама она — не документ.

+++ Диаграмма Ганта для проекта создания фазозаместителя (доступна в формате MS Project) ### Да, модель работ в файле (указан формат файла) — документация системы создания фазозаместителя (диаграмма Ганта описывает работы проекта, то есть описывает поведение создателей фазозаместителя).

--- Работы проекта фазоудавителя ### Нет, это работы, а не документация. Документация должна представлять какое-то описание. Работы ничего не описывают.

--- Фазоудавитель с запасом концентрата к нему ### Нет, фазоудавитель ничего не описывает, это не документация. Документация должна представлять какое-то описание.

Первоклассник учит арифметику. Что у него в проекте рабочий продукт?
--- Выражение 2*2 ### Нет, просто выражение, без материального носителя — не документация, не рабочий продукт. Выражение абстрактно, у него нет объёма, занимаемого в физическом мире.

--- Натуральные числа ### Нет, просто числа, без материального носителя — не рабочий продукт/артефакт. Натуральные числа абстрактны, не занимают места в физическом мире.

+++ Счётные палочки, которыми он обозначил число 3 ### Да, это рабочий продукт. Счётные палочки занимают место в физическом мире и представляют описание (число 3 что-то описывает).

+++ Закаляканный листочек с ответами на обороте ### Да, это материальный предмет, документация ответов, рабочий продукт.

+++ Сборник задач ### Да, это материальный предмет, представляющий задачи, документация, то есть артефакт/рабочий продукт.

+++ Учебник ### Да, учебник занимает место в пространстве-времени, это физический объект/рабочий продукт/артефакт/документация.

--- Знания о 2*2 ### Нет, просто знания, без материального носителя — не документ, не артефакт.

+++ Часть головного мозга с запечатлённым в ней знанием 2*2 ### Да, это материальный предмет, рабочий продукт/артефакт.

--- Таблица умножения (значения операции умножения для однозначных чисел) ### Нет, просто таблица, без материального носителя — не рабочий продукт.

+++ Таблица умножения на плакате, который повесила в классе учительница Марья Ивановна ### Да, это материальный предмет как носитель описания, т.е. рабочий продукт/документация.

Инженера Константина попросили продемонстрировать потребности внешних проектных ролей. Он долго искал, но так и не нашёл документа с точно таким заголовком. Какие объяснения ниже будут верны?
--- Альфа «Потребности внешних проектных ролей» в проекте могла называться и по-другому (например, «потребности», или «Stakeholder needs»), надо было точнее узнать её название и смотреть документы по

этому названию. ### Нет, не верно. Названия альф и названия документов (рабочих продуктов) часто не слишком связаны, а ещё в документе может быть отражено состояние нескольких альф и речь будет идти только о части документа.

+++ Документы могут называться по-другому, смотреть нужно было по их содержанию, а не по заголовкам, и искать в содержании что-то про интересы/потребности/предпочтения внешних проектных ролей ### Да, именно так, заголовки у документов (артефактов, рабочих продуктов), содержащих потребности внешних проектных ролей, могут быть самые разные — разбираться нужно по сути, а не по использованным словам-терминам.

--- Потребности внешних проектных ролей находятся у исполнителей этих ролей, а у них он не искал ### Нет, интересы внешних проектных ролей вполне могут быть отражены в каких-то рабочих продуктах проекта (документах, в том числе электронных — например, базах данных), надо продолжать искать. Редко какой проект не имеет описания того, что хотят внешние проектные роли от его результатов. Но если таких документов и впрямь не нашлось, их надо разработать!

--- Потребностей внешних проектных ролей не бывает, бывают только требования к системе ### Нет, потребности внешних проектных ролей (stakeholder requirements, они же stakeholder needs) действительно отличаются от требований к системе (system requirements), при этом современная инженерия предпочитает вообще не упоминать требования к системе, это понятие устарело. Потребности внешних проектных ролей вполне могут быть отражены в каких-то рабочих продуктах выполняющей проект организации, надо искать. Например, можно поискать документы с концепцией использования, там наверняка что-то найдётся на эту тему. Это беда, если найти такой рабочий продукт не удаётся, тогда его надо обязательно разработать!

--- «Потребности внешних проектных ролей» — это альфа, а альфу можно только обсуждать, продемонстрировать её состояние нельзя, это же описание ### Нет, состояния альф отражаются в документах/артефактах/рабочих продуктах, и эти документы как раз демонстрируют их состояние. Потребности внешних проектных ролей (они же — интересы/предпочтения внешних проектных ролей к надсистеме) вполне могут быть отражены в каких-то рабочих продуктах реализующей проект организации, надо искать соответствующие документы, базы данных, переписку, стандарты и т.д., наиболее вероятно их найти где-то в документах по концепции использования. Их и нужно будет демонстрировать, но документы необязательно будут называться так же, как альфы, состояние которых они отражают.

Инженера Вячеслава попросили продемонстрировать потребности внешних проектных ролей. Он принёс два десятка самых разных

документов с самым разным содержанием, и начал зачитывать оттуда какие-то куски, утверждая, что это и есть запрошенное. Прав ли он?
+++ Да, так обычно и бывает ### Да, именно так, потребности внешних проектных ролей могут быть разбросаны по самым разным документам.

--- Нет, одной альфе обычно соответствует один документ ### Нет, этого достичь ещё никому не удавалось, альфы и документы (базы данных и т.д.) связаны отнюдь не как 1:1.

--- Ничего продемонстрировать нельзя, так как потребности — это альфа, их можно только обсуждать, и думать о них, но нет возможности их демонстрировать ### Нет, это только формально верный ответ. Но демонстрировать состояние альф легко — это делается по отражающим их состояние документам. Потребности вполне могут быть отражены в документах, и такой демонстрацией и был занят инженер Вячеслав.

--- Потребности должны демонстрировать сами исполнители внешних проектных ролей ### Нет, потребности внешних проектных ролей должны быть выявлены и документированы, без их понимания нельзя сделать концепцию использования, без неё нельзя сделать концепцию системы. А уж эти документы потом может демонстрировать кто угодно даже в отсутствии исполнителей внешних проектных ролей, в этом и суть документирования.

Менеджер Ирина в ответ на предложение продемонстрировать состояние альфы «Работа» показала содержимое issue tracker (софта для управления задачами, стоящими перед инженерами). Это был правильный ответ на предложение?
--- Нет, нужно было показать работающих людей ### Нет, по виду людей сделать заключение о состоянии альфы «Работа» затруднительно. Человек, выполнивший работу, и не выполнивший работу, и даже выполняющий работу, выглядит одинаково — но состояние альфы «Работа» будет разное.

--- Нет, ведь содержимое issue tracker — не документ ### Нет, содержимое issue tracker (база данных) — это вполне носитель информации для описания, т.е. документ, и этот документ представляет/описывает работы.

+++ Да, всё ОК ### Да, issue tracker документирует состояние альфы «Работа», это был правильный ответ на предложение.

Яблоки из жизни, яблоки из задачи

В реальном мире мы видим конкретные физические объекты, с которыми мы имеем дело:

- Если речь идёт о задаче изобретения, то это физический объект (у которого могут быть ипостаси роли, конструктива, места и т.д.). Функциональные объекты существуют только тогда, когда они воплощены конструктивом в

необходимом месте пространства-времени.

- Если речь идёт об отслеживании состояний предмета метода, то это «рабочие продукты»/артефакты (физические объекты, но часть из этих объектов может быть документацией, и тогда нас будет интересовать не сам физический объект, а описание из него). Состояния альфы как предмета метода мы можем отследить только по рабочим продуктам, а не «умозрительно».

Физические объекты (в том числе рабочие продукты) мы можем пощупать, пнуть ногой, указать на них пальцем (условным пальцем, если они слишком маленькие), погрузить в тачку (если они не слишком большие). Они имеют место в пространстве и протяжённы во времени (в какой-то момент начинают существовать, а в какой-то момент исчезают).

В реальном мире мы видим конструктивы или артефакты «яблоки» и действия с ними как предметами разных методов, например, «съесть яблоки», «сосчитать яблоки». Яблоки тем самым представляются двумя разными типами объектов из двух разных теорий/знаний/онтологий/дисциплин/моделей/объяснений/алгоритмов двух разных методов:

- Тип «пища» из метода «съесть» (пищу::«предмет метода» едят::«способ/метод работы»).
- Тип «счётный объект» из метода «сосчитать объекты» («счётный объект»::«предмет метода» сосчитать::метод).

Если вы не знаете метод работы, то вы не знаете тип объекта (это будет тип из неизвестной вам теории/онтологии/объяснений неизвестного метода). Если не догадались о типе объекта как типе предмета какого-то метода, то вам будет непонятно, что и как делать с этим объектом — это будет проблема, «не знаю, что делать в этой ситуации: какие объекты брать во внимание и какие с ними проводить работы, чтобы получить желаемое».

Это и есть связь жизни и руководства/регламента/«описания рабочего процесса» или учебника: вы должны сообразить тут, какого типа из мета-модели (предметной области) и мета-мета-модели (фундаментальных дисциплин: физики, математики, логики, в том числе типов объектов системного подхода), задаваемых учебниками, встретился объект в жизни. А если метод говорит, что объект какого-то типа должен быть — то надо активно искать этот объект (а иногда и создать, если найти не удаётся) и дальше приводить его в состояние, диктуемое теорией метода, теми действиями, которые задаёт алгоритм метода, то есть дальше работать по методу. Помним: метод выполняют, а дисциплина/теория только описывает метод, так что ещё нужно догадаться, что там за метод и какой его инструментарий скрывается за дисциплиной, какое мастерство будет требоваться — и дальше действовать!

Тут нужно привести байку про яблоки из задачи и из жизни, например, в таком варианте[392]:

> пришла в ... школу учительница, которая начиталась работ о дидактической функции наглядных пособий и считала, что надо учить на

[392] http://yandex.ru/yandsearch?text=%D1%8F%D0%B1%D0%BB%D0%BE%D0%BA%D0%B8+%2F%2B1+%D0%B8%D0%B7+%2F%2B1+%D0%B7%D0%B0%D0%B4%D0%B0%D1%87%D0%B8&lr=213

> наглядных пособиях. А проходили они в этот момент задачку на сложение: «3+5». И она принесла три яблока и ещё пять яблок, выложила их на стол и говорит: «Дети, вот вы видите здесь — раз–два–три — три яблока, а здесь вот — раз–два–три–четыре–пять — пять яблок. Вот я их соединяю, сколько получится всего яблок?» Дети пялятся на яблоки, слюни у них текут, но задачи не понимают. Второй день проходит, третий — рекорд: в таком классе обычно за день это проходили. Она приходит в учительскую, жалуется, что вот–де она применяет новые методы, наглядно все, а результата нет. И вот на пятый день с задней парты тянется рука, и ученик говорит: «Мэм, я теперь понял: эти яблоки, которые вы выложили на стол, не настоящие — это яблоки из задачи». — «Да, а что?» — «Ну тогда, мэм, совсем другое дело». И с этого момента, когда класс понял, что это не настоящие яблоки, а яблоки из задачи, все моментально пошло. Почему? Когда вы кладёте реальные яблоки — что с ними надо делать? Их надо есть. А чтобы считать, нужны рисуночки.

Вот ещё про то же самое[393]:

> – Мы займёмся арифметикой... У вас в кармане два яблока...
>
> Буратино хитро подмигнул:
>
> — Врёте, ни одного...
>
> — Я говорю, — терпеливо повторила девочка, — предположим, что у вас в кармане два яблока. Некто взял у вас одно яблоко. Сколько у вас осталось яблок?
>
> — Два.
>
> — Подумайте хорошенько.
>
> Буратино сморщился, — так здорово подумал.
>
> — Два...
>
> — Почему?
>
> — Я же не отдам Некту яблоко, хоть он дерись!
>
> — У вас нет никаких способностей к математике, — с огорчением сказала девочка.

Про физические тела из учебника физики (дисциплины/теории/мышления физики, документированной в учебнике) мы знаем, что при наличии силы тяжести они летят::двигаются::метод по параболе. Ускорения и масса — это характеристики физических тел как предметов методов/способов движения. Если кинуть камень и не суметь соотнести его с «физическим телом» как предметом метода «механическое движение», то нельзя сказать, что он полетит по параболе! Нет таких учебников, в которых описываются полёты::«механическое движение»::метод именно камней как предметов метода! Ричард Фейнман в своих заметках по преподаванию физики ровно об этом сожалел: по всему миру ученики физики не могут сопоставить свои отличные знания из учебника явлениям

[393] https://ru.wikipedia.org/wiki/Золотой_ключик,_или_Приключения_Буратино

окружающего физического мира[394].

Артефактов, которыми реализуются типы предметов методов (альфы) «из учебника», бесчисленное количество, про каждый из них учебники не напишешь! Поэтому учебники пишут про альфы как типовые предметы метода: короткий текст из учебника можно поэтому применить к тысячам и тысячам ситуаций в жизни, учебники пишут на некотором уровне абстракции, в них не описываются конкретные объекты и слишком узкие типы.

Альфа — это не просто предмет метода, а только такой предмет метода, изменения которого важны для отслеживания в ходе проекта. Альфы нужны для управления вниманием к изменениям в проекте. Но рассуждение про предметы метода и то, что не так-то просто найти эти объекты в жизни, даже если они описаны в учебнике, полностью приложимо к альфам.

Альфа обычно не очень формально специфицирована. Онтологи/методологи, настаивающие на строгой и однозначной формализации мира, остаются недовольными тем, как определены альфы. Сами альфы появились так: команда разработчиков OMG Essence задалась вопросом: а что у нас присутствует в каждом проекте по разработке софта? Какие предметы метода изменяют свои состояния в каждом варианте метода разработки софта (software process)? И просмотрела в 2011-2013 годах порядка 250 конкретных проектов разработки софта, придя к предложенному набору 7 основных альф стандарта (kernel в терминах этого стандарта, этот набор «примера kernel» из стандарта мы перестали использовать в нашем руководстве, он безнадёжно устарел и весьма специфичен именно для софта). И там уж что нашли, то нашли: никакой дополнительной формализации, приведения к одному онтологическому типу для альф (сами понятия метода, альфы, рабочего продукта — language в терминах этого стандарта, объекты kernel типизированы объектами language) не делалось.

Задача мыслящего человека — сопоставить описание объектов метода из учебника как типа для играющих их в жизни предметов, рабочих продуктов как физических объектов в их различных ипостасях и описаний из документов как рабочих продуктов. Только после этого можно будет применить теоретическое знание из учебника к ситуации в жизни. Нет ничего практичней хорошей теории (знаний/объяснений/дисциплины/онтологии/алгоритма), если ты распознаёшь объекты с типами из этой теории в жизни, то есть если твоё внимание умеет для типов из теории выхватить какие-то соответствующие им объекты из окружающего пёстрого фона самых разных объектов в проекте.

Примером, когда неумение работать с типами сочетается с незнанием основ системного мышления, может быть дискуссия среди разработчиков подхода active inference/деятельного познания (в котором вы в познании меняете ваши модели мира и себя, но деятельно — меняя не только модели, но и себя и мир!). Там была предложена концепция Markov blanket как «граница системы», но «как-то странно» — странность была отмечена[395] и породила огромную дискуссию[396]. Если бы участники дискуссии были знакомы с современным фундаментальным

---

[394] http://www.abitura.com/modern_physics/Feynman1.html

[395] http://philsci-archive.pitt.edu/19726/

[396] https://www.reddit.com/r/PredictiveProcessing/comments/rycjcg/commentaries_on_the_emperors_new_markov_blankets/

мышлением, включая системное мышление, они бы быстро нашли сразу три источника путаницы (все они отмечены в нашем руководстве как важные для мышления):

- Путаница в том, является ли Markov blanket математическим/ментальным объектом или физическим (скажем, граница клетки имеется в виду геометрическая как математический объект, или это граница клетки в физическом мире, и тогда из чего она состоит и какой толщины?)
- Путаница между функциональным и конструктивным объектом. Markov blanket — это функциональный объект с каким-то поведением, или физический объект или группа, которые будет демонстрировать соответствующее поведение? Граница клетки с каким-то поведением — это неважно из чего она состоит, или чётко понимаем, что речь идёт о мембране клетки, или мембране «со всем, что на неё налипло изнутри и снаружи».
- Путаница между понятием **интерфейса** и **интерфейсного модуля.** Интерфейс — это граница между двумя конструктивами/модулями, обеспечивающая их сопряжение в ходе работы, а вот интерфейсный модуль — это конструктив, имеющий два интерфейса: один заданный и обсуждаемый вовне системы, но и второй «внутрь системы». Так интерфейс Bluetooth — это стык/граница между двумя системами, метод работы систем на этом стыке и предметы этого метода описываются стандартом интерфейса Bluetooth, а интерфейсный модуль Bluetooth — это какая-то микросхема, с одной стороны имеющая тот самый интерфейс Bluetooth, а с другой стороны интерфейс соединения с остальной электроникой внутри системы, какие-то ножки микросхемы, через которые идут сигналы и питание. И это ещё не полное описание, ибо мы не коснулись вопроса антенны как интерфейсного модуля: у неё с одной стороны интерфейс электрических сигналов высокой частоты стандарта Bluetooth, а с другой стороны интерфейс радиоволн Bluetooth — и эта антенна или входит в интерфейсный модуль Bluetooth, или не входит. Обсуждали живую клетку: Markov blanket там интерфейс (нулевые физические размеры, «междумордие»: то, что между модулями) или интерфейсный модуль? Ибо в разговоре это то граница клетки на внешней мембране, то мембрана клетки, имеющая и интерфейс во внешнюю среду, и второй интерфейс во внутреннюю среду.

Все эти дискуссии можно было бы сильно сократить, если использовать системное мышление: быстро договориться, что для обсуждения нужно не одно понятие/тип Markov blanket, а целый набор понятий/типов: описание системы, воплощение системы, функциональная часть, конструктивная часть, интерфейс, интерфейсный модуль с двумя интерфейсами. Использовать для всего этого одно понятие/тип Markov blanket — это надолго затормозить проект из-за тотального непонимания того, что же именно обсуждается!

Обычно занимающиеся по нашему руководству проходят следующие стадии при изучении системного мышления:

1. **Ничего не понимают, ибо неспособны соотнести наставления руководства с окружающим их миром (то есть типы объектов из руководства с объектами окружающего мира).** Действительно, в их инженерных, менеджерских, культурных проектах нет никаких альф как предметов метода, меняющих в ходе проекта свои состояния. А в руководстве нет описания каких-то рабочих продуктов, какие есть в их проектах: сочных

гроздей винограда, густого мазута, восторженной толпы и гневных заголовков в политических СМИ, информационных моделей здания и мастерства продвижения сотрудников службы маркетинга. Более того, каждый проект уникальный — в них ведь вообще нет ничего общего (они все работают с очень ограниченными видами систем, привязанными к определённым масштабам), а руководство одно и то же для этих самых разных проектов (безмасштабно и по возможности деантропоморфизировано в своих наставлениях)! И эти проекты не описаны в руководстве, примеры именно из них не приведены (**хотя в литературе, на которую ссылается руководство, этих примеров множество, некоторые из них даже по предметным областям ваших проектов! Вы смотрите эту литературу по ссылкам? Примеры там!**). С учебником физики на этой стадии точно так же: в ситуациях на кухне и производстве не обнаруживается никаких физических тел, и ни одна глава учебника физики не описывает эти ситуации! Возникают странные вопросы типа «а к каким ситуациям применимо описанное в учебнике» — в ожидании, что найдётся ситуация, в которой для системного мышления будет «целевая система», а для физики механического движения будет «физическое тело». Но это не «найдётся очевидная ситуация», а просто будет выполнена операция присваивания типа, как это говорилось в руководстве по рациональной работе. Дальше во всех руководствах просто описываются самые разные типы, которые надо уметь присваивать объектам окружающего мира (и ещё надо помнить, что это не брать любые объекты окружающего мира и присваивать им типы, нет, наоборот: брать типы из руководств и выявлять для них объекты окружающего мира, подходящие под типы). Это типы разных методов работы — фундаментальных методов мышления, методов описания систем, методов создания систем.

2. **Всё понимают про приводимые в руководстве и дополнительной литературе к руководствам примеры, а также про проекты одногруппников на стажировке, однокурсников в вузе и коллег по работе, но при этом ничего не понимают про свои собственные проекты.** Конечно, ибо чужие проекты — это «проекты из чужой задачи» (помним, «яблоки из задачи — их можно считать»), а свои проекты — это «реальные мои проекты» («яблоки из жизни», их нужно есть!). То есть с типами работаем на чужих проектах, ибо мы не знаем деталей этих проектов, «всё прозрачно, это всё как в учебнике». Если вы не знаете деталей проекта, не вовлечены в него, вы быстрее находите главное, задаёте о нём вопросы, склонны действовать по способу, указанному в руководстве. Вы собранны, ибо детали не отвлекают вашего внимания!

3. **Всё понимают и про свои проекты, и про проекты коллег. Но ничего из понимаемого не делают, ибо системное мышление изучается не для того, чтобы его применять (то есть составлять и заполнять таблички с важными для отслеживания в проекте объектами), а «для самообразования и развития», «для сдачи зачёта» и т.д.** Применять знания мешают начальники, текучка, лень, отсутствие единомышленников, помогать применять знания некому. Это верно и для физики. Задачку на трение покоя из задачника по физике решат, а почему так трудно сдвинуть с места стол, когда его двигаешь по кухне — загадка! Поздравляю, вы аналитик, а не деятель! Квалификационная степень

«специалист».

4. **Следуют наставлениям руководства в своих проектах, ибо так работать оказывается качественней, легче и в конечном итоге быстрее.** Очень часто это происходит только через год-два после знакомства с руководством, после приобретения некоторой беглости в использовании понятий системного подхода. В первом разделе нашего руководства об этом было сказано подробно. С учебником физики (и любыми другими учебниками) происходит примерно то же самое: требуется время, чтобы добиться мастерства в методе. Конечно, речь не идёт о «правиле десяти тысяч часов» на достижение мастерства[397], но речь обычно для серьёзных методов идёт о годах осознанной практики (deliberate practice), а не о паре месяцев. Наш опыт показывает, что для овладения системным мышлением достаточно пары лет, но за эту пару лет нужно ещё поднять свой опыт и в других фундаментальных методах мышления (поставить беглую работу «машинки типов», научиться удерживать себя в рациональном рассуждении, поднять уровень собранности и т.д.). Тут как в спорте: за три месяца тренировок результаты обычно не будут видны, но через пару лет тренировок будут серьёзные отличия от «людей с улицы», через десяток лет тренировок — можно думать о чемпионстве. Это степень квалификации «практик».
5. **Следуют наставлениям руководства, чтобы организовать деятельность других людей в масштабах организации или даже сообщества.** Подробней об этом — в руководстве по системному менеджменту. Тут надо не просто организовать работу по какому-то методу себя, любимого, но организовать других людей на работу по методу. Квалификация тут — «мастер» для организации работы людей в масштабах предприятия и «реформатор» для организации работы людей в масштабах сообществ.

Моделирование: классификация типов мета-модели типами мета-мета-модели

Опишите ваш проект небольшим (до 10Кзнаков) текстовым эссе, аннотировав (через **::**) объекты и отношения «из жизни» (в терминах типов метаУ-модели или метаС-модели) в тексте эссе типами из нашего руководства (то есть типами «целевая система», «надсистема», «система создания», «функция», «роль» и т.д.). Постарайтесь использовать не менее десятка разных типов. Мы даём это задание как задание по моделированию, а не как задание на пост в блоге — это не столько «мышление письмом», сколько «мышление моделированием».

Описание проекта

Альфы

Воплощение системы, описание системы, создатель — это всё примеры **альф**,

[397] https://vc.ru/flood/17116-10000-rule-not-work

«яблок из учебников и задачников», предметов метода из набора типов системного подхода, мышление о проекте создания систем проводится с использованием объектов именно этих типов. Когда вы пишете, или говорите, или видите «Боинг 737» — то как системный мыслитель вы представляете себе некоторый класс самолётов, которые системы «Боинг 737»::самолёт::«воплощение системы», а воплощение системы это один из предметов метода разработки. Воплощение системы материально, оно будет как альфа в ходе проекта проходить состояния от «в сырье» до «полусобрано» через «собрано» и «испытано» к «эксплуатируется», «модернизировано», «выведено из эксплуатации» (обратите внимание, что «эксплуатируется» тут дано в настоящем времени, а остальное — через прохождение каких-то операций в прошедшем времени). В речи проявится только «Боинг 737», но в мышлении и рассуждениях у системного мыслителя будет вот эта раскладка по типам (поэтому и «воплощение системы полусобрано», а не «Боинг 737 полусобран»). Самолёт «в сырье» совсем не похож на самолёт в небе, язык разговора самолётостроителей совсем не похож на язык разговора пилотов, но альфа позволяет удержать внимание на самолёте во время всего его воплощения (опять замечаем: «воплощение системы» в этом абзаце было двух разных типов — как физический объект самолёта и как процесс изготовления самолёта в «металле». Вы это заметили?).

Рабочие продукты/артефакты, такие как документация различных моделей целевой системы и надсистемы, систем создания и их работ на бумаге или в базах данных, исполнители проектных ролей, материальное воплощение системы — это «яблоки, которые мы едим», их можно легко найти в проектах, ткнуть в них пальцем (указать место в пространстве). Скажем, мы хотим отследить состояние альфы «описание системы» для «Боинг 747». Как это сделать? По рабочим продуктам: описание системы содержится в различных базах данных, и в этих базах данных с информационной моделью самолёта можно узнать, например, приняты ли какие-то проектные решения или нет, имеем ли мы описание с детальностью технико-экономического проекта, эскизного проекта, рабочего проекта. Это удобно: альфу рассматриваем независимо от того, в какой базе данных отражено описание самолёта, каким шрифтом это описание выводится на экран, диаграммы в этом описании или таблицы, или даже просто тексты. Но рассмотрев рабочие продукты мы делаем вывод о том, в каком состоянии описание системы, насколько оно продвинулось от «описания нет» через «описание с основными архитектурными решениями готово» и «описание достаточно для изготовления» до «описание используется при эксплуатации». Все эти состояния условны, дальше будут детализированы через состояния подальф — в каждом проекте определяют свой набор альф и адаптируют его к ситуации проекта.

Подробней об альфах и о том, как моделировать продвижение проекта при помощи альф, будет говориться в руководстве по методологии.

Многие альфы проекта обычно хорошо описаны в учебниках предметной области проекта, относятся к метаУ-модели, а часть из них ещё и детализирована в регламентах или рабочих стандартах предприятия, относится к метаС-модели (и там в регламентах и рабочих стандартах часто ещё и говорится, по каким рабочим продуктам надо отслеживать изменение состояний этих альф).

Но самые общие альфы как самые общие предметы самых общих методов работы — это объекты мета-мета-модели, они даются нам в знаниях/теориях/объяснениях/алгоритмах фундаментальных методов мышления,

принципиально оторванных от предметных областей конкретных проектов. Именно поэтому дисциплины интеллект-стека фундаментальных методов мышления (физики, математики, онтологии, логики, рациональности и т.д.) называют трансдисциплинами: они «транс-», по ту сторону от предметных прикладных дисциплин, они дают нам абстрактные типы мета-мета-модели, а не более конкретные прикладные типы мета-модели.

Как сопоставить друг с другом объекты разных уровней мета-моделирования (разных онтологических уровней) — это главная трудность в освоении не только системной инженерии, но и любого другого метода работы, описывающего окружающий мир и его закономерности: как объекты каких-то описаний (например, любой теории/объяснений, которые обычно даны на довольно высоком уровне абстракции) совмещать с конкретными физическими объектами или даже объектами описаний с более низким уровнем абстракции.

Это как раз то, что физиков и инженеров отличает от математиков/чистых логиков: физиков волнует, чему в реальном мире соответствуют их формулы, а математиков/чистых логиков не волнует. Физики озабочены тем, чтобы аннотировать типами математики типы физических объектов (делать гипотезы о таких соответствиях, проводить эксперименты с измерениями и оценивать, насколько поведение абстрактных объектов отличается от измеренного поведения реальных физических объектов), математики главным образом описывают описания (хотя уже и идут разговоры про экспериментальную математику, ибо вы сразу попадаете в физический мир и проведение экспериментов, если на математические построения вам нужны ресурсы самого математика как агента, в своём мышлении моделирующего математические/абстрактные объекты).

Если в физическом мире есть объект «физическое пространство», то это означает, что в мире математических объектов есть объект «математическое пространство» и физик постулирует, что он будет описывать какие-то свойства реальности, которые он назовёт «физическим пространством» (а ещё в физике есть фазовые пространства и самые разные другие пространства), используя объект математического пространства, который ведёт себя в рассуждениях так, как ведёт себя в эксперименте (в измерениях в реальном физическом мире) физическое пространство. Это всё обсуждает семантика, работа с типами, в ней надо один раз разобраться, иначе будете путаться, когда физик говорит про «пространство» из физического мира, а когда «пространство» из математического мира.

Один из студентов МФТИ получил два балла на экзамене по системному мышлению за то, что не мог определиться: «товары к отгрузке» это лежащие на полке склада физические товары, или это список, который отображён где-то в базе данных его компьютера. Один раз у него в разговоре это были товары, а один раз — описание товаров, которое он называл товарами, ибо он главным образом писал программу с описанием товаров, а самих товаров в глаза не видел. И дальше эти же товары ещё нужно было называть целевой системой — и тут же оказывалось, что в компьютере у этих «якобы товаров» (на самом деле — описании товаров табличками в базе данных) совсем другое окружение, нежели у реальных товаров! Не будьте таким студентом!

Работайте с типами: описания идут на многих уровнях, но это описания! А физический мир — это физический мир, и там уровни не абстракции (по отношению классификации), а системные уровни (по отношению

композиции)! Отличайте отношение композиции («часть-целое») для физических объектов и «классификации» (класс-экземпляр) для описаний!

Когда вы работаете с альфами, отражающими состояние рабочих продуктов, то вам надо отличать альфы от рабочих продуктов. Так, альфа «инженерная команда» в жизни представлена агентами как участниками команды, а узнаём о составе команды и принятых решениях о том, как команда организована, из каких-то описаний (приказов о формировании команды, протоколов совещаний, распоряжениях о назначении распределения ролей в команде) — они будут свидетельствовать прохождение командой различных состояний.

Альфы (alphas) — это объекты, по которым мы судим о продвижении/progress («как много мы уже сделали?») и здоровье/health («всё ли в проекте идёт хорошо?») проекта, отслеживая изменение состояний каких-то предметов метода.

Альфы — это абстракция того же сорта, какого «физическое тело» является абстракцией реальных физических объектов. Это физическое тело имеет массу, а геометрическая точка имеет координаты. Но мы отождествляем физические тела и геометрические точки как идеальные/абстрактные объекты с реальными/физическими объектами. Мы «склеиваем» в мышлении идеальные и реальные объекты, реальные объекты наследуют характеристики идеальных объектов — о реальных объектах мы начинаем думать так, как описано в учебниках про абстрактные объекты. Поэтому у нас и у гири, и у атомной станции есть масса, и у пёрышка, и у самолёта тоже есть масса — мы умеем (**после довольно долгой тренировки по отождествлению физических объектов из жизни и абстрактных объектов из учебника!**) понять, что это физические тела и перенести характеристику «масса» этого физического тела на гирю, атомную станцию, пёрышко и самолёт. Если вы узнали, что в проекте основной альфой будут работы системы (вот прямо таки реальное задействование системы, которая что-то там меняет в своём окружении, производит работу/функционирование), то мы можем отслеживать что там происходит с работой: «работы системы ещё не было», «работа на тестовых данных прошла», «работа идёт на реальных данных», «работа прервана», «работа закончена» (и там ещё в графе состояний цикл по «работа идёт на реальных данных» и «работа прервана»). Зачем нужна альфа работы? Чтобы отслеживать — система просто создана, или создана и выполняет уже свою функцию. Работает или не работает система — это важно, за этим надо следить. Вот вы как инженер-менеджер освоили системное мышление (мы близки к концу руководства!), у вас есть уже какое-то мастерство. Работает ли ваше мастерство? А ещё вы член десятка самых разных команд — работают ли они, или просто созданы, а работа стоит как вкопанная?

Об экземплярах альф в проекте принято говорить так, как будто они вполне реальны и существуют в мире, несмотря на все абстракции — об этих альфах как мыслительных объектах мы судим по артефактам/рабочим продуктам.

Какая концепция использования автоматизированной зубочистки в нашем конкретном проекте, идущем прямо сейчас, 28 мартобря 2028 года? Я могу судить о том, что нужно внешним проектным ролям и как вписана зубочистка в её окружение (то есть судить о концепции использования, это же описание, абстрактный объект, да ещё и меняющий своё состояние в ходе проекта) по имеющейся документации концепции использования. Это могут быть несколько

разных документов с частями этой модели (частями концепции использования одной из моделей автоматизированной зубочистки), но они также могут содержать и многое другое, что мне сейчас не нужно. Концепция использования — одна из альф, документация с концепцией использования — набор артефактов/рабочих продуктов. Какие у нас внешние проектные роли только представлены агентами в проекте на сегодня, а какие внешние роли представлены агентами, которые уже сотрудничают с командой? Я могу посмотреть на реальных людей-исполнителей (или даже не людей, а роботов), играющих сегодня в проекте какую-то роль — и судить о том, представлены ли они как-то в проекте, и сотрудничают ли уже (например, настроены, если это роботы), или команде для этого нужно ещё поработать (или даже ещё обсудить, считает ли команда AI-робота членом команды, или считает его просто чьим-то инструментом).

Альфы проекта в их совокупности дают «компактное описание мира»/теорию/дисциплину/онтику/знания/модель/алгоритм/объяснение, удобные для описания продвижений в проекте. Альфы изучаются методологией, это предметы методов прежде всего — ведь проект это просто совокупность работ по принятому в проекте инженерному процессу/методу (то, что есть и иные варианты того, что обозначает термин «проект», мы уже обсуждали. Но классическое понимание — это совокупность работ по созданию какой-то системы, то есть совокупность работ по всем методам создания какой-то системы, а в последнее время с учётом третьего поколения системного мышления — совокупность работ по созданию и развитию какой-то системы). Концепты методологии (метод, предмет метода, альфа как самый важные предмет метода, который нужен для отслеживания изменений в проекте, работы по методу) служат для привлечения внимания команды проекта к самому важному в проекте, изменений чего никак нельзя игнорировать, выделяют это важное из пёстрого фона многочисленных неважных для того или иного предмета интереса деталей.

Вся эта работа с абстракциями нужна, чтобы иметь возможность повторно использовать известные нам способы рассуждений и решения задач для самых разных объектов. Так, мы думаем о «физическом теле» и «геометрических точках» единообразно, «как в учебниках физики и геометрии», а применимо это мышление к самым разным «реальным объектам вокруг нас», от коробка спичек до галактик.

В этой экономии мышления (учимся думать один раз, затем похоже думаем в самых разных ситуациях) и заключается смысл разделения альф и артефактов/продуктов, только речь идёт не о физике и математике, а о самых разных проектах.

Например, учимся думать о концепции использования как разрабатываемой модели один раз, а применяем потом это мышление к определению состояния разработки для конкретной бумажной или электронной документации, которую можно найти на производстве «в реале» в самых разных проектах, бессчётное количество раз:

- требованиям (если это старая инженерия!), с рабочим продуктом «спецификация требований» (в современной разработке такого нет, говорят о разных гипотезах по нужным характеристикам системы и её поведению, при этом стараются их ещё и не собирать в одном рабочем продукте, это ведь быстро становится «бутылочным горлышком» в разработке, поэтому функциональные описания системы стараются сегодня не складывать в одном рабочем продукте)
- сценариям использования, а рабочими продуктами тут будут какие-то файлы

с этими сценариями, изложенным на специальных для них языках (если они будут использованы) для каких-то моделеров,
- описаниям из текстов стандартов, а сами файлы с текстами стандартов тут будут рабочими продуктами
- user stories, рабочие продукты — карточки с user stories,
- пожелания (даже не потребности, а просто «хотелки») внешних проектных ролей, а рабочими продуктами могут быть, например, записи этих «хотелок» в базе данных или файлы анкет с этими «хотелками»,
- … и т.д. Артефактов, которые могут документировать концепцию использования и разные описания в её составе — тысячи их, а вот разобраться с тем, что это такое и зачем концепция использования нужна, при изучении системной инженерии нужно всего один раз.

Помним, что **системное мышление указывает на концепцию использования как необходимый объект внимания в проекте**, **методология учит, что изменение состояний концепции использования нужно будет отслеживать по альфам, а сама она будет изложена в самых разных рабочих продуктах/артефактах,** но вот **методам работы по разработке концепции использования** (дисциплине/знаниям/теории, дающим набор альф по выявлению внешних проектных ролей, их потребностей, выявлению важных событий в использовании, выявлению сценариев happy path и сценариев при обнаружении затруднений в использовании, а также использованию моделеров для документирования концепции использования, и т.д.) **учит системная инженерия**.

Несмотря на всю «идеальность» и «абстрактность», об альфах говорят как о вполне существующих в физическом мире, примерно так же, как физики используемые ими для моделирования реальности математические объекты считают вполне реальными (принимают их всерьёз, пока не будет найдено расхождение с экспериментом). Так что альфа «концепция использования» — это вполне реальный объект, у него есть какое-то состояние (сначала её нет, потом разработана начальная, потом она будет реализована созданной системой, потом продолжит непрерывно уточняться и реализовываться новыми и новыми версиями системы), хотя в жизни это будут самые разные файлы и записи в базах данных на самых разных компьютерах. Файлы тут ничто (форма, хотя она и важна для логистики в проекте, информация должна быть у тех кому она нужна, причём в той форме, в которой ей можно удобно воспользоваться), концепция использования — всё (содержание). Альфы — про содержание, рабочие продукты — про форму. В мышлении о методах работы будут присутствовать главным образом альфы, а работы будут выполняться с рабочими продуктами.

О состоянии Принца Гамлета мы судим, разглядывая выражения лица и вслушиваясь в интонации Васи Пупкина, играющего роль Принца Гамлета. А вот о состоянии концепции использования (как продвигается работа по методам разработки концепции использования) судим по рабочим продуктам-документам, в которых она описана. Возможны и более хитрые случаи: о состоянии физической системы (альфа «воплощение системы», например, «воплощение информационной системы выдачи зарплаты») на заседании правления судим по рабочему продукту «отчёт о состоянии разработки системы выдачи зарплаты», в котором описано состояние системы, а не разглядывая саму систему, которая может существовать ещё в виде исходных кодов (конечно, тут мы не говорим, что «воплощение

информационной системы существует сначала в виде сырья», но вот «целевые сервера для информационной системы определены» тут вполне нормальный ход на обсуждение воплощения).

Так, можно определить альфу функционального объекта «моя любимая игрушка» — и, хотя в детстве у меня это был нарисованный на ватмане пульт управления космическим кораблём, а сегодня это мой ноутбук, я могу говорить про прохождение «моей любимой игрушкой» состояний «полюблена», «разлюблена», «поломана», «в игре», «заброшена» и т.д. — независимо от того, чем реализована эта игрушка прямо сейчас. Более того, наблюдая саму игрушку, мы не сможем сказать её состояние — наблюдать-то нужно за мной и моим состоянием! Рабочий продукт для «моей любимой игрушки», по которому можно определить её состояние — это я!

Если в мышлении и разговоре появляется «требование» (у практикующих старые варианты системной инженерии это очень часто, системный мыслитель тут просто улыбается и понимает, что происходит и как смягчить вредные последствия использования такой альфы) — то неважно, рабочим продуктом тут будет пункт протокола совещания с представителями заказчика, или запись в базе данных системы управления требованиями, или фрагмент диаграммы какой-то модели требований, или требования из электронного письма заказчика. В мышлении это будет распознано как что-то, относящееся к альфе описания системы, подальфе концепции использования (заодно системный мыслитель сменит деонтическую модальность «долженствования» на доксическую «веры» — «требую» заменит на «высказываю догадку, что так будет хорошо, хотя могу и ошибаться») — и после этого становится понятно, что с ним делать, как думать о содержании этого «требования». Конечно, этой альфой «требование» будет описание целевой системы на внешней границе, которое надо будет увязывать с потребностями внешних проектных ролей. Сразу понятно, почему важно удерживать на нём внимание, но поскольку это ещё и альфа, то уместно будет спрашивать о состоянии разработки: что делали, что предстоит сделать, в каком состоянии концепция системы. Конечно, желательно убедиться, что в случае «требований» по возможности устранены недостатки этого старого понятия: нет убеждённости, что можно их разово «удовлетворить», ожидаются изменения в любой момент, нет «испорченного телефона» из множества аналитиков, не утеряно понимание, какие там были сценарии использования, которые привели к таким требованиям, и т.д.

Это «требование» будет обсуждаться как реальный (хотя и абстрактный, не имеющий места в пространстве-времени — это же описание!) объект, существующий в мире, имеющий своё состояние — и в этот момент неважно, что у этого «требования» есть ещё и какие-то особенности выражения на конкретном носителе информации, ровно как при счёте яблок неважно, что их ещё и едят. Мышление об альфах чуть сложней, чем мышление о физических объектах, но оно устроено очень похожим образом.

В стандарте OMG Essence[398] для ALPHA придумали аббревиатуру, Abstract-Level Progress Health Attribute, но эта аббревиатура про health attribute была для альфы подобрана задним числом (просто потому, что альфы нужны были для отслеживания их изменения в ходе проекта — если не изменяются, то проект «нездоров»), а слово alpha часто пишут поэтому маленькими буквами, чтобы не

[398] http://www.omg.org/spec/Essence/

акцентировать липовую «аббревиатурность».

Повторим: в стандарте OMG Essence есть две части: язык/language, который вводит понятия альфы, рабочего продукта, работы (и мы их используем), и ядро/kernel как предлагаемый стандартом начальный набор **основных/ядерных/kernel альф** (в kernel также предлагаются и какой-то набор стадий/этапов проекта, мы их тут не рассматриваем), чаще всего встречавших в софтовых проектах некоторое время назад. Предложенные стандартом в kernel альфы — проектные роли, возможность создания системы, требования, воплощение системы, команда, способ работы, работы. Стандарт активно разрабатывался где-то до 2015 года (последние косметические изменения там сделаны были в 2024 году, версия 2.0), поэтому одна из основных альф предложенного примера kernel в стандарте — требования. Но инженерия требований как один из частных методов общеинженерной работы оказалась не такой удачной идеей, когда перешли к непрерывной гибкой/agile разработке, поэтому требования перестали разрабатывать. Так что мы используем language из этого стандарта, но предлагаем другой набор основных альф (kernel), что прямо предусмотрено в стандарте (подробней об этом будет в руководстве по методологии). В нашем наборе альф для kernel — воплощение системы, описание системы, метод работы системы и работы системы в области интересов каждой из систем.

Альфы — это объекты, меняющиеся в ходе проекта. Альфы как предметы методов работы меняют свои состояния в ходе работ по методам. Альфы — это те важные объекты, изменения которых в проекте мы хотим знать, понимать, отслеживать, проводить, направлять, контролировать и для этого надёжно удерживать эти объекты во внимании команды. Наверняка вы это уже запомнили. Это означает, что если:

- Самолёт::система, то мы воображаем его сразу в операционном окружении в момент, когда он летит, как физический объект, прежде всего функциональный физический объект, но воплощённый конструктивными физическими объектами, которые все занимают какие-то места и сколько-то стоят. Это первое поколение системного мышления. **Первое поколение главным образом использует отношение композиции, системные уровни тут, разговор про подсистемы.**
- Самолёт::«альфа воплощения системы», то мы воображаем его сразу предметом метода разработки, проходящим какие-то состояния, например, «самолёт только-только задуман», «самолёт спроектирован», «изготовлен» и «самолёт в его операционном окружении в момент, когда он летит». Это предмет метода работы создателей в графе создателей, второе поколение системного мышления. **Второе поколение системного мышления главным образом использует отношение создания, системы как предметы методов работы создателей тут. Разговор про альфы и изменение состояний альф создателями системы тут.**
- Самолёт::«версия системы», то мы воображаем ряд феномов (воплощений системы, аналог фенома[399] в биологии — «фактическое описание взрослого организма»), соответствующих разным мемомам[400] (описаниям в проекте/design системы, аналог генома[401] в биологии — гены, описывающие

[399] https://ru.wikipedia.org/wiki/Феном
[400] https://en.wiktionary.org/wiki/memome
[401] https://ru.wikipedia.org/wiki/Геном

взрослый организм как проект/design). **Третье поколение системного мышления обсуждает эволюцию систем, в том числе дарвиновскую для живых систем (на основе геномов, которые хранятся в каждой клетке организма) и техно-эволюцию, поэтому основным отношением кроме отношения создания является отношение развития (между версиями системы, трассируемое, конечно, к версиям описания — как развитие видов организмов как феномов в ходе эволюции трассируется к развитию их геномов). Разговор про непрерывность разработки и версионирование — тут.**

Альфы удобны в том числе и потому, что если нам хочется отслеживать состояние не только физических объектов (чаще всего — систем в их различных ипостасях), но и описаний этих систем как мыслительных/абстрактных/идеальных объектов, документированных в самых разных рабочих продуктах/артефактах, то мы вполне можем это делать — все они являются предметами самых разных методов, работающих с самыми разными объектами самых разных типов. Повторим, что это плохо формализуется в современной классической онтологии, но нас философско-логические нюансы пока не интересуют.

Рабочий продукт/артефакт/исполнитель роли представляет/represent (или в OMG Essence говорят не термин из семантики «представление/representations» знаком понятия, а о свидетельствовании/evidence объектом модели состояния другого объекта) собой альфу в реальном мире. Это «яблоко из жизни», «его едят» — продукт в реальном мире определяется в физических границах, он материален. Продукты/артефакты — это документированные спецификации, документированные тесты, документированные диаграммы и какие-то документированные модели, базы данных, а также и физические объекты (люди-исполнители проектных ролей, оборудование, материалы, здания и сооружения). Рабочим продуктом могут быть и какие-то мероприятия (совещания, презентации, уборка рабочего места), которые можно представлять «вещественно» как совокупность участвующих в этом мероприятии взаимно изменяющихся объектов. Мероприятие как артефакт тоже может увеличивать детальность в своей подготовке — оно может быть готово «в основных чертах», «спланировано», «организаторы сами организованы», «люди приглашены», «площадки подготовлены», «проходит», «завершено», «уборка площадок закончена». Но нельзя объявлять системой процессы/методы. Помним: если мероприятие или перформанс с чётким временем начала и конца, в него можно ткнуть пальцем (в материальное место, где оно проходит) — моделировать можно, а вот если это что-то долгое-непрерывное типа «течение воды», или «человеческая жизнь», или «работа предприятия», то моделировать процесс как продукт/артефакт нельзя.

Одну альфу могут представлять несколько продуктов/артефактов, ибо у альфы в деятельности могут проявляться много различных свойств, требующих различных представлений/representations/«знак представляет понятие» в мире, а эти представления могут быть представлены/presented «материальный носитель представляет знак» в разных артефактах/рабочих продуктах (например, документах).

Скажем, концепция использования может быть представлена в трёх разных документах — фрагменте начального технического задания как приложения к договору (часто под названием «предмет работы»/scope of work), тексте учебника по прикладной области, где используется система подобного сорта (или стандарте,

где описываются подобные системы), а ещё в протоколе совещания, где обсуждались потребности внешних проектных ролей. О состоянии концепции использования (насколько продвинулись в её разработке, в чём там основные проблемы, что нужно с ней делать дальше) можно будет сказать, только рассмотрев все эти три документа — то есть состояние «концепции использования»::альфа (состояние концепции использования как предмета метода разработки концепции использования в ходе работ по этому методу) свидетельствуется в данном случае тремя документами. А какие документы свидетельствуют состояние концепции использования в вашем проекте?

Но и несколько альф могут быть представлены в одном артефакте/продукте. В документе «Техническое задание» про концепцию использования (и то, до сих пор во многих организациях тут будет не «концепция использования», а «требования») говорится только в одном из разделов вперемешку с желаемыми заказчиком частями концепции системы и архитектуры, а в других разделах прописывают работы (полностью же документ обычно называется «техническое задание на работы» — его главная задача указать поручаемые контрактору работы, а не описывать методы работы или описывать системы как результаты работ).

Эта связь продуктов и их альф обычно определяется в рамках какого-то метода работы. Рабочие продукты и инструментарий метода ситуативны для каждой конкретной организации и даже конкретного проекта, а вот альфы более стабильны. Скажем, в методе проектного управления теория проектного управления (в науке — operations research[402]) одна и та же, а вот инструментарий (софт проектного управления, моделер для проектов) в разных проектах может быть очень разным. Метод/способ/практика работы определяется по его теории/дисциплине/объяснениям с их предметами, но в целом метод описывается не только теорией и дисциплиной работы с предметами прикладной предметной области, но и включает в себя описания работ с инструментарием. Инструментарий обычно уникален для проекта, он состоит из разной аппаратуры, оборудования, программно-аппаратных комплексов, которые часто надо ещё и разработать в рамках самого проекта, поэтому работы по одному и тому же методу могут сильно отличаться — рытьё котлована лопатами и экскаватором имеет один и тот же метод «рыть», предмет метода «котлован», альфа котлована будет отражать его состояния (начиная с «потребность в котловане осознана», дальше через «проект/design котлована готов», «инструментарий рытья выбран», «рытьё начато», «котлован вырыт»), но вот как это в мире будет выглядеть — отличаться для лопат и экскаватора будет сильно. А ещё котлованы роют взрывом.

Экономия мышления заключается в том, что часто в проекте даже не очень большой сложности состояние одной альфы отражается/представляется в десятках разных артефактов/продуктов (часто документах или базах данных, но помним и о просто физических объектах, и даже людях). Мышление/рассуждения тем самым ведутся только для одного объекта-альфы, но при разбирательстве с какими-то конкретными характеристиками ситуации (concerns) вытаскиваются отдельные индивидуальные продукты, в которых отражены именно эти характеристики (дано соответствующее описание/view, следующее методу описания/viewpoint, отражающему нужную важную характеристику/concern). Например, «насос готов к проведению пуско-наладочных работ?» — в то же время свидетельства готовности

[402] https://en.wikipedia.org/wiki/Operations_research

насоса::воплощение системы к пуско-наладочным работам может быть разбросано (кроме факта наличия самих рабочих продуктов, представляющих насос «в металле») по десяткам разных документов: актов сдачи работ различными подрядчиками, актов предварительных испытаний, писем контрагентов, записей в базах данных систем управления активами предприятия, сообщений о наличии расходных материалов и т.д.

Альфы (предметы метода) могут быть как входными для какого-то метода (включая метод инженерии/разработки в целом, «входными для проекта» как всех работ по методу инженерии/разработки), так и представлять собой результаты применения метода (включая быть результатами проекта в целом), или же быть полезными только команде, не являясь ни входными, ни результирующими для проекта.

Мы будем смотреть на какие-то артефакты, чтобы понимать, в каком состоянии находятся представляемые ими альфы — главные объекты нашего внимания в проекте. **Как альфа меняется, проходя по мере выполнения с ней работ метода разработки свои состояния, продукты/артефакты меняются в ходе проекта, проходя уровни детальности.** Код компьютерной программы может быть записан в виде идеи, в виде псевдокода, в виде кода с ошибками, в виде компилируемого без ошибок кода, в виде хорошо откомментированного кода — детальность нарастает, но это вовсе не означает, что альфа «код компьютерной программы» при этом приближается к готовности: вполне возможно, что не нужно было хорошо комментировать код, а нужно было даже поменять идею этого кода, документировать совсем другую идею! Подробней это различие состояний альф и уровня детальности рабочих продуктов разбирается в OMG Essence, мы не будем это разбирать в нашем руководстве. Работа с состояниями рабочих продуктов рассматривается в инженерии, мы же в системном мышлении и тем более методологии будем сосредоточены на альфах: на важнейших объектах внимания в проектах, изменение состояний которых говорит о ходе проекта.

Продукты/артефакты в ходе инженерного (предмет метода инженерии — киберфизическая система), менеджерского (предмет метода инженерии — организация), учебного (предмет метода инженерии — мастерство), культурного (предмет метода инженерии — сообщество) создаются, модифицируются, уничтожаются. Они вместе с их уровнями детальности наблюдаемы, в отличие от альф, о состояниях которых мы можем судить только «по приборам» — по рабочим продуктам. Но главное — это состояние альф!

Вася Пупкин нас интересует ровно в той мере, в которой он справляется с исполнением роли Принца Гамлета, по факту нас волнует вообще не он, а его мастерство исполнения роли. Техническое задание как рабочий продукт нас волнует ровно в той мере, в которой это ТЗ с каким-то уровнем детальности документирует начальную (она будет дорабатываться всё время!) концепцию использования и те работы, которые необходимо выполнить для получения MVP (и мы будем искать формулировки в ТЗ, что там предусмотрено с работами дальше, как там относятся к непрерывному развитию — даже если ничего об этом не написано, вряд ли обойдётся без какого-то развития системы, «выполнил и забыл» вряд ли получится). А вот концепция использования и работы важны в каждом проекте — и ТЗ нам важно ровно в той мере, в которой оно документирует концепцию использования и работы. Если мы не обнаружим в ТЗ нужного нам знания о концепции использования и работах (например, мы уже три месяца

занимаемся проектом, и концепция использования поменялась, равно как понимание, что там за работы), то мы будем смотреть не в ТЗ, а в другие документы.

Подальфы (sub-alpha) определяются по отношению к основной альфе как такие альфы, при продвижении состояний которых к конечному их состоянию в проекте мы можем считать, что этим самым как-то продвигается и их основная альфа. Это определение из стандарта OMG Essence не эквивалентно определению отношения часть-целое, но оно очень похоже. Если вы сделали ножку от стула (подальфа воплощения системы для стула как целевой системы), то тем самым и стул в целом стал «более сделанным». Подальфы тоже могут составлять иерархию по отношению «продвигает состояние». Обратите внимание, что «под-» тут коварная приставка, она не упоминает тип отношения в иерархии. Для подсистем — это «часть-целое», для альф — «продвижение состояния». А вот для методов это может означать что угодно: и какую-то темпоральную часть метода (метод стадии/этапа/фазы/шага работы), и вид метода (альтернативные способы какой-то работы: плавание как метод передвижения пловца::роль в воде может быть или брассом, или кролем, и весьма вероятно, что кто-то назовёт это не видом или вариантом метода, а под-методом).

Альфу системного описания можно представить проходящей состояния (помним, что эти состояния нужно адаптировать для разных проектов: для какого-нибудь музыкального, или учебного, или медицинского проекта системное мышление вполне применимо, но вот многие термины для состояний альфы системного описания в этих проектах лучше бы заменить на термины мета-модели из музыки, обучения, медицины, равно как уточнить сами состояния):

- Замышлено (ясно, каково будет описание системы)
- Непротиворечиво (описание системы создано и непротиворечиво)
- Используется для изготовления (описание используется для изготовления воплощения системы)
- Используется для инженерных обоснований воплощения (описание используется для проведения тестов и испытаний)
- Используется для эксплуатации (используется проектными ролями для эксплуатации воплощения системы)
- Используется для вывода из эксплуатации (используется для ликвидации и/или переработки воплощения системы)

Концепция использования — это подальфа описания системы (system definition): чем более готова концепция использования, тем более готово описание системы. Помним, что одна и та же концепция использования, одно и то же описание системы могут быть отображены на разных носителях разными методами, с использованием даже разных нотаций — какая-то степень готовности описаний означает их содержательную готовность, а не степень документированности! Степень документированности — это степень роста детальности в изготовлении рабочего продукта, а степень готовности концепции использования — это про альфу, предмет метода, про готовность содержания (насколько продвинулись в выявлении сценариев использования), независимо от оформления в рабочем продукте/документации.

Концепцию использования как подальфу системного описания можно представить себе проходящую состояния (эти состояния мы берём из альфы «требования» OMG Essence, англоязычные термины оттуда, но помним, что инженерии требований

больше нет, сегодня ограничиваются созданием концепции использования, в которой основным будет описание системы как «чёрного ящика» в формате разнообразных сценариев использования):

- Замышлена (conceived, внутренние проектные роли согласились в необходимости новой системы для удовлетворения потребностей внешних проектных ролей)
- Ограничена (bounded, основная функция/назначение/сервис новой системы понятна)
- Непротиворечива (coherent, концепция использования непротиворечиво описывает основные сценарии использования, в том числе существенные характеристики новой системы)
- Приемлема (acceptable, концепция использования описывает такую систему, которая приемлема как для внутренних, так и для внешних проектных ролей)
- Отвечена (addressed, воплощение системы удовлетворяет концепции использования настолько, что система принимается внешними проектными ролями в эксплуатацию)
- Удовлетворена (fulfilled, воплощение системы полностью отвечает концепции использования, внешние проектные роли считают систему успешной)

Если альфа концепции использования проходит по этим состояниям, то это как-то продвигает по своим состояниям и системное описание в целом (в системное описание входит ещё и архитектура как набор архитектурных решений, и проект/design системы, причём подальфа проекта/design проходит состояния концепции системы, технико-экономического проекта, эскизного проекта — и только потом рабочего проекта, по которому будет идти изготовление системы. Это в машиностроении, в других сферах деятельность, конечно, всё будет по-другому).

Речь необязательно идёт о всей системе, но речь может идти о какой-то фиче системы, или части концепции использования, описывающей более-менее автономную фичу. Но и эти концепции использования фич можно считать подальфами концепции использования, подальфы второго уровня — ибо концепция использования как подальфа описания системы иначе просто будет застревать на всё время проекта в состоянии «отвечена», это никак не будет характеризовать происходящее в проекте. В таких ситуациях и делают подальфы, в больших проектах на много уровней: когда что-то в проекте активно меняется, находится в разных состояниях, а в целом нельзя сказать, что именно происходит.

Архитектура — это тоже подальфа описания системы: чем более готова архитектура, тем более описана система. «Рабочка» (детальные модели/чертежи, а в случае софта это исходный код, готовый для компилирования) тут тоже подальфа описания системы.

Отношения альф и подальф определяются деятельностно (а не как отношения часть-целое), но мы думаем об этих отношениях похоже: если проектные роли при реализации своих предпочтений двигают подальфы по их состояниям, то они тем самым двигают альфу этой подальфы. Если проектные роли двигают всю альфу, то по факту они это делают, двигая различные подальфы этой альфы.

Системное описание

Системная документация/документация системы (system description) — это

документы/артефакты/рабочие продукты, реализующие альфу **системного описания/описания системы** (system definition). Описание системы в предыдущей фразе дано с типом альфа — это означает, что предлагается онтика, в которой создатели будут разрабатывать описание системы какими-то методами описания, оно будет предметом метода, будет проходить какие-то состояния, которые будут отслеживаться в ходе проекта, и будет представлено какими-то артефактами.

Если система есть, то она считается описанной: у неё существует концепция использования системы (альфа, не рабочие продукты!), существует её проект/design (который включает концепцию системы, архитектуру как набор архитектурных решений, а ещё инженерные обоснования, детали изготовления, описание метода/технологии изготовления и т.д.). Мы можем просто ещё не знать это описание, хотя оно «логически есть» — система-то в пространстве-времени, и время тут может быть в некотором будущем, а мы в настоящем, и поэтому просто ещё не подумали об этих описаниях и не документировали их. Альфы есть, но рабочих продуктов нет — поэтому нельзя ничего об этих альфах сказать. Но мы считаем, что они уже есть как предметы каких-то методов, они проходят какие-то состояния, но вот какое состояние у этих альф в проекте прямо сейчас — ничего сказать не можем, если нет артефактов, по которым мы эти состояния сможем узнать.

Или мы просто не нашли ещё исполнителей, делающих такие описания проектных ролей, и не догадались ещё задать им правильный вопрос — и они нам дадут недостающие части описания, они-то их знают, это только мы пока не знаем. В любом случае: есть система — есть концепция использования, есть концепция системы, есть её архитектура, возможно просто пока не выявленные/discovered (либо не найдены уже имеющиеся, либо их ещё не придумали и не документировали): типы мета-мета-модели мы уже знаем, осталось только получить сами объекты, а затем документировать эти описания.

Это не так, что «мы не знаем, есть они или нет». Они есть, вот же их типы — значит, описания должны быть, рано или поздно, внимание к ним как альфам (они будут менять свои состояния в ходе проекта) уже есть, внимание наводится уже известными вам типами! Надо только их выявить или даже спроектировать — предложить догадку о том, каковы они, согласовать догадки разных людей, попробовать изготовить по этим описаниям систему (тогда такие описания назовут проект/design).

Когда вы идёте в совсем новый проект, вы уже много знаете о том, за чем вам там надо следить, изменения состояний каких альф отслеживать, у вас не полная неизвестность! Вы уже знаете фундаментальную мета-мета-модель, а иногда и прикладную мета-модель (если будете заниматься чем-то в прикладной предметной области, практику которой вам уже довелось освоить). Поэтому что-то о мире вы уже знаете, но что-то (модель ситуации, безо всяких «мета») придётся узнать в самом проекте, да ещё и отслеживать смену состояний объектов быстро меняющегося проекта.

Если речь идёт об обуви, то вы уже знаете, что там будет подошва, каблук, стелька, форма называется «фасон» — достаточно одного слова «обувь», чтобы начать обсуждать объекты этих типов. Они там точно есть, даже если каблук там «без

каблука» (пробел — тоже знак!). Если вы идёте в проект, то вы знаете, что там создаётся система, есть создатель, отслеживать надо альфы, и для многочисленных систем проекта надо отслеживать как минимум их четыре (целевая система, описание системы, метод работы системы, работы системы) да ещё и бить на подальфы (про подальфы будет чуть позже).

Чтобы явить миру (выявить) описание системы, его нужно не только узнать (или даже предложить новое), но и записать на каком-то носителе, чтобы появилась системная документация — записать все эти модели (а иногда и тексты на естественном языке, и фотографии, которые не очень похожи на модели, тем не менее они тоже отражают только что-то важное в реальной системе, описывают что-то в ней, так что смело считаем их тоже моделями, разве что это менее формальные описания) в электронном виде как записи в базе данных. Представление на бумаге сегодня даже не рассматриваем, это уже прошлый век. Но когда говорят об альфе системного описания (system definition), то рабочий продукт системной документации (system description) по большому счёту неважен — это вопрос больше удобства работы, чем что-то принципиальное. Сейчас удобно работать с записями в базе данных (локальные представления на электронном носителе), потом, возможно, с какими-то нейросетями, отражающими эти описания (распределёнными представлениями на электронном носителе), а в прошлом веке работали с локальными представлениями на бумаге (например, бумажными чертежами механических систем).

Есть система — значит кто-то её выделил из окружающего мира для каких-то своих целей («система в глазах смотрящего»), согласовал это выделение системы из мира с другими создателями в графе создателей (обычно ведь коллективные проекты), думает о ней, удерживает внимание на предметах метода работы с системой. Думает — значит проводит какие-то операции с её описанием, без этого нельзя в том числе проводить и другие работы с инструментами, меняющими систему в физическом мире. Варианты создания и развития систем без каких-то работ по её проектированию или восстановлению проекта/design в ходе «обратной инженерии»/«reverse engineering» мы тут не рассматриваем, «не пишешь — не думаешь», проекты/projects без проектной/design документации имеют минимальные шансы на успех.

Если какая-то роль не может описать какую-то ипостась системы, значит систему из окружающего мира она ещё не выделила. **Мы должны чётко различать существующее всегда описание системы как альфу (возможно в состоянии «описание системы в объёме короткой идеи о концепции использования») и существующую не всегда документацию системы как рабочий продукт/артефакт (в том числе электронный документ).**

Иногда переводят описание системы/system definition как «определение системы». В этом случае нельзя путать термин со «словарным определением», типа «наша система — это то-то и то-то». Нет, это самая разная информация о воплощении системы, она включает самые разные модели целевой системы, а часто это конструктивное описание (в смысле конструктивизма: «чтобы описать объект, надо задать операции его построения»), так что явно или неявно также описываются создатели из графа создателей и их операции — это всё и будет описанием системы (system definition). Одной фразой «определения из словаря» её не заменишь, в system definition совсем другое значение термина definition, это не отсылка к «определению, как в словаре».

В жизни при прямой инженерии обычно всё начинается с разработки (выявления, придумывания, согласования между собой) и документирования каких-то ролевых видов описания (концепции использования, концепции системы, архитектурных решений и так далее), а потом от проектных описаний команда проекта переходит к воплощению::метод системы в физическом мире (изготавливает/производит систему, вводит её в эксплуатацию) — и всё это в непрерывном цикле развития системы, через множество версий системы, собираемых из множества версий её частей. Если это обратная инженерия/reverse engineering, то всё наоборот: берём систему в физическом мире и восстанавливаем все описания для уже готовой системы. **Обычно в проектах для надсистемы идёт обратная инженерия, а для целевой системы — прямая.**

Если вы попали в какую-то организацию (клиента, подрядчика, или только пришли в новую для вас команду проекта), то вы делаете обратную инженерию этой организации: понимаете, что там происходит в графе создателей. Если вам надо, наоборот, организовать создателя системы — делаете прямую инженерию, готовите все эти описания как проектные и затем убеждаете агентов в организации следовать этим описаниям (руководство по системному менеджменту как раз об этом, помогает это делать вся программа рабочего развития инженера-менеджера).

В философской логике рекомендуют всегда начинать с воплощения системы, даже если это воплощение будет существовать только в будущем: **нужно представить себе (породить/generate описание на базе имеющихся у вас порождающих моделей/generative models мира, знаний о мире), а затем документировать это описание возможного мира/possible world, в котором воплощение системы уже существует!** Ни на секунду не нужно забывать, что нужно-то воплощение системы, а описание (данное нам в виде информационных моделей, чертежей, инструкций по изготовлению и т.д.) нужно только в силу того, что без описания очень трудно воплотить в жизни работающую систему. Запоминаем: технолог, когда прописывает технологию производства кефира — делает кефир, а не просто описывает метод работ по изготовлению кефира. От него не документы в конечном итоге с описанием технологии (метода изготовления, нужного для работ по этому методу инструментария) изготовления кефира нужны, от него в конечном итоге требуют кефир в его физическом воплощении — это его целевая система. Но кефира-то ещё нет, пока технолог описывает технологию его изготовления, и инструментария по изготовлению кефира нет, пока технолог его не описал. Ничего страшного, в возможном мире они уже есть, в будущем! Вот это и описываем.

Заданного направления чтения (слева направо, или справа налево, или снизу вверх, или сверху вниз) диаграммы альф (объектов внимания, которые будут менять своё состояние в ходе проекта) для системного описания нет. Разные проектные роли для разных своих целей читают эту диаграмму в разных направлениях. Но главный элемент диаграммы — это, конечно, воплощение системы. Ради него вся деятельность и затевается, его всегда нужно держать во внимании.

Все подобные диаграммы — это чеклисты альф, которые нужно удерживать во внимании при мышлении о проекте, а для каждого объекта-альфы будет ещё и чеклист по состояниям, через которые её нужно провести. Подробней о чеклистах будет в руководстве по методологии, в нём также будут приведены примерные чеклисты для примерно трёх четвертей основных альф, которые встретятся вам в проектах. И будет наставление, как готовить эти описания. **Описания альф и их**

состояний — это описания альфы «метод работы» создателя — это предметы метода, которые меняются в ходе работ какого-то создателя.

Стандарт ISO 42010:2022 даёт рекомендации о том, как думать о системном описании/описании системы. Сам стандарт говорит только об архитектурном описании, причём в старом понимании архитектуры, но его положения вполне применимы к любым описаниям, то есть и концепции использования, и концепции системы, и архитектуре.

Вот задающая мышление об описаниях системы диаграмма из этого стандарта, модифицированная с использованием положений OMG Essence, а также обобщённая с архитектурного описания какого-то предмета интереса (entity of interest) на системное описание в целом:

Диаграмма системного описания начинается с уже знакомого нам различения воплощения (realization) и описания (definition) системы.

Для простоты диаграммы большинство отношений (кроме одного:

удовлетворяет/описывается в одну сторону и характеризует/описывает в другую сторону) приведены только с одним наименованием связи, но это не значит, что не существует обратного отношения. Это общее свойство всех подобных схем: даже отношение «часть-целое» в одном направлении читается «включает_в_себя» или «состоит из», в другом направлении читается «является_частью» или «входит в состав».

Каждая связь между объектами этой диаграммы может быть прочтена в двух направлениях, например «описание системы выражается документацией системы/системной документацией», «системная документация документирует системное описание». Говорят и «системная документация», и «документация системы», и «системное описание», и «описание системы».

Обратите внимание, что документация не фиксирует, а документирует постоянно меняющиеся в ходе проекта описания (системное, ролевое, модель)! Избегайте слова «фиксирует», потому как люди обычно думают, что это означает не документирование (как думаете вы, произнося слово «фиксирует»), а объявление описания уже не изменяющимся, «заморозку требований» или «заморозку проектных описаний». «Запиши» — это документируй, а «зафиксируй» — это «перестань менять», совершенно другое действие! Вас обязательно поймут с «фиксацией» как «дальше не обсуждаем, а выполняем».

Фиксация или фиксирование требований в инженерии прошлого (сейчас так иногда продолжают говорить о фиксации концепции использования, а ведь исключение фиксации/заморозки требований и было одной из причин отказа от разработки требований в пользу разработки концепции использования) для большинства людей означает не запись требований на бумагу или в базу данных, а объявление текущего состава требований стабильным и неизменяемым. Но в системном мышлении вы лучше и лучше описываете систему (или описываете всё более лучшую систему, по мере понимания того, как именно она будет использоваться, в каком окружении), меняя и меняя тем самым концепцию использования — документирование этому не мешает, модели системы не «фиксированы» ни в один момент времени проекта! Вы можете какую-то версию административно объявить стабильной. Инженеры говорят в этом случае о базисе/baseline, но это не означает «фиксации»! Просто изменения системных описаний будут собираться и документироваться в других артефактах (документе «рабочей версии», а не документе уже утверждённого «базиса»), а потом базис будет меняться по какой-нибудь административной процедуре управления изменениями проекта/change management. Но и тут изменения: магистральная разработка/trunk-based development[403] подразумевает, что вся разработка не порождает отдельные долго живущие ветви, а в современной инженерии (например, на заводах Tesla, в SpaceX) каждый отдельный автомобиль или ракета имеют уникальные описания (project description), выпускаются по уникальному проекту и производственному процессу каждый, нет двух одинаковых автомобилей или ракет. Но с этим разбирается уже третье поколение системного мышления, там речь о версионировании — и как жить, когда версия каждого экземпляра системы уникальная, а сами эти системы непрерывно развиваются.

Пока же запоминаем: диаграмма не говорит о том, что разово делается одна версия

[403] https://trunkbaseddevelopment.com/

системного описания, которая потом будет задействована при изготовлении системы. Нет, диаграмма просто говорит о том, какие объекты внимания в проекте у вас есть, когда вы описываете систему и документируете эти описания. Это альфы, так что вы должны удерживать во внимании не просто эти объекты, а изменения состояний этих объектов (предметов самых разных методов инженерии и менеджмента) в ходе проекта.

Вот эта же диаграмма на английском языке:

Описание системы (system definition) состоит из **ролевых/аспектных/видов описаний** системы (views), которые определяются важными характеристиками (concerns, ролевыми предметами интереса/озабоченностями) и задаются аспектными методами описаний (viewpoints). **Ролевые/аспектные/виды описания системы** — это как раз концепция использования (разрабатывается ролью инженера-проектировщика), архитектура (формулируется ролью инженера-архитектора), и так далее — описания, которые делаются различными инженерными ролями, даются в руководстве по системной инженерии. Но в менеджменте это будет, например, не просто архитектура, а орг-архитектура, а в

качестве «концепции использования» у организации будет выступать бизнес-модель, где обосновывается необходимость организации (и за неё будет ответственной роль бизнесмена). Аспектные описания — это что угодно, что описывает систему удобным способом для обсуждения ролевых интересов.

Конечно, ролевых/аспектных/видов описаний может быть множество. Например, роль могут интересовать финансы. Тогда ролевое описание — финансовое, модели там будут — бухгалтерский баланс, отчёт о прибылях и убытках, поток денежных средств. Как сделать эти описания? Эти описания могут быть сделаны по методу описания РСБУ (Российская система бухгалтерского учёта) или МСФО (Международная система финансовой отчётности), а мета-модели — это форматы представления бухгалтерского баланса, отчёта о прибылях и убытках, потока денежных средств в соответствующих ролевым методам описания (РСБУ или МСФО). Если финансиста заинтересуют оба метода описания, то он получит два описания — и по РСБУ, и по МСФО. Как будут документированы эти описания? Или на бумаге, или в табличках Excel, или в базе данных компьютерной бухгалтерской программы — в любом случае, содержание описаний можно обсуждать без обсуждения формата представления на носителях информации.

Моделирование: описания, методы описания, предметы интереса описания, заинтересованные стороны

Заполните табличку для не менее чем пяти видов описаний, которые вы нашли в вашем проекте.

Номер описания	Описываемая система	Вид описания	Документ, в котором приведено описание	Метод описания	Документ, в котором приведён метод описания	Область/предмет интереса описания	Заинтересованные в этом описании роли

Системное описание (предметов) деятельности предприятия

Схема/schema[404] в мышлении — это не обязательно диаграмма, а просто набор понятий и отношений между ними как некоторый шаблон для мышления, опора для внимания в мышлении (framework, «о чём думать», что держать во внимании, чеклист важного). Схема вполне может быть выражена и текстом на естественном языке, и текстом на формальном языке (моделирования или программирования), но можно выразить и диаграммой. Подробнее про схемы, диаграммы, тексты и разные другие виды представления описаний говорится в руководстве по рациональной работе. В последнее время реже говорят о схеме, чаще говорят об онтологии (многоуровневое множество хорошо согласованных понятий и их отношений) или онтике (небольшое часто одноуровневое множество понятий, которые удобны для

[404] https://plato.stanford.edu/entries/schema/

описаний предметов какой-то предметной области, вот её и называют обычно framework).

Весь текст нашего руководства даёт некоторую онтологию/«большую схему системного мышления» — связанный отношениями набор понятий системного мышления. Для некоторых частей этого набора понятий и их отношений (онтологии системного мышления) мы приводили не только текстовое описание, но и давали диаграмму: графическое изображение онтологии как графа, где понятия изображены какими-то узлами, а рёбра графа означают отношения между понятиями. Была диаграмма для проектных ролей, в предыдущем подразделе мы рассмотрели диаграмму описания системы.

Жизнь показала, что инженеры-менеджеры не считают отдельные наборы понятий и отношений по каким-то темам из руководства по системному мышлению частями одной связной онтологии, одной «большой схемы» (повторим: схема тут термин для набора понятий, связанных отношениями, может быть представлена как текстом, так и диаграммами. Текст нашего руководства — это как раз и есть представление набора понятий и отношений системного мышления).

Если говоришь «проектная роль», то инженеры-менеджеры тут же вспоминают про предпочтения, но забывают, что роль выполняет какой=то метод работы (это же в других абзацах!). Если говоришь «важная характеристика», то тут же вспоминают про предпочтение, но забывают о том, что ролевое описание/view отвечает на вопросы по этой характеристике ровно для того, чтобы понять, что там происходит с предпочтением, а сам предмет интереса отражается/framed ролевым методом описания/viewpoint (это же в совсем другом разделе нашего руководства!).

Проблема в том, что все понятия из теорий/знаний/объяснений интеллект-стека фундаментальных методов мышления (включая и понятия методологии, и понятия системного мышления) применяются в рассуждениях совместно, а не клочками. Если ты в ходе работы по какому-то методу подумал о чём-то важном (помним, что понятия для предметов метода представляют чеклист того, что нужно бы продумать в той или иной ситуации), то дальше подумай и о тех важных предметах метода, что связаны известными из теории/объяснений отношениями с тем предметом, о чём ты сейчас подумал — и так тебе будет удаваться не пропустить в мышлении какие-то важные предметы метода.

Если вспомнил про какого-то человека (или даже более крупное оргзвено) в проекте, что у него есть «трудовая роль»::тип (применил тип из нашего руководства!), то вспомни и о её «предмете интереса/важной характеристике системы»::тип, и о «предпочтении в характеристике»::тип, но также вспомни и о «методе работы роли»::тип, и о «ролевом/аспектном описании»/view»::тип, а также его «методе описания»/viewpoint::тип! Что тут «вспомни»? Вспомни о том, что в рассуждении тебе нужны объекты этих типов, и найди эти объекты в реальной жизни! И тут мало будет просто вспомнить об альфах, состояния которых меняются в ходе всего проекта, ещё надо будет проявить собранность: помнить о них в ходе всего проекта, постоянно удерживать на них внимание (то есть лучше бы не просто вспомнить, но и записать вспомненное, чтобы тут же не забыть).

Если важные объекты в проекте почему-то сходу не находятся — задай вопросы участникам проекта, получишь неожиданные ответы! Этот список «о чём нужно подумать, состояния чего нужно отследить» не так велик: в жизни столько всего мелькает, воет и трясётся в проекте, отвлекая внимание, что фокусирование

внимания на вот этих понятных и привычных ходах мысли по объектам-типам-предметной-области, для которых мы находим объекты-в-жизни, сэкономит много времени на сокращении размышлений о неважном. Как отделить важное от неважного? Важное в непонятных ситуациях — это то, для чего человечество выбрало типы мета-мета-моделей (они прошли довольно долгую эволюцию, интеллект-стек фундаментальных методов мышления продолжает развиваться), а важное в более-менее понятных ситуациях — это то, для чего есть типы мета-моделей конкретных предметных областей.

Вот пример диаграммы для системного описания (предметов) деятельности предприятия. По факту это диаграмма альф, ибо альфы — это предметы деятельности, деятельность — один из синонимов «метода».

Это иллюстрация, которая показывает, что рассмотрения предметов методов работы разных оргролей в графе создателей предприятия, тесно связаны. Для каждой оргроли, которая выступает как система в графе создателей какой-то системы, приведено моделирование её предметов метода как альф, но кто там что создаёт (кто какие предметы метода меняет своими работами) — вот это и можно посмотреть.

Мы не поощряем работу с подобными диаграммами при моделировании рабочих проектов, ибо в рабочих проектах диаграммы надо менять часто, а это обычно более чем трудоёмко, плюс разные части диаграммы надо обычно менять разным людям, да ещё и моделеры для диаграмм малодоступны, не на каждом рабочем месте такой найдёшь. В руководстве по методологии будет рассказано, как моделировать альфы предприятия не такими диаграммами, а списками (сверхкраткая форма) и табличками (удобная форма, поддерживаемая практически любым операционным софтом). Мы даём тут диаграмму только в качестве иллюстрации, нам надо показать, как делается моделирование целевой системы и графов создателей какого-то предприятия с использованием основного набора (kernel) из четырёх альф для каждой системы: воплощение системы, описание системы, метод работы системы, работы системы. Подробно эта диаграмма раскрывается в руководстве по системному менеджменту, которое посвящёно созданию и развитию организаций.

Будет ли после разглядывания этой диаграммы[405] (если вы сможете там что-то прочесть из-за мелкого шрифта, проблема большинства больших диаграмм на маленьких экранах!) при появлении в мышлении роли::«тип какого-то объекта в предметной области» (скажем, при взгляде на экран с «Аристархом Владимировичем»::агент-AI вспоминается, что он инженер-архитектор::роль) приходить в голову теперь не только желание подумать про его предметы интереса и предпочтения в них, но и приходить желание подумать про практики архитектурной деятельности? Желание подумать про методы описания, которые устанавливают соглашения для описаний, используемых в архитектуре (например, ADR как architecture decision records), чтобы поддержать содержательный разговор с Аристархом Владимировичем как инженером-архитектором? Не факт. Много текста, который продемонстрирует примеры рассуждений (то есть ещё и выполнил заземление/grounding), тут поможет больше.

Нейронная сетка людей тренируется лучше употреблением слов, а не разглядыванием картинок: человечество получило цивилизацию после того, как научилось писать, картинок для передачи знания не хватает. К каждой картинке всё равно приходится писать (а если не писать, то «приговаривать устно», что ещё хуже) много-много слов. Картинки трудно прочесть, их трудно редактировать.

Сама по себе «картинка» иллюстрации к описанию предметов работы предприятия содержит очень немного знаний, всё равно её приходится описывать сотнями страниц текста как нашего руководства, так и руководства по методологии. Не факт, что такая картинка что-то дополнительно проясняет: чем больше понятий и отношений содержит диаграмма, тем менее она читаема, менее понятна, менее употребима. А если её разбить на части — то и мышление разбивается на части, у него нет уже целостности всей дисциплины. И возвращаемся к исходному вопросу: как связать в мышлении хорошо усвоенные кусочки мышления про целевую

[405] Вот эта диаграмма в высоком разрешении — https://ic.pics.livejournal.com/ailev/696279/247262/247262_original.jpg, оригинал файла делался в редакторе yEd, https://disk.yandex.ru/d/qMDMgCZTkkT4-w

систему, создателя (инженерную команду) целевой системы, создателя инженерной команды (менеджерскую команду), создателей менеджерской команды (команду инвестпроекта), а ещё увязать работу службы продвижения и даже работы клиентуры, клиенты которой должны потрудиться, чтобы купить наш продукт и использовать его в своих надсистемах? Тут ещё проблема в том, что всем этим надо заниматься разным людям — и каждому придётся разбираться подробно со своим кусочком диаграммы.

Самая большая беда будет, если вот такая диаграмма будет задействована в оргработе на предприятии: она будет как картина хорошего художника — смотреть на неё приятно, но вот поправить что-то на ней будет очень трудно, поэтому эти правки не будут выполняться, а диаграмма будет довольно скоро существенно расходиться с жизнью, будет вечно неактуальной. Все модели в системном мышлении должны быть легкоизменяемыми и ставиться под управление конфигурацией (всегда должен быть ответ на вопрос, кто что когда менял, изменить должно быть легко, равно как легко проверить, нет ли в изменениях каких-то ошибок). Диаграммы для постоянных изменений существенно хуже текстов на формальных языках и хуже таблиц. Поэтому в жизни вместо диаграмм используем табличное моделирование: системное мышление (табличным) моделированием. Примеры такого моделирования — это выполнение заданий по моделированию в нашем руководстве.

В любом случае, нужно понимать, что постепенно и последовательно излагаемый в руководстве набор понятий как типов объектов из жизни, которые задаются знаниями/дисциплинами/онтиками интеллект-стека фундаментальных методов мышления (включая набор понятий системного подхода), представляет собой некоторую «большую схему/онтику системного метода описания». Отсутствие для такой «большой схемы/онтики метода описания» диаграммы как графического представления схемы ровно ничего не означает. Вы смело можете считать, что всё руководство просто своим текстом постепенно описывает набор типов мета-мета-модели, в котором какими-то отношениями связаны все упоминаемые в руководстве понятия, но **огромной диаграммы-картинки (даже разбитой на части) для них всех не будет. Такая картинка бесполезна (это очень контринтуитивно, но такая картинка и впрямь бесполезна!), но вот иные формы представления (тексты на естественном языке, тексты на формальных языках, табличные представления) — они вполне работают.**

Системное описание, описание, метод описания

В системном описании есть различные аспектные/ролевые/частные описания (views), которые отвечают на вопросы о каких-то «областях интереса»/«area of concerns». Проектная роль может иметь и несколько областей интереса, и множество интересов в каждой из областей интереса. Это всё интересы к предметам методов работы этой роли.

Каждое ролевое/аспектное описание позволяет описать систему так, чтобы поддержать разговор с проектной ролью по предмету/теме/зоне/области её интересов, ответить на интересующие её вопросы по достижению её предпочтений в интересующих характеристиках систем. Таких описаний делается множество, значительное число таких описаний гибридны (совмещают описания каких-то областей интереса, показывают несколько аспектов системы и связи между ними, например концепция системы как ролевое описание разработчиков и архитекторов

системы показывает согласованные между собой описания функциональные, конструктивные, компоновочные и стоимостные).

Каков результат использования множественных описаний одной и той же системы, да ещё и многоуровневых по разным видам системных разбиений (то есть речь идёт о системном описании — системное это не «полное по видам описаний», это «описание с учётом множества системных уровней, определяемых по различным критериям разбиения»)? Результатом тут будет успешная система: если проектная роль соглашается с описанием, то воплощение системы, удовлетворяющее этому описанию, эту роль, скорее всего, тоже удовлетворит (помним про инженерные обоснования: мы ещё и обоснуем в ходе испытаний, что воплощение системы ведёт себя так, как это описано в её системных описаниях).

Успешность системы мы часто можем определить ещё тогда, когда воплощения системы ещё нет, а есть только описание системы, и даже не полное описание (ошибки неучёта каких-то интересов могут быть видны и по частичным описаниям). Использование 3D-информационных моделей, чертежей, имитационных моделей в инженерии, нот в музыке и любых других записей в проектировании и конструировании как раз имеет целью подробно обсудить реализацию самых разных предпочтений в важных для всех ролей в проекте характеристиках/concerns разных систем проекта ещё до того, как целевая система будет воплощена. Рассуждение не меняется, если речь идёт просто о выпуске очередной улучшенной версии системы (не greenfield разработка «с нуля», создание системы, а brownfield разработка улучшений, развитие системы).

Проектные роли при этом получают доступ к описаниям, но описания сами по себе нематериальны, абстрактны. В реальности агенты-в-ролях смотрят документацию/«рабочие продукты»/артефакты: без носителя информации нет способа предъявить описания для их обсуждения. Так что **первым делом все описания нужно документировать, не держать в голове!**

Описаний же много: разные проектные внутренние и внешние роли интересуются крайне разными **ролевыми/аспектными описаниями/views** системы. В этих описаниях/views отражаются **важные для самых разных ролей характеристики/concerns** систем самыми разными **ролевыми/аспектными методами описаний/viewpoints, по которым и делаются эти описания/views**. И, конечно, каждое описание может быть документировано разными способами (например, с использованием разных нотаций — числа могут быть записаны арабскими цифрами, римскими цифрами, в двоичной системе), на разных носителях (бумага, магнитная лента, компьютерная оперативная память, даже мозг живого человека — носитель очень малоёмкий, плохой и ненадёжный, но это ведь тоже носитель информации! Другое дело, что запись информации в мозг делается в нелокальных/распределённых представлениях, а запись информации на компьютерные носители делается сейчас в локальных/символьных/знаковых представлениях. В инженерии сейчас используют знаковые представления, они устойчивы к накоплению ошибок при многократной репликации/копировании. Так что важно не держать всё в голове, это крайне ненадёжно!).

Все эти многочисленные ролевые/тематические/«по интересам»/аспектные описания (views) — подальфы системного (полного/всестороннего/для всех ролей/для всех предметов интереса, для всех системных уровней) описания. Их,

конечно, много больше, чем концепция использования, концепция системы, архитектура, они часто хитро структурированы (так, концепция системы в машиностроении развивается до технико-экономического проекта, затем эскизного проекта, затем рабочего проекта — который и идёт на производство). Тут и финансовые модели, и имитационные физические модели, и прогнозы надёжности, а также многочисленные описания того, что должны делать системы создания (скажем, для стройки это будет ПОС, план организации строительства — то, что должны делать строители во время возведения здания или сооружения, совмещённое описание методов работы строителей и самих работ строительства): описаний много, для какой-нибудь атомной станции в бумажном виде лет двадцать назад это было несколько грузовиков с бумагой, и это была ещё не вся документация!

Разница между описанием (альфа) и документацией (рабочие продукты) принципиальна, но в жизни очень часто в речи их путают. Так, легко путают архитектуру (часть описания системы) и архитектурную документацию (рабочие продукты/артефакты/документы/файлы, отражающие архитектурные решения на каком-то информационном носителе). Очень часто слово «документация» опускают и предупреждают читателей, что из контекста должно быть понятно — об архитектуре (описании системы) говорят или об архитектурной документации (наборе рабочих продуктов/артефактов/документов/записей баз данных с записями архитектурных решений архитектуры).

В литературе (например, ISO 42010:2022) часто слово «описание» даётся не для абстрактного объекта-альфы, а для документации/артефакта, и на английском это будет description. В системной инженерии наше абстрактное «системное описание» будет system definition. В русском языке слово «определение» путается с словарным, поэтому мы перевели более подходящим по смыслу словом «описание», а description тоже перевели более подходящим по смыслу русским словом «документация». Путаница с описаниями вездесуща. Так, онтология (или архитектура, или цвет системы) могут означать:

- То, что есть в жизни (нарезка мира на какие-то объекты, или архитектура какой-то системы, или цвет какой-то системы — они «в реальности»)
- Описание (обычно признаётся, что это какой-то аспект из разных описаний, тогда view, если это проектное описание, «как будет», то это definition). Тут часто говорят «онтология», «архитектура», «цвет» — но имеют в виду онтологическое описание (или синонимично — описание онтологии), архитектурное описание (описание архитектуры), цветовое описание (описание цвета). И это мы ещё не трогаем путаницы с «описанием»::«предмет описывания::метод». Метод/процесс/деятельность описывания ведь тоже часто называют описанием! У нас тут — предмет метода, а не сам метод. Метод описания по-английски — viewpoint.
- Документ с описанием, и тут тоже часто говорят онтология, архитектура, цвет. Но это уже название документа!

Вы вполне сможете разобраться в любом варианте терминологии, употреблённой в том или ином описании (pun intended), если понимаете, что всегда существует три разных объекта (описываемый, описание как идеальный, и ещё материальный как описание на носителе информации) — они иногда обозначаются разными словами, принятыми в разных профессиональных сообществах или разных естественных

языках, но очень часто — одним и тем же словом.

Слова важны (кроме слов у нас ничего больше нет), и не важны (слова обычно означают не то, что вы думаете — и в словаре их значение не подсмотришь, ибо словарей много, язык живой). Внимательно отслеживайте типы тех объектов, которые скрываются за подозрительными терминами (подозрительные — это которые вызывают у вас онтологический дребезг, это было в руководстве по рациональной работе), и всё будет в порядке.

Какие термины могут оказаться подозрительными? Не только «описание», но почти все термины системного мышления, включая термин «система»! Очень часто ведь «система» используется в бытовом значении, просто для придания «научности» или «инженерности» говоримому, иногда как синоним слову «объект» («что угодно, что рассматриваем») — и никаких за ним системных уровней, эмерджентности, графа создателей, непрерывного развития системы — ничего подобного говорящим «система» не подразумевается, ибо не все говорящие слово «система» знакомы с современным системным подходом. И «концепция использования», «архитектура» тоже вполне могут использоваться не как подальфы для альфы системного описания (как предметы метода инженерной работы, с целью отследить изменения их состояния в ходе проекта создания и развития какой-то системы), а в каких-то совсем других значениях (например, означать документы с названиями «концепция использования» и «архитектура»).

Полное системное описание (description) встречается в жизни довольно редко, хотя всегда в проектах ведутся попытки как-то собрать в одном месте многочисленные документы с разными ролевыми описаниями. Много чаще встречаются просто отдельные ролевые описания системы, документированные в разных документах (разных информационных системах, разных базах данных).

Это только геном обязательно будет в каждой клетке организма, собран в одном «документе», на одном носителе информации. Геном[406] сам по себе — это носитель информации, «совокупность наследственного материала, заключённого в клетке организма», то есть это набор всех хромосом[407], каждая из которых образуется у эукариот одной очень длинной молекулой ДНК. Геном — это документ с описанием генетической информации. Более чем часто имеют ввиду под геномом не набор хромосом/«молекул ДНК» (документ), а саму генетическую информацию, документированную геномом! В инженерии речь идёт о мемоме (по аналогии с геномом), который хранится где-нибудь в конструкторском бюро и является результатом не биологической (дарвиновской), а техно-эволюции. Договорённости о том, что мемом — это материальные носители (проектная документация) нет, поэтому это может быть равным образом проектная/design документация, но может быть и проектное/design описание. Идея о том, чтобы собрать всю проектную документацию в одном месте — она обычно провальна. Даже если считать, что вся проектная документация находится в каком-нибудь конструкторском бюро, то немедленно окажется, что есть производство, на котором тоже есть часть документации, часть документации находится у подрядчиков (например, поставщиков заказных комплектующих), часть документации будет у наладчиков и операторов (например, какие-то настройки системы, о них даже проектировщики

[406] https://ru.wikipedia.org/wiki/Геном
[407] https://ru.wikipedia.org/wiki/Хромосома

не будут знать). Так что идея о том, что системное описание (systems description) отражается в каком-то «супердокументе» — она провальна.

Исходя из этого, view обычно переводится просто как «описание», без указания на то, что оно ролевое/аспектное/тематическое/предметное/частное, а не (полное) системное. Каждый раз, когда вы слышите «описание», оно будет давать ответы на вопросы только по части предметов интереса/важных характеристик, для ответов на вопросы по другим предметам интересов нужны будут другие view, подготовленные по другим методам описания/viewpoint.

Концепция использования — это тоже описание системы (хотя сценарии использования считаются существующими, а в концепции использования мы имеем не сами сценарии использования, а описания сценариев использования, но слово «описание» будет пропущено), и архитектура — описание (та же ситуация: архитектура есть у системы, а архитектурные описания будут содержать описания принятых важных архитектурных решений, в жизни это всё будет называться просто «архитектура» и чаще всего будет означать описания), и отчётность по РСБУ и МСФО — описания, но они каждое — не полное системное описание.

Тем не менее, мы говорим об альфе системного описания (system definition), поскольку в проекте совокупность всех описаний необходимо отслеживать в части их готовности. Продвинуть альфу системного описания по её состояниям можно, продвигая работами проекта по их состояниям подальфы отдельных частных/аспектных/ролевых описаний (views). Этих подальф описаний/views для альфы описания системы довольно много.

Прежде всего, самые разные роли в проекте интересуются состояниями продвижения по ходу проекта для функциональных описаний, конструктивных/модульных/продуктовых описаний, описаний размещения и стоимостных описаний как частей концепции системы. Это главное, их будут проводить через состояния «концепция системы готова», «технико-экономический проект готов», «эскизный проект готов», «рабочий проект готов» (помним, что в разных сферах деятельности это будут очень разные состояния, даже в машиностроении, из которого мы позаимствовали этот пример). Но дальше может быть ещё огромное количество описаний, интересующих самые разные трудовые роли в проекте: например, синхронизации во времени разных подрядчиков, структуры владения частями окружения, информационных потоков и т.д.

Чем сложней (больше системных уровней, больше подсистем на каждом уровне) система, тем бóльшего количества описаний/views можно ожидать. **Важные для людей-в-ролях характеристики систем многообразны, и успех системы определяется как раз полнотой их учёта (записать!), а затем полнотой согласования значений этих характеристик. Не учёл, а поэтому не согласовал описания, затрагивающие, например, морозостойкость::concern — не получил по таким описаниям успешную систему. Или не учёл, а поэтому не согласовал различные описания, которые показывают поведение системы на разных системных уровнях — тоже не получил успешную систему.**

Разнообразные объекты, с которыми работают в проекте разные роли, являются:

- С точки зрения методов работы этих ролей — предметами метода,
- с точки зрения мотивов, побуждающих агентов к действию, являются предметами интересов,

- с точки зрения изменения состояния этих предметов, являются альфами,
- с точки зрения мышления об этих объектах — классифицированными типами различных моделей, мета-моделей, мета-мета-моделей, и даже мета-мета-мета-моделей.

Это нормально, но это означает сразу, что самые разные объекты будут каждый описан много раз с использованием множества разных методов описаний. В системном мышлении постулируется:

- Необходимость системных описаний (system definition). Они важны, их надо делать и документировать, даже если очень лень и некогда.
- Необходимость явного указания для каждого из описаний/view в составе системного описания их методов описаний (viewpoint). Если вы не знаете, каким методом вы что-то описываете, то это не значит, что вы описываете это без метода, поэтому подумайте о методе и укажите его.
- Представленность/presentation описаний в физическом мире в виде документов/файлов/артефактов. Не держите описания в чертогах вашего разума, а делайте их в читаемом другими агентами (людьми, машинами) виде — и первым из этих людей (или машин, это руководство проходят и AI-агенты) будете вы сам, а эти описания помогут вашему биологическому (или не биологическому) мозгу быть надёжным в памяти и удержании внимания.

Документация проекта (system description) должна отражать значения важных характеристик систем в проекте, чтобы в ходе обсуждений/согласований предпочтений в этих характеристиках разных ролей проекта удерживать на них внимание не «ментально, усилием воли, хорошей памятью людей-агентов», а технически, используя «внешнюю память», экзокортекс. Об этом наставляется во всех наших руководствах, начиная с руководства по рациональной работе: человечество ушло от «мышления голыми мозгами» примерно так же далеко, как от «работы голыми руками», без каких-либо инструментов.

Как именно делать эти описания в самых разных рабочих проектах, рассказывается в соответствующих методах описания самых разных методов прикладной инженерии: классической «железной» системной инженерии, программной/software инженерии, обучения как инженерии личности, менеджмента как организационной инженерии, социальной инженерии и так далее.

Например, музыку записывают в виде партитуры[408] в нотной записи на бумажных листах. Как сделать партитуру для новой свежесочинённой песни? Это прикладное знание, и оно будет крайне разным в плане нотаций для партитуры на бумаге и может принимать совсем другую форму, если речь пойдёт о компьютерной музыке. В системном мышлении вы только должны помнить, что:

- если есть «песня» (предмет метода),
- то она должна как-то быть описана, для этого нужно использовать метод описания. Уж как именно описана (нотация roll piano, или древнерусская нотация крюками, или современная нотная запись) — это изучается прикладными методами работы с песнями.
- это описание должно быть документировано. Не документировать нельзя (хотя есть и разные другие мнения, ведь ещё есть импровизационные жанры,

[408] https://ru.wikipedia.org/wiki/Партитура

но там это будет не «песня»[409])!

Если у вас в проекте есть и партитуры с нотной записью, и чертежи, и расписание, и органиграмма, то системное мышление позволяет обсудить все эти отдельные описания, не потерять их — оставив прикладным дисциплинам подробности того, как именно они делаются и как ими пользоваться.

Системное мышление — это фундаментальные, пронизывающие все прикладные методы мышления приёмы («маленькие методы/способы») мышления на базе понятий системного подхода. **Знания системного мышления не могут заменить знания прикладных практик. Знать о том, что «описания всегда есть на каком-то уровне абстракции мета-моделирования, но они могут быть не конкретизированы и не документированы» — мало. Нужно ещё знать, как делать эти описания/view, то есть нужно знать прикладные методы классической и программной инженерии, менеджмента, медицины, правоохраны, политического действия и т.д., а в этих методах ещё и знать включённые в них методы описаний (viewpoints). Только мастерства системного мышления или даже интеллекта (как фундаментального мастерства мышления в новых ситуациях, когда непонятно, что делать) для работы в проектах по созданию и развитию систем не хватает, нужно ещё владеть прикладным мастерством в том методе работы проекта, который вам надо использовать в вашей проектной роли.**

Метод описания (viewpoint) отражает (frame) важную характеристику/предмет интереса (concern), а описание (view) отвечает на вопросы о значении целевой характеристики, удовлетворяет к ней интерес. Например, «легенда карты фауны»::метод описания отражает «распределение фауны на определённой территории»::concern, а карта::описание отвечает на вопросы о «наличии или отсутствии фауны в тех или иных конкретных местах»::«значение целевой характеристики/предмета интереса».

Тем не менее, предмет интереса/concern, и «метод описания»/viewpoint, как и описание/view нещадно путают (то есть путают «распределение фауны как характеристику какой-то эко-системы/территории»::concern, «легенду карты [распределения] фауны»::viewpoint и саму «карту фауны»::view). Конечно, всё это близкие понятия, но все они привлекают внимание к разным объектам, поэтому для обсуждения нюансов они нужны все: надо найти в проекте объекты всех упоминаемых типов: concerns, viewpoints, views.

Если у меня ролевой предмет интереса/concern/важная характеристика — цвет системы, а моё предпочтение/preference/интерес — яркий цвет, у меня будет недовольство тем, что цвет неяркий. Я выбираю метод описания цвета (отражаю интересующую важную характеристику при помощи метода описания), включающий возможность описать значения насыщенности цвета — это позволит мне прогнозировать яркость цвета в системе моего интереса, испытывать полученную яркость окрашенной системы, формулировать предпочтения для яркости цвета, договариваться с другими проектными ролями по поводу насыщенности цвета. Другая проектная роль — инженер, который должен покрасить систему. И он может заботиться о том, чтобы краска была дешёвая (удовлетворять интерес ещё одной проектной роли — менеджера-финансиста), а

[409] https://polit.ru/article/2007/10/05/martynov/

яркость этой краски его волновать не будет. Но дешёвая краска яркой не бывает. Если я документирую нужную мне цветность, описав при помощи стандартных характеристик colorfulness, chroma, saturation, плюс учту внешнее освещение (brightness), то мы как минимум сможем провести переговоры на тему приемлемых характеристик цветности, по итогам переговоров инженер сможет подобрать дешёвую, но достаточно яркую на мой ролевой вкус краску (ролевой вкус! Как агент я могу предпочитать неяркие цвета, но в проектной роли я делаю профессиональное суждение по поводу яркости: реализую не собственное «как человека», а ролевое предпочтение) — моё ролевое предпочтение «яркого цвета»::«значение важной характеристики» будет реализовано.

Выбор метода описания оказывается нелёгким: описывать цветность и насыщенность цветов можно по-разному[410]. Мы в конечном итоге выбираем «метод/способ описания»/viewpoint, и в рамках этого выбранного метода описания обсуждаем мои хотелки/предпочтения иметь цвет поярче — моё намерение тут или заставить краску выбрать подороже, или даже сделать дополнительную подсветку (что ещё дороже, но я буду искать союзников, которым понравится эта опция для реализации каких-то своих предпочтений).

Если не договорились, то я::агент продолжаю реализовывать предпочтение своей роли иметь цвет поярче. Для этого я::агент перехожу в режим изобретательности и устраиваю какую-нибудь интригу (например, звонок какого-нибудь большого начальника::агент-с-полномочиями актёру, исполняющему роль инженера, и от этого же начальника звонок исполнителю роли финансиста). На следующих переговорах они оба сдаются, и в системную документацию попадает описание цвета системы как достаточно для меня яркого. Для того, чтобы описать эту яркость, используется тот метод описания цвета, о котором мы договорились.

Далее я убеждаюсь, что это описание будет реализовано, проверяя характеристики цветности закупаемой краски. Всё в порядке, моё предпочтение удовлетворено — против заданной в проекте насыщенности цвета инженер и финансист не возражают. Система в части цвета будет успешна (если не окажется, что есть ещё какая-то другая роль, которая, наоборот, добивается более бледного цвета и более изобретательна в реализации своих намерений).

Напомним, что описание (view) состоит из моделей, то есть отдельных маленьких описаний, отражающих какие-то важные характеристики, но не все характеристики, важные для суждения о каком-то аспекте системы, а только их часть. Метод описания состоит из мета-моделей (карта и легенда карты — это модель территории и мета-модель территории, мета-модель описывает модель, говорит о том, какие типы объектов и отношений могут появляться в модели). Мета-модели дают возможность создать модель, прогнозирующую состояние предмета моего интереса/характеристики системы. Создание viewpoints/«методы описания» — это работы мета-моделирования::метод (иногда говорят просто «моделирование», мета-модель это ведь тоже модель!).

Модели дают возможность оценить предмет моего интереса, и либо удовлетвориться, либо перейти к активным действиям по изменению ситуации, проверке того, что система действительно соответствует своим описаниям, или другим действиям, позволяющим скорректировать оценку в реализации предпочтений в сторону «ОК, успех, предпочтения удовлетворены». Модели — это

[410] https://en.wikipedia.org/wiki/Colorfulness

про описания с использованием методов описаний, знания которых включают мета-модели.

Иерархии моделей, мета-моделей, мета-мета-моделей посвящена онтология как метод мышления из состава интеллект-стека, но в нашем руководстве мы рассматриваем только особенности моделирования и мета-моделирования в рамках системного мышления — создание системных описаний, известное как **системное моделирование**. Этому будет посвящён следующий раздел нашего руководства.

Вопросы. Описания и важные характеристики

Царь послал в чужеземные края трёх мастеров, чтобы каждый из них описал заморскую самоходную тележку. Что должны спросить мастера, чтобы не попасть впросак?
--- спросить у царя, какой метод описания они должны выбрать ### Нет, они ещё не знают, какой их предмет интереса. А после того, как узнают, они сами должны выбрать подходящий метод.

+++ спросить у царя, что именно он хочет знать ### Да, без выяснения предмета интереса они не будут знать, что им в тележке описывать.

+++ опросить, кроме царя, всех бояр да дьяков - какой у них предмет интереса к тележке ### Да, обязательно! «Царь» не имеет ярко выраженного ролевого предмета интереса, это «ответственный», «начальник» — поэтому правильно опросить всех возможных содержательных экспертов, чтобы найти реальную проектную роль, стоящую за запросом начальника, и её ролевой/содержательный/предметный/профессиональный предмет интереса/важную для неё характеристику самоходной тележки.

+++ узнать друг у друга, какая специализация каждого мастера (сперва каждый мастер знает только свою) ### Да, ибо в зависимости от выявленного интереса в их команде может просто не найтись человека с нужным мастерством в методе описания (например, если интересуют магические свойства тележки, то должен присутствовать маг, а не механик, а если нужно перевозить инфекционных больных, то неплохо бы иметь в команде медика).

--- одно ли от них описание ждут, или три — от каждого мастера по одному независимому ### Нет, это не важно (особенно если ещё неизвестно, что именно описывать, не ясны предметы интересов для описания).

--- на сколько страниц должно быть описание ### Нет, это не важно (особенно если ещё неизвестно, что именно описывать). На сколько страниц нужно будет для удовлетворения интереса (понимания, что там будет с реализацией предпочтений в важных характеристиках) описывать, на столько и напишут — и число страниц мастера определят сами.

+++ спросить, кто будет читать привезённое описание ### Да, это крайне важно: это же будут как раз исполнители проектных ролей, у которых есть предметы интереса к самоходной тележке! Без знания этих предметов интереса будут привезены не те описания (на интерес к художественному дизайну привезут описания принципов самоходности, на интерес к самоходности привезут описания художественного дизайна, на интерес к экологичности привезут описание цены в эксплуатации, и т.д.).

--- есть ли где ещё другие такие же тележки ### Нет, мастера сами найдут «такие же» тележки — особенно, когда разберутся, что именно от этих тележек нужно, что делает их «такими же».

--- на пергаменте им описывать, на бересте или на бумаге ### Носители информации для документирования описаний обычно неважны, важна сама информация, само описание.

При проектировании источника питания для СВЧ модуля инженер должен уложиться в указанные конструктором габариты. Какой метод описания габаритов может использовать конструктор, передавая задание инженеру?
+++ Эскиз компоновки СВЧ модуля на салфетке, с указанием габаритов в см отдельной строкой ### Да, инженер сможет использовать этот документ для удовлетворения своего интереса к габаритам.

+++ Вылепленный из пластилина в масштабе 1:5 макет СВЧ модуля ### Да, инженер сможет использовать этот рабочий продукт/артефакт для удовлетворения своего интереса к габаритам. Метод описания физической моделью вполне подойдёт.

+++ Чертеж СВЧ модуля с источником питания, выполненный по ГОСТам ### Да, инженер сможет использовать этот документ для удовлетворения своего интереса к габаритам.

--- Показ габаритов модуля на пальцах ### Нет, габариты не будут документированы, метод описания не приемлем.

--- Продиктовал габариты модуля по телефону ### Нет, габариты не будут документированы, и метод описания и не указан, и не приемлем.

+++ Табличка с габаритами модуля в мм ### Да, инженер сможет использовать этот метод описания для удовлетворения своего интереса к габаритам.

Задания: мета-моделирование

Поставьте отметки о выполнении:

1. Опубликован пост со своими мыслями по итогам чтения материала и занятий моделированием текущего раздела руководства.
2. Проведены три беседы с коллегами, в которых обсуждался ваш рабочий

проект. Вы удерживали в голове системность как многоуровневость и многоаспектность описания, но не требовали этого от коллег. В ходе разговора вы отслеживали, какие упоминались типы из мета-мета-модели, излагаемой в нашем руководстве, но не называли эти типы вашим коллегам. Опубликован пост, в котором вы описываете, как вам удались эти беседы.

11. Системное моделирование

Описания и методы описания, модели и мета-модели

Описание/view каких-то аспектов системы состоит из одной или множества отдельных **моделей** (models). Тут, конечно, «состоит из» не отношение композиции для физических объектов, надеемся, что вы это уже отметили для себя. Часто поэтому слова «состоит из» заменяют уклончивым «группирует» (описание группирует модели), а в обратную сторону — «группируются в» (модели группируются в описание).

Модель — это набор каких-то объектов (физических или абстрактных/информационных), который ведёт себя в каком-то смысле похоже на поведение моделируемого объекта, то есть описывает этот моделируемый объект (необязательно моделируемым объектом будет система — можно моделировать что угодно, «объект — всё, что является предметом нашего рассмотрения»). Типы этих моделируемых объектов задаются мета-моделью и берутся из теорий/объяснений метода описания/моделирования (viewpoint). Модели могут быть самой разной степени формальности (вплоть до того, что самые неформальные описания в инженерных стандартах могут считаться «не-модельными описаниями», но мы так считать не будем).

Например, полное системное описание включает в себя «финансовое описание»::view предприятия::создатель (и между описанием и описываемым предметом отношение представления/representation, обо всём этого говорилось в руководстве по рациональной работе).

В финансовом описании можно в свою очередь выделить разные составляющие его модели, нужные для ответа на разные вопросы интересующейся финансами проектной роли:

- баланс (balance),
- отчёт о прибылях и убытках (profit and loss, P&L),
- денежный поток (cash flow).

Если у вас есть только баланс, то вы не ответите на вопрос о безубыточности работы предприятия, а если у вас отчёт о прибылях и убытках и даже баланс, то вы не сможете обсудить кассовый разрыв без наличия документов по денежному потоку. Одно описание для одной области интересов (с несколькими предметами интереса), три разные модели для трёх более узких областей, вплоть до отдельных предметов интереса (конечно, предметы интереса/важные характеристики могут уточняться!).

Все эти решения о том, как детально, широко или узко рассматривать какие-то важные характеристики системы — субъективны и являются результатом договорённостей в проекте. Но вот общие принципы достижения таких договорённостей, что обсуждать — вот это неплохо бы знать, чтобы достигать таких договорённостей быстрее.

Каждая из моделей должна быть выполнена с использованием одной из мета-моделей/meta-models/model kinds (это подробно обсуждалось в руководстве по рациональной работе, когда говорили об онтологических описаниях), причём каждая из мета-моделей устанавливает определенные языки, правила и прочие **соглашения о моделировании** (например, о том, каким объектам мы присваиваем какие типы мета-модели или мета-мета-модели и как будем выбирать имена объектов и имена типов из длинных рядов синонимов).

Так же как отдельные модели (models), отвечающие за уторговывание какого-то интереса, объединяются в более полное описание (view), отвечающее за уторговывание интересов в какой-то области интересов, так и мета-модели (metamodels/model kinds), определяющие язык, правила и приёмы моделирования, используемые при ролевых/тематических/видов описаниях, объединяются в **методы описания** (viewpoints), отражающие/frame предметы интереса (concerns) в областях интереса (area of concern). Методы описания с их мета-моделями и прочими соглашениями о моделировании описываются часто в специальном документе, который так и называется: «соглашение о моделировании». Часто для моделей с какими-то данными о реальном мире мета-модель называется «модель данных» (данные = модель системы, модель данных = мета-модель системы).

Например, для финансового описания/view создающей системы-предприятия нужно прежде всего выбрать один из «методов описания»/viewpoint — РСБУ (Российские стандарты бухгалтерского учёта) или МСФО (международные стандарты финансовой отчётности). При составлении баланса (одна модель финансового описания) и отчёта о прибыли и убытках (другая модель финансового описания) нужно использовать мета-модели (соглашения по тому, как делать модели) для этих видов моделей из какого-то одного метода описаний — либо РСБУ, либо МСФО.

И помним, что есть ещё и уровень мета-мета-модели (то, что РСБУ::viewpoint — это как раз использование наставлений из нашего руководства для присвоения типа, мы сказали, что РСБУ — это метод описания, использовали тип мета-мета-модели фундаментальной дисциплины, общий тип для всех дисциплин. А РСБУ представляет собой мета-модель, метод описания предметной/прикладной дисциплины).

Проще всего считать, что метод описания — это набор условных обозначений для многослойных карт-моделей, которые описывают территорию.

Напомним то, что вы уже знаете из руководства по рациональной работе:

- Карта — это не территория, а модель территории. Ходят не по карте, а по местности. Но выбираете вы маршрут обычно по карте. Другое дело, что набор условных обозначений (viewpoint) не даст вам карту (view), её придётся делать самостоятельно, заниматься моделированием какой-то местности (domain). А потом генерировать несколько маршрутов походов по этой карте, критиковать их («вау, этот маршрут ведёт в болото! Не пойдём туда!»), выбирать согласно какой-то из теорий решений наилучший способ действий, наилучший маршрут (лучший метод работы), а затем ещё и заставить себя (да и уговорить других) пойти по этому маршруту, то есть кроме выбора метода ещё и сделать саму работу по методу. Это всё метафора, конечно, но вам нужно разобраться в этой метафоре. В любом случае, ваша мысль должна дотягиваться до реальной работы, для которой вы должны определить метод (стратегировать: понять метод в ситуации,

когда метод неизвестен), в каждом методе есть его предметы метода из его предметной области, они даны в какой-то модели, а эта модель имеет свою мета-модель.

- Есть несколько уровней абстрактности, они же мета-моделирования, от самой общей модели теории понятий (theory theory — «всё есть объекты и отношения», это можно считать foundational ontology, не путать с «foundation model»/LLM) через мета-мета-модель как набор типов для объектов и отношений из фундаментальных методов мышления (в том числе набор типов системного подхода — «система», «окружение», «системный уровень», «создатель», и т.д.), затем мета-модель какой-то предметной области (в том числе менеджмент с его типами: орг-архитектура, органиграмма, выпуск/проход/throughput и т.д.), затем модель с её уже значениями (например, выпуск/проход 3 сепульки в день на $30). Конечно, этих уровней может быть и больше (например, можно выделить метаУ-модель, метаС-модель). По факту, есть территория (мир), карта (модель), мета-модель (карта карты, то есть система обозначений для domain), есть мета-мета-модель (карта карты карты, то есть система обозначений для системы обозначений карты, универсальная для всех систем обозначений, то есть для всех domain), и есть даже мета-мета-мета модель (система обозначений для системы обозначений для системы обозначений карты). В системном мышлении вы должны это всё удерживать и не путать. Если вы вдруг заговорите терминами мета-мета-модели (словами из нашего руководства), то вас никто не поймёт. Обсуждайте не системы обозначений, а карту. Но сами вы в голове должны удерживать то, откуда пришли обозначения на карте, и откуда пришли обозначения того, что дано в таблице обозначений.
- На любом из уровней абстракции вы можете или моделировать строго (карта в масштабе, с точными обозначениями в какой-то проекции) или не очень строго (детская карта, карта метро — где важно донести плохо понимающим в картографии людям какую-то одну идею, а не иметь точную модель). В моделировании это выбор уровня формальности — от формальности с учётом унивалентных оснований математики (они будут построже, чем основания математики, построенной на теории множеств, и подразумевают конструктивизм — даже не theory theory с объектами и отношениями, а пока плохо определённую и крайне нераспространённую фронтирную теорию понятий с объектами и операциями их конструирования, подробней это затронем в следующем разделе нашего руководства) до формальности былин и эпосов, когда вы от мета-мета-мета-модели (foundational ontology) на базе объектов и отношений вдруг переходите к рассуждениям по аналогии (скажем, прототипной теории понятий, когда фрукты определяются как «разные варианты яблок, всё более и более на яблоки не похожие»), concept blending и прочей уже не логике, а чуть ли не лингвистике. Люди всё-таки в основном используют быстрый и лёгкий режим мышления S1, а не трудный и неудобный режим мышления S2. Рассуждения в S1 трудно критиковать, они «аналоговы», а вот рассуждения в S2 критиковать легко — там работает логика, и там либо вы правы, либо ошиблись. Если вам важна точность, то надо быть формальным: на неточных рассуждениях ракету не построишь, подлодка утонет, предприятие разорится. Но если вы будете только строгим логиком, то вы не сможете разговаривать с «простыми людьми».

В любом случае, **нельзя ничего описывать, если ты не осознал, как именно**

ты описываешь, каким способом описания! Нельзя понять чьё-то описание, если ты не осознал, как именно, каким способом описания это чьё-то описание сделано!

Нельзя делать карты и/или давать рассматривать другим проектным ролям карты, если изготовителю карты и/или другим проектным ролям неизвестна легенда карты и методы картографирования. Карты, сделанные по разным методам описания, потом нельзя сопоставлять друг с другом. Нельзя делать карту, используя слои, подготовленные согласно разным методам картографирования — нельзя брать подготовленную геодезистами карту водных ресурсов города и подготовленную службами метрополитена карту-схему станций метро и просто накладывать их друг на друга. Нельзя брать баланс по РСБУ, а отчёт о прибылях и убытках делать в МСФО. Все модели (models) из описания (view) должны быть подготовлены с использованием мета-моделей из одного метода описания (viewpoint). И да, метод описаний нужно согласовывать с проектной ролью, ибо в зависимости от предмета интереса и предпочтений роли в характеристиках этого предмета интереса разные проектные роли будут требовать использовать разные методы описаний! Например, финансисты из бухгалтерии будут требовать финансовые описания по МСФО (и у них важным будет описание себестоимости), а операционные менеджеры будут отказываться от таких описаний и настаивать на ведении собственного финансового учёта, потому что они будут считать «себестоимость» вредной и сбивающей с толку характеристикой, настаивая на совершенно других методах финансового расчёта, более пригодных не для целей налогообложения, а для целей операционного менеджмента. Они могли бы настаивать на использовании методов описания, принятых в «учёте прохода»/«throughput accounting» из теории ограничений Голдратта[411].

Вопрос, а как же тут быть — если всем нужны какие-то разные описания? Как быть, например, в случае того же финансового учёта (для управленцев, pre mortem) и бухгалтерского учёта (для налоговой инспекции, post mortem?). Конечно, надо просто делать несколько разных описаний! Описываемому предмету никак не помешает, если вы сделаете его фотографии с разных ракурсов, разными фотокамерами, а иногда ещё и видеосъёмку, а ещё и просветку в рентгеновских лучах, а потом ещё добавите описание текстом в стихах.

Так, управленцы могут организовать СебеУчёт по Голдратту, а ТебеУчёт1 по РСБУ (для своей налоговой) и ТебеУчёт2 по МСФО (например, для акционеров). Это ОК, это нормально, так и надо делать — только приглядывать за тем, чтобы описывался один и тот же объект (деньги предприятия и их движение), а не разные. Иногда говорят о правилах соответствия (correspondence rules) между объектами разных описаний, но эти правила соответствия не всегда возможно определить, тут не надо обольщаться: соответствий между фотографией кипящего чайника в инфракрасном свете и ведомостью комплектующих для изготовления чайника может не обнаружиться. Но вот в разных инженерных моделерах могут встретиться разные значки для отображения функционального объекта — и тут можно указать, что есть соответствие/correspondence, например, между какой-нибудь functional block из DIN 66001 и functional module из IEC 61346. Конечно, для того, чтобы обнаружить такое соответствие, вы должны подняться на мета-уровень по отношению к этим стандартам: понять, что все эти объекты — функциональные объекты, а не

[411] https://en.wikipedia.org/wiki/Throughput_accounting

конструктивные или объекты размещения (места).

Метод описания чаще всего бывает **библиотечным** (library), это означает, что вместо раскрытия его содержания приводят просто ссылку на литературу по этому методу. Это всё равно как вместо полного текста и картинок легенды карты можно было бы просто дать ссылку на книгу или картографический стандарт, где приведён список условных значков и подробный текст о том, как использовать эти значки для изображения деталей рельефа, флоры, фауны, полезных ископаемых, плотности населения и т.д. Но если нет ещё текста с готовым методом описания, то вам придётся описывать какой-то предмет интереса таким способом, который до сих пор не использовался. Тогда вместо этой ссылки на библиотечный метод описания вы обязаны к описанию приложить документированный вами самими метод описания: без указания метода описания сами описания делать нельзя. Метод описания — это альфа, так что для свидетельствования состояния этой альфы потребуется артефакт/рабочий продукт/документ. И описание требует документирования, и метод описания требует документирования.

Главное — это запомнить: **любое описание — это описание системы, любое описание системы сделано с использованием метода описания** (даже если описывающий этого не осознаёт), доступно людям описание становится только после его **документирования**, метод описания — это описание описания (поэтому уместны вопросы и про метод описания метода описания, и про документирование метода описания метода описания!).

Метод описания (viewpoint) оформляет (frame) какую-то важную для выполнения ролью какого-то метода работы характеристику (concern этой роли). Одно из следствий: **если у проектной роли/stakeholder нет соответствующей «озабоченности»/concern, то описание/view для удовлетворения интереса к этой отсутствующей характеристике не делается: описания, в которых не затрагиваются важные для методов работы каких-то ролей характеристики, не нужны, они избыточны.** Соответственно, это описание не документируется, и метод описания для него тоже не нужен.

Даже если какой-то используемый вами регламент или стандарт (который вроде бы надо исполнять) требует какого-то описания, но вы обнаружили, что никто его никогда читать не будет (например, менеджеры просто проверят факт наличия описания «для соответствия стандарту», но даже в случае реализации каких-то рисков никто в эти описания не полезет разбираться, их не откроет и не использует для разбирательств по существу) — такое описание не нужно делать, это бессмысленная трата ресурсов проекта. И наоборот: **если проектной роли важна какая-то характеристика/«предмет интереса»/concern и она намерена её обсуждать с другими проектными ролями, или даже использовать для собственного мышления и принятия ролевых решений, то описание с включением этой характеристики делается и обязательно документируется.** Речь никогда не идёт об устных ответах на вопросы. На бумаге, или в базе данных, или в файлах, но описание должно быть документировано.

Скажем, вы приняли какое-то решение на совещании. Это решение — описание будущей ситуации, какого-то состояния каких-то объектов. Не записано (и не в протокол, ибо протоколы не выживают больше пары недель, дальше они забываются) — значит решение не принято, а просто «обсуждено». Решение в письменном виде наверняка будет иным, нежели всем «послышалось» как оно

звучит устно, там будут обнаружены ошибки, несоответствия другим ранее принятым решениям и т.д. Описания — это отнюдь не только предписанные в каком-то инженерном или менеджерском стандарте описания! Надо очень жёстко отличать системы (и ситуации, то есть системы в каких-то состояниях, учитывающие происходящие с ними изменения состояний) и описания систем, а также выполнять все требования к описаниям, какие бы они ни были. Например, решение на совещании: если это описание (view), то там обязательно должен быть и метод описания (viewpoint). Что именно вы описываете, какие типы основных объектов будут в вашем описании решения совещания?

Подходы к описаниям (views) согласно ISO 42010:2022 могут быть двух видов: прожекторные (projective) и синтетические (synthetic).

Прожекторные описания — это как в театральном прожекторе, в котором лампа белого цвета, но мы делаем цветной луч, просто отфильтровывая все цвета кроме того, который нам понравится. По факту это означает, что у нас есть большая база данных, в которой хранятся все связанные между собой разные модели разных ролевых описаний — все вместе, соответствующие какой-то модели данных. Эта модель данных — общая онтология самых разных предметных областей. Когда нам нужны данные одной модели (model), то просто из этой совместно хранящейся одной базы данных как документа с системным описанием моделей системы для самых разных мета-моделей, которые увязаны типами одной мета-мета-модели отфильтровывается только то, что нужно в части мета-модели (запрошенные ролевые описания, или даже запрошенные отдельные модели из этих описаний) и отображается в подходящих форматах документов. Если мы смотрим на какие-то отдельные классы компьютерного корпоративного софта (ERP, CRM, PLM и прочие «трёхбуквенные аббревиатуры»), то внутри каждого такого софта можно обнаружить базу данных, которая пытается воплощать в себе прожекторное описание. Если хочется получить какое-то отдельное описание с отдельными характеристиками системы, то надо просто выполнить какой-то запрос на языке запросов к этой базе данных.

Синтетические описания — это когда наоборот: исходные ролевые описания даны в виде отдельных документов моделей, причём каждая модель документирована не просто как часть одной общей для всех моделей базы данных (как в прожекторных описаниях), а отдельный бумажный или электронный документ в отдельной онтологии. Между этими автономными моделями из отдельных документов по возможности устанавливаются **правила соответствия** (correspondence rules, «этот объект этой модели — это вон тот элемент вон той модели»), и общая модель системного описания тем самым получается синтетически, объединением отдельных автономных видов моделей. Рассуждения про системные/полные и ролевые/аспектные описания системы при этом не меняются от того, как собираются эти описания из документации моделей — сразу в общем документе (прожекторный подход к описаниям) или после создания отдельных документов моделей (синтетический подход). Нельзя сказать, какой метод побеждает:

- Прожекторный метод очень удобен: всё собрано в одном месте, да ещё и согласовано. Проблем типа «вот тут у меня длина в футах, а у вас в метрах, а вот тут у них в попугаях» не возникает, мета-моделирование централизовано!
- Это кошмар для архитекторов: чтобы чуть-чуть пошевелить какое-то описание, надо шевелить сразу все описания вместе, поддерживать это

согласование! Прожекторный метод становится «бутылочным горлышком», запрещая по факту автономную работу команд, занимающихся разными описаниями.

- Если у вас уже есть автономные моделеры, удобные для документирования каких-то отдельных описаний в рамках синтетического подхода, очень трудно заставить от них отказаться, чтобы перейти на новые моделеры, которые хранят свои модели в общей базе данных в рамках прожекторного подхода — усилия по переучиванию людей и переналаживанию инструментария обычно считают не слишком оправданными, даже если постоянно сталкиваются с какими-то коллизиями между моделями. Конечно, количество коллизий заведомо больше при синтетическом подходе.

Синтетические и прожекторные описания оказываются эквивалентными в плане мышления о работе с описаниями. В ISO 42010:2022 явно указывается, что все основные идеи по работе с описаниями (например, требование указания viewpoint для любого view) работает для обоих вариантов. Просто в синтетических описаниях приходится много сил тратить на согласование независимо подготовленных описаний, «интеграцию данных жизненного цикла». А в прожекторных описаниях приходится много сил тратить ещё до начала моделирования конкретной системы на то, чтобы согласовать методы описания для документирования всех аспектных/частных описаний в одной базе данных. В общем случае, конечно, побеждают синтетические описания, поскольку они не подразумевают предварительной огромной работы по интеграции инструментов моделирования. Но желание иметь прожекторные описания, с которыми потом удобно работать, достаточно сильно. Поэтому современные проекты имеют несколько видов прожекторных описаний, вместе составляющих синтетическое (ибо ни один из видов прожекторных описаний не отражает все проектные предметы интереса).

Как увязывают описания в каком-то большом проекте? Обычно используют идеи из стандарта ISO 11354-1:2011, framework for enterprise interoperability (он будет помянут в подразделе «Платформы и инструментальные стеки» нашего руководства, а затем в руководстве по системной инженерии в подразделе «Интеграция данных»):

- Интеграция: есть общий метод описания, которому следуют все участники проекта, это прямая отсылка к прожекторному подходу. Все части проекта в момент создания согласовываются между собой. Дорого, долго, но гарантированно работает! Это вариант «в нашем проекте все говорят на окситанском языке[412], другие языки запрещены. Кто начинает у нас работать, должен выучить окситанский язык».
- Унификация: согласовываются только описания, которыми обмениваются подсистемы (например, подразделения в организации или космический корабль с ускорительным блоком), а внутри каждой подсистемы описания более-менее автономны/инкапсулированы. Прожекторный подход в каждой подсистеме, далее объявленная публично модель взаимодействия — и прожекторный подход в системе в целом. Это вариант «вы все говорите как хотите, хоть по-испански или по-английски, или вот тут у нас три китайца говорят на мандаринском, но рабочий язык у нас будет — окситанский, совещания только на нём, словарь и учебник опубликованы в стандарте».

[412] https://ru.wikipedia.org/wiki/Окситанский_язык

- Федерирование: увы, в этой ситуации согласовать заранее ничего нельзя. Скажем, мы пытаемся согласовать описания, которыми обмениваются системы, разработчики которых ничего не знали друг о друге. Это будет соответствовать синтетическому подходу. В жизни это будет два варианта. Обычный: «вот тут надо наладить общение между китайцем и индийцем, давайте найдём мандаринско-хинди переводчика». Жизнь становится ужасной при этом варианте, начиная с буквально трёх общающихся: для них нужно будет найти трёх переводчиков, которые будут находить соответствия и переводить с одного языка на другой (да ещё часто это надо делать в обе стороны). После того, как эта проблема в проекте обнаружена, предлагается какой-то из этих языков сделать «нейтральным рабочим», например, английский (даже если на этом языке никто из участников проекта не говорит, но совещания теперь будут вестись на нём). Конечно, это попытка перейти от федерирования к унификации. При этом потребуется только переводчики с «рабочего английского» на язык каждого участника совещаний, но в больших проектах договориться о том, будет ли это «рабочий английский», «рабочий мандарин», «рабочий окситанский» — это довольно трудно.

Отдельное замечание — это мы говорим пока только об описаниях в локальных представлениях. Описания в распределённых представлениях (например, в виде больших языковых моделей в AI-агентах, или в виде просто нейросетей, если речь идёт о каких-то простых описаниях производственных процессов в непрерывном производстве, например, химическом производстве или генерировании электроэнергии) тут мы не рассматриваем. Но это не означает, что их нет и их можно не учитывать. Если рассматривать эти описания, то они сделаны тоже каким-то методом (например, методами глубокого обучения[413]). В этой области пока нет какого-то понимания методов работы с такими описаниями, но она довольно бурно развивается. По большому счёту, работа с описаниями в человеческой голове тоже проходит в распределённых представлениях. Лучшая практика тут — попробовать отмоделировать нужные части этих описаний при помощи локальных представлений. Тут работает ход, аналогичный вышеописанным ходам с интегрированием, унификацией и федерированием: внутри каждого такого описания может быть что угодно, но для коммуникации надо будет перейти к локальным представлениям — и тогда будет работать всё, что известно про системные описания, данные в локальных/символьных представлениях.

Моделирование: цепочки классификации (операционная модель, мета-модель, мета-мета-модель)

Приведите примеры цепочек по отношению мета-моделирования для объектов из вашего проекта, по образцу в первой строчке. Не слишком заботьтесь о различении функциональных и конструктивных объектов и типов для них. Важно, чтобы вы различали реальный объект из физического мира проекта, его тип из дисциплины в предметной области проекта и тип дисциплины из интеллект-стека фундаментальных методов мышления (тип из нашего руководства) — и кроме различения ещё и понимали их соответствие.

Операционная модель	Тип из мета-модели	Тип из мета-мета-модели

[413] https://en.wikipedia.org/wiki/Deep_learning

Ноутбук Dell XPS 13 2020 с серийным номером 12345	Экзокортекс для мышления письмом	Инструмент метода «мышление письмом»

Системная модель, мета-модель, мета-мета-модель

Моделирование в системном мышлении — это главное средство борьбы со сложностью в проектах. Большая сложность — это когда чересчур много деталей в описаниях, поэтому мозгу (и компьютерам, ему помогающим) нужно делать много вычислений. Число вычислений можно резко сократить, если выкинуть вычисления над неважным.

Сходу обычно непонятно, что важно описывать, а что не так важно. Тут всё просто: важные объекты (экземпляры) предметной области задаются методом описания этой предметной области (мета-модель domain, онтология предметной области), а важные объекты для работы с методами описаний предметной области задаются мета-мета-моделью, общей трансдисциплинарной/фундаментальной онтологией методов мышления (её часто называют upper ontology, наше руководство раскрывает часть понятий этой онтологии).

Моделирование — это и есть описание только самого важного, и опускание при описании неважного. Хотите бороться со сложностью — мыслите моделированием!

Документирование моделей позволяет не просто описать важное, но и удержать важное во внимании столько времени, сколько нужно для рассуждений. А неважное во внимании не удерживается, остаётся только в разговорах в ходе моделирования, когда вы обсуждаете метод моделирования и высказываете догадки о том, что должно попасть в модель.

Собранность и моделирование идут рука об руку, биологическому мозгу в моделировании не верим, все результаты моделирования записываем! Мышление моделированием — это не «ментальный акт», это просто явное использование какого-то метода описания в ходе мышления письмом. **Не пишешь — не моделируешь, не думаешь!**

Для моделирования на каждом уровне абстракции нужно знать три вида объектов:

- **Моделируемый объект** из ситуации в реальной жизни (как-то связанные между собой предметы в какой-то предметной области/domain). В конечном итоге это какие-то предметы физического мира, о них известно что-то кому-то важное, и известно что-то совсем неважное. Может быть и какой-то граф моделирования, и тогда речь идёт о каких-то моделях других моделей (моделирование моделей тоже вполне возможно! Одна модель позволяет тогда вынести суждения о поведении какой-то другой модели в каких-то важных аспектах. Например, вы можете составить таблицу, описывающую города и их население, а информацию для этого взять из какой-то географической карты, где эта информация есть — если вам не нужно взаимное расположение городов, а нужна только информация о населении).

Информации много, объектов внимания много, задача моделирования как раз справиться с компактификацией мышления: вычислитель агента (мозги и компьютеры, а хоть и трудового коллектива), который действует в предметной области, перегружается обработкой маловажной информации. Такие тонкости, как различия реальности и действительности тут в расчёт не берём. Реальность — это всё, что есть, а действительность — это всё, что мы знаем о том, что есть в реальности и как-то можем описать, поэтому воспринимаем мы не всю реальность, а только действительность. Разбирательство с предметной областью (самими объектами, а не их описаниями!) и есть предмет онтологии, ответ на вопрос «какие объекты есть в мире» (как нарезан физический мир, а также ментальный/абстрактный мир на объекты, почему на эти, а не на другие). Скажем, вы моделируете какие-то фрукты: пишете, что на столе есть яблоко-1, яблоко-2 и единственный банан, который мы дадим с номером, чтобы подчеркнуть его «экземплярность», банан-1. А тарелки, салфетки и всё прочее остальное игнорируете. Вот эти физические два яблока и банан — они и есть предмет моделирования.

- Типы объектов, которые нужно найти в мире. Они задаются методом описания, который содержит мета-модель. Эта **мета-модель** задаёт типы объектов, которые нам важны в предметной области. Если мы описываем стол, определяя на нём фрукты, то это задание типа «фрукт» и видов фруктов (специализаций фрукта) «яблоки» и «бананы». Есть и мета-мета-модель с типами, которая позволяет найти объекты самой мета-модели. Так, банан::фрукт::«физический объект». Мета-мета-модель позволяет рассуждать о мета-моделях: сравнивать их, объединять, дополнять, уточнять, проверять по ним рассуждение и вообще обсуждать то, какая мета-модель сейчас нужна для описания моделируемого объекта.
- Сама **модель** (иногда для определённости её называют «операционной моделью» или «моделью экземпляров»). В случае физической модели это какие-то предметы, сохраняющие в себе основные характеристики, задаваемые мета-моделью, а в случае информационной модели — это какие-то объекты описания, типы которых задаются мета-моделью. Важно, что модель тут появляется в обсуждении только после того, как мы определили мета-модель. Другими словами: описание появляется только после того, как мы определили кусок реальности, который хотим моделировать, далее задали типы объектов, которые хотим в реальности найти — и только потом создаём и документируем модель. В нашем примере мы напишем строчки символов «яблоко-1» и «яблоко-2», а также «банан-1» — но это уже объекты в мире описаний. Более точно, конечно, было бы писать «яблоко-1»::яблоко::фрукт, «яблоко-2»::яблоко::фрукт, банан-1::банан::фрукт.

Вот эта тройка объектов есть на каждом уровне моделирования. Ход мышления от моделируемых экземпляров в мире к модели, далее мета-модели и далее мета-мета-модели и так далее называют «мета-моделированием». Скажем, берём мета-модель и для каждого её типа указываем тип мета-мета-модели, более абстрактный — это мета-моделирование. Берём наоборот — мета-мета-модель и для её типов создаём модель предметной области, мета-модель, это будет шаг на моделирование.

Запоминать надо этот принцип мета-моделирования (от мира ко всё большим уровням абстракции в типах) и моделирования (от типов с самыми большими уровнями абстракции к меньшим), ибо число уровней мета-моделирования (число

уровней иерархии по отношению классификации) может быть очень и очень разным. Как и в любой иерархии, на каждом её уровне есть уровень выше и есть уровень ниже, тип отношения должен сохраняться во всей иерархии без изменения, а на крайних уровнях надо быть крайне осторожными в рассуждениях.

Например, «крайний уровень» может оказаться не крайним даже для upper ontology, выше него пойдут уровни объектов разных теорий понятий. А внизу иерархии может быть переход не по отношению классификации, а по отношению представления между моделью и моделируемым объектом.

Как эту троицу «моделируемый объект, модель, мета-модель» обнаружить в жизни? Очень просто: в описаниях обращайте внимание на заголовки и их типы в каких-то человекочитаемых форматах, а в машинночистаемых форматах — на модели данных. Обычно:

- Моделируемый объект — он где-то в мире, в описании отсутствует. Например, эксельный документ, который описывает технико-экономический проект/design сепульки на 8 Ватт: сама сепулька будет в мире, а в технико-экономическом проекте только её описание.
- То, что озаглавлено — относится к модели. Например, будет указано, что этот документ — технико-экономический проект именно сепульки, причём мощность сепульки — 8 Ватт.
- Но вот «технико-экономический проект» а также «мощность» (и единица измерения мощности) — это будут элементы мета-модели. Можно сделать множество моделей с этими типами. Грубо говоря, «форма документа, просто заголовки с незаполненными полями/реквизитами» (иногда её называют шаблоном, иногда бланком, иногда «пустографкой») — это и есть мета-модель, эти заголовки и есть типы мета-модели.

Где искать мета-мета-модель мира (типы объектов из фундаментальных методов мышления, которые мы и изучаем в наших руководствах)? Очень редко эти типы явно вписывают в описания, но чаще всего — нет, их «додумывают в уме», прежде всего в уме составителей форм/шаблонов/пустографок. Так «технико-экономический проект»::описание, и то, что речь идёт не о самой сепульке, а о её описании «технико-экономический проект» — это приходится удерживать в уме, как-то определять «из контекста». То, что «важная характеристика»/«предмет интереса»/concern для пользователя сепульки — мощность в ваттах, («мощность в ваттах»::concern), это тоже «додумывается». В некоторых базах данных мета-мета-модель надо указывать явно. Но всё-таки в большинстве случаев её не указывают.

Системное мышление (и другие методы мышления из состава интеллект-стека) заставляют удерживать внимание не только на трёх уровнях (объект в жизни, модель как значение в форме, мета-модель как заголовок значения, например заголовок реквизита в форме или колонки в таблице или даже заголовок документа, но и добавляют четвёртый, «в уме» — тип мета-мета-модели из методов мышления. С этого момента вы знаете, какие мыслительные операции можно делать с этими объектами.

Помним: если A::число, то умножать число на число можно, но разбить молотком — нельзя. Если A::орех, то разбить молотком можно, но умножать орех на орех — нельзя! Типы уберегают от ошибок и позволяют договариваться о важном. Вот мета-мета-модель и говорит нам о реально важном, что важнее даже заголовков (мета-модели) в формах/шаблонах описаний!

Как определить, что важно? Важно в описаниях только то, что помогает работать по каким-то методам (в том числе и по методам описания, но только «в том числе»). Если вы рубите::метод лес::«предмет метода», то вам описания красоты леса, оттенка зелёного в листьях деревьев, количества муравьёв на гектар леса — все они не нужны, это не надо моделировать. Но надо моделировать толщину деревьев, количество подлежащих рубке деревьев на гектар, просеки для вывоза срубленных деревьев и т.д.

Для разных проектных ролей важными характеристиками в системе являются разные характеристики, и двигать их значения разные проектные роли хотят в разные стороны, и им для этого нужно знать значения многих других характеристик (в системах же всё связано, части системы взаимодействуют, поэтому значения всех характеристик взаимоувязаны, все роли это понимают). То, что важно для какой-то роли в выполнении её метода работы, задаётся мета-моделью предметной области этого метода из теории/знаний/дисциплин/объяснений/алгоритмов метода. Предметная область — это предметы какого-то метода, важны не предметы сами по себе, а состояния этих предметов и возможные с ними операции, переводящие эти предметы из одного состояния в другое.

Мета-модель для каждого уровня есть всегда (так, мета-моделью для прикладной предметной области является набор типов фундаментальных методов мышления). Чтобы не запутываться, мы смотрим на иерархию мета-моделей по отношению классификации и просто добавляем для каждого отношения «мета-», чтобы сохранить не столько относительное наименование уровней (моделируемый объект, модель, мета-модель — на каждом уровне абстракции/классификации), сколько абсолютное (моделируемый объект, модель, мета-модель, мета-мета-модель, мета-мета-мета-модель, но это может быть организовано и как моделируемый объект, модель, метаУ-модель, метаС-модель, мета-мета-модель и т.д., или ещё каким-то образом). Помним при этом, что число уровней мета-моделей может существенно различаться от проекта к проекту, поскольку могут быть приняты самые разные соглашения о моделировании, уместные в контексте конкретного проекта. Так что надо разобраться с принципом рассуждения из этого абзаца, а не пытаться просто скопировать его в своём проекте. Мета-моделей (и даже мета-мета-моделей) может быть много, и даже уровней «мета-» может быть много.

Часть мета-моделей в иерархии мета-моделирования определяется аппаратурой мозга человека или аппаратурой компьютера, воспринимающего мир. Прежде всего речь идёт об аппаратуре для foundational ontology как теории понятий[414], определяющей высокие уровни мета-моделирования, какие вообще могут быть понятия в моделях всех уровней абстракции. Но и моделируемые объекты тоже могут зависеть от аппаратуры. Скажем, если у нас нет способа измерить напряжение и ток, то мы не можем их хорошо отмоделировать (понятия вольта и ампера появились в физике относительно недавно).

Важность какого-то объекта в мире, достойного для него отдельного понятия определяется не в один шаг моделирования, а во множество шагов: даже то, что в мире есть объекты и отношения, оказывается результатом моделирования — мета-мета-мета-моделью, или foundational ontology. Нужно быть крайне аккуратным в том, сколько там уровней моделирования, и дошли ли мы до последнего уровня, или где-то остановились (обычно, если дошли — это называем «онтология», а если

[414] https://plato.stanford.edu/entries/concepts/

остановились достаточно низко на мета-модели какой-то предметной области и не слишком были озабочены тем, какие там типы мета-мета-модели — это называют «онтика», эдакая «недоонтология», ибо игнорируется полная иерархия мета-уровней и согласованность с ней всех моделей). Тут, как всегда, онтология как изучение разных онтологий и онтик и конкретные онтологии или онтики обозначаются одним словом (как геометрия и, например, геометрия Римана или логика и логика Тарского).

Конечно, в онтологии как учении о существующих в мире объектах (точнее, типах объектов), есть множество школ, которые пользуются разной терминологией. Так, «мета-мета-модель» из нашего руководства, следующего инженерной традиции мета-моделирования, в специальной литературе называется upper ontology (объекты, описываемые в фундаментальных теориях/знаниях интеллект-стека фундаментальных методов мышления, то есть описываемые в трансдисциплинах, например, понятие «система» как раз оттуда), а мета-модель называется middle ontology или прикладной онтологией (это моделирование предметной области какого-то прикладного метода работы, например, машиностроения как метода создания «машин» — транспорта, станков и другого оборудования, задействующего механические силы). Модель экземпляров/операционная модель — это просто «модель» каких-то экземпляров объектов, а не типов объектов. Но также «моделью» называют и любую из мета-моделей, легко запутаться. Тем не менее, уровней «мета-», то есть уровней задания типов для типов (уровней абстракции, уровней понятий в иерархии по отношению классификации) обычно хватает трёх-четырёх.

Всё это подробно обсуждалось в руководстве по рациональной работе. Мы ещё несколько раз повторим это рассуждение, ибо оно важно: вы обращаете своё внимание не на произвольные объекты в мире и других моделях, которые вам «бросаются в глаза» (они ведь могут быть как важными, так и неважными. И неважные объекты могут сильно отвлекать внимание от важных!). Вы обращаете внимание на важные объекты (даже специально их ищете, а если речь идёт о проектировании, то специально предлагаете), но сам факт важности существования этих объектов задаётся мета-моделью, которая в этот момент у вас есть. А если мета-модели у вас нет, то вы ищете её объекты точно так же: обращая внимание на те объекты, которые задаются мета-мета-моделью, ибо это рассуждение работает на любом онтологическом уровне. Если вы ищете что-то важное в мета-мета-модели, то у вас в голове в этот момент есть мета-мета-мета-модель (теория понятий), которая по факту мета-модель для мета-мета-модели. Не знаете теории понятий (скажем, theory theory — что мы представляем мир главным образом в объектах и их отношениях, а не в прототипах) — не сможете разобраться в мета-мета-модели (материале нашего руководства). Не знаете мета-мета-модели — будете тупить при разбирательстве с онтиками предметных областей.

Важность задаётся многоуровнево, мы выхватываем своим вниманием не какие-то случайные объекты. Образование (обучение мета-моделированию в частности) нужно для того, чтобы узнать, какие именно важные объекты мы ищем в мире или (в случае придумывания чего-то нового) в своём воображении. Образование нужно для экономии времени на вычислениях по неважным поводам! **Образование нужно в том числе и для того, чтобы понять, зачем оно нужно (как ни странно это звучит).** Например, нужно образование, чтобы понять: без знания онтологии (как мета-моделировать на многих уровнях) никакого системного

мышления не будет, ибо вы не сможете осознанно строить системные описания. Вот буквально следующий абзац содержит описание реальной ситуации: «Финансист хочет предотвратить кассовый разрыв». Разметим типами из нашего руководства.

Финансист::роль хочет предотвратить::предпочтение/интерес «кассовый разрыв»::предмет интереса. Из текущего раздела руководства мы знаем, что нужно обязательно сделать (и документировать!) описание/модель, которая будет отражать этот самый кассовый разрыв. Это модель денежного потока, она будет отвечать на вопросы по этому предмету интереса. Всё остальное даже из финансов (баланс, прибыль и убытки) при этом не так уж важно, другие модели из других описаний будут использоваться для ответов на другие вопросы этой же, или другой проектной роли. И пока вы не вытащите в переговоры это описание, финансист со своим интересом будет не удовлетворён, как вы его ни будете уговаривать «успокоиться». Нет, пока не отмоделируете денежный поток (или он сам не отмоделирует его и не убедится, что кассового разрыва не будет), финансист не успокоится.

Деятель с системным мышлением позаботится о том, чтобы в работах по проекту создания системы была предусмотрена и работа менеджера, в том числе по методам работы с финансами, в частности по методам предотвращения кассового разрыва. Поэтому для организации создателя будет выполнено финансовое описание с включением в него документированного (записанного на каких-то носителях информации, а не просто выполненного «в уме») описания денежного потока, и эта модель будет отвечать на вопросы финансиста в этом предмете интересов, а также служить основой для разговора финансиста с другими ролями (это будут разговоры разных агентов и их компьютеров, а то и разговоры самих компьютеров с AI, а в небольших проектах это будет разговор разных ролей в одной голове и/или компьютере, если агент играет сразу множество ролей).

Почему надо так сложно? Потому что вы можете просто тупо запомнить: «финансисту нужно иметь описание денежного потока, чтобы он предотвращал кассовый разрыв». Но дальше вам нужно будет запомнить тысячи подобных утверждений — и голова лопнет. Вместо этого вы запоминаете способ рассуждения, и в каждой конкретной ситуации просто понимаете, какие виды описаний нужны (и их нельзя не делать), какие виды описаний не нужны (и их наоборот, нельзя делать — только ресурсы зря тратить), а ещё как именно делать эти описания, чтобы в них было меньше ошибок. А если ситуация будет такова, что вы в непонятной роли начинаете работать непонятным методом, то вы сможете разработать для себя новые виды описаний, ибо будете понимать, как думать об «описаниях вообще» (например, вы будете знать, что в описаниях — модель и мета-модель в заголовках, а мета-мета-модель надо «приговаривать» к заголовкам, и тоже удерживать её во внимании).

А теперь повторим всё это ещё раз. Увы, онтология не самый простой предмет в интеллект-стеке фундаментальных методов мышления, поэтому повторение может оказаться не лишним. Впрочем, надёжное усвоение начинается где-то с пятидесятого повторения, поэтому повторений в тексте руководства и даже всех руководств программы рабочего развития инженеров-менеджеров будет недостаточно, надо будет все эти рассуждения про системное моделирование многократно повторить в рабочих проектах. Беглость у вас появится после создания пятидесятой таблички (если это моделирование в табличках), где вы проследите, чтобы у вас там был озаглавлен тип объекта описания самой табличкой, колонки

таблички соответствовали мета-модели, вы явным образом проговорили типы для каждой колонки, значения модели были бы в каждой строчке таблички, а сами экземпляры моделируемых предметов были в жизни (физичны или брались из других описаний) и соответствовали строчкам таблички. Это не так долго: по три таблички в день через день — и можно справиться за месяц! Чудес не бывает, просто от прочтения наставлений руководств моделировать вы не научитесь. Учебник езды на велосипеде помогает научиться ездить на велосипеде, но без реальной езды — ничего не получится.

«Предмет интереса»/concern/«важная характеристика» в описании/модели (view/model) оформляется/framed «методом описания»/viewpoint/meta-model, а в этом методе описания как раз указывается, что самое важное в моделируемом объекте, что нужно учесть в различных моделях, получаемых при помощи этого метода описания. Это «самое важное» для какой-то модели и называют мета-модель, и выражена она главным образом в типах самых важных объектов, названия типов надо искать в разнообразных заголовках. **Если заголовков нет (например, лист электронной таблицы, где цифры есть, а заголовков для этих цифр нет), или в заголовках указаны типы мета-мета-модели (например, вместо «хлеборезка» вы видите «оборудование») — ждите ошибок!**

Для карты, являющейся моделью территории, мета-модель — это легенда карты, типы изображённых на карте объектов. Карта содержит ничтожную часть информации о территории, но это только важная информация. А легенда карты указывает на то, что на территориях важно, и поэтому нужно отображать в моделях. Дорогу на дорожной карте вы увидите, это важно, а вот цвет асфальта на этой дороге — нет, это не важно! А вот на карте животного мира вы и дороги не увидите, ибо не это важно. Тип важных объектов «дорога» для дорожной карты и «место обитания фауны» для карты животного мира задаётся мета-моделью, легендой карты. А сами объекты на дорожной карте — типа «дорога», а на карте животного мира типа «места обитания фауны». Тип один (задаётся легендой карты, методом описания, мета-моделью — это всё способ говорить об одном и том же: «что важное нас интересует»), а объектов этого типа в описании может быть множество, и этот тип может быть использован для построения множества описаний. Можно представить себе комбинированную карту, где изображены и дороги, и места обитания фауны. На таких картах удобно обсуждать, например, миграцию фауны и препятствия для этой миграции. Можно совмещать это на одной карте (говорят при этом о прожекторном описании, как у театрального прожектора внутри все цвета, так и тут в одном описании собрано несколько разных, и одно описание можно от других отделить фильтрацией), можно положить рядом две отдельных тематических карты и смотреть на них одновременно (это синтетическое описание: берём несколько разных описаний и синтезируем из них одно).

Ничего не меняется, если карту заменить диаграммой. Тип мета-модели — значок диаграммы, его форма. Ничего не меняется, если диаграмму заменить таблицей — тип мета-модели указывается обычно заголовками колонок таблицы (если транспонировать таблицу, заголовками строк — но это относительно редко). Ничего не меняется, если таблицу заменить текстом: тип мета-модели указывается прямо в тексте (не просто 8, а «мощность сепульки 8 Ватт» — тип «мощность в ваттах» — это тип мета-модели предметной области электротехники, а тамошние типы заимствованы из физики электричества).

Бывают и мета-мета-модели, ибо одни описания могут моделировать другие. Так, холодильник моделируется для инженера-ремонтника его принципиальной схемой, на которой обозначены функциональные части холодильника и какие-то соединения между ними (на принципиальных схемах обычно изображены какие-то знаки для функциональных частей, обрабатывающих потоки чего-то, и между этими знаками для функциональных частей изображают линии, указывающие на эти потоки. Электрические схемы показывают электрический ток, в принципиальной схеме холодильника мы увидим ток тепломассопереноса между функциональными частями холодильника). Принципиальная схема тут модель холодильника, а вот её обозначения (легенда) будут мета-моделью холодильника. Мета-модель холодильника — это те типы объектов, которые мы должны увидеть в холодильнике и отразить в его модели. Но когда мы говорим о том, как в компьютерном редакторе принципиальных схем моделируются обозначения для принципиальных схем холодильников, речь идёт уже о мета-мета-модели холодильника. Моделирование многоуровнево, и для компьютерных структурных[415] моделей обычно бывает три-четыре уровня мета-моделирования.

Множество связанных друг с другом моделей, сделанных с использованием разных методов описания (неважно, в рамках прожекторного или синтетического подхода к описанию системы) обычно называют **мульти-моделью**. Это всё равно как сборник множества карт для одной и той же территории: флоры, фауны, плотности населения, рельефа и т.д. Неважно даже, всё это отображено на одной карте-документе (прожекторная модель), или разных (то есть физически это сборник разных документов, синтетическая модель). Рассуждения абсолютно одинаковы для обоих этих случаев.

Обычно моделирование системы мультидисциплинарно. Каждая дисциплина/теория методов, которыми работает система и методов, которыми работают создатели над созданием и развитием системы задаёт свою онтику (набор типов мета-модели, иногда говорят framework), позволяющую удобно моделировать и затем обсуждать на этих моделях важные характеристики предметов какого-то метода работы. Электрики обсуждают в своих моделях чайника мощности, а промышленные дизайнеры — цвета в своих моделях (изображениях) чайника. Это прикладное моделирование монодисциплинарно, но системное моделирование — мультидисциплинарно, это мульти-моделирование, оно задействует множество монопредметных онтик (мета-моделей, если речь идёт о синтетическом подходе) или довольно обширную мультипредметную онтологию (мета-мета-модель и множество типизированных ей мета-моделей, если речь идёт о прожекторном подходе).

Мульти-модель. Мега-модель.

Само системное мышление с системным моделированием как обсуждением состава моделей в **системной мульти-модели** трансдисциплинарно/фундаментально, оно работает с монопредметными прикладными моделями, собирая эти модели вместе. Для этого задействуется интеллект-стек. Скажем, какая-нибудь электротехника::«метод инженерии систем, использующих электрические цепи» (электротехника ещё и задействует множество методов описания, заимствованных из физики) расскажет, как делать функциональные/принципиальные схемы

[415] Почему важно уточнять, что речь идёт именно о структурных моделях? Потому что сейчас есть и коннекционистские (нейронные) компьютерные модели, для которых эти рассуждения неприменимы.

электрических цепей в системах. Но системное мышление скажет, что функциональная модель в «системном описании»::мульти-модель обязана быть. Конечно, может существовать мульти-модель и без функциональной модели системы в её составе (в том числе без функциональной модели электрических цепей, если в системе они есть), но тогда это уже не будет системной моделью. И так со всеми основными видами описаний. В мульти-модели кроме основных может быть множество описаний, но в системной модели:

- Системные описания даются многомасштабные, то есть на каждом системном уровне (иерархия по отношению композиции), это принципиально. Описывается окружение, система, подсистемы (и это рассмотрение идёт на несколько уровней вверх и вниз от границы нашей/рассматриваемой системы, в том числе целевой системы)
- Обязательно включают описание функциональное, конструктивное, размещение, стоимостное (в том числе гибридные описания: концепция использования, концепция системы)
- Обязательно включают описание систем создателей в их графе (как минимум на одно отношение создания).
- Включают какое-то описание общего метода создания и развития (учитывают то, как происходит эволюционирование системы во множестве её «поколений», учёт «непрерывного всего»).

Не знакомые с системным подходом люди с трудом воспринимают идею множественности моделей, описывающих сложную систему. Обычно они требуют в мульти-модели указать «главную модель», которая является «правильной» по отношению к другим «вторичным» или «вспомогательным» моделям — но в системном мышлении нет «главной модели», для каждой проектной роли просто даётся её набор моделей для ответа на вопросы её ролевых предметов интереса. Нет «главной модели», нужны все модели, каждая в своей онтике/мета-модели/meta-model/framework. Хотя «главная модель» (определяющая набор всех этих моделей и то, что вообще должно моделироваться) таки есть, но это не модель и её мета-модель, но мета-мета-модель. При этом помним, что модели чаще всего задаются на формальном языке, мета-модель тоже обычно имеет формальное выражение (с определениями «как в математике»), а вот мета-мета-модель намеренно избегает определений, остаётся довольно общей (как говорит John Sowa, «намеренно недоспецифицированной»/underspecified, чтобы её типы классифицировали самые разные мета-модели). Понятия мета-мета-модели задаются их употреблением в самых разных контекстах, обычно в рамках изложения текстов на естественном языке — и ровно это и делаем мы в нашем руководстве по системному мышлению и других руководствах программы рабочего развития инженеров-менеджеров.

Если присмотреться, то и для формально определённых моделей в математике есть некоторое количество текста на естественном языке, задающего тип и примеры использования понятий мета-модели, и вот этот текст «пояснений и комментариев» обычно и задаёт фрагмент использованных для определения/definition мета-модели понятий мета-мета-модель/upper ontology (а в самых абстрактных случаях — мета-мета-мета-модель/foundational ontology).

Системное моделирование также должно быть «многомасштабным», обязательно выполнено на нескольких системных уровнях: модели нужны не только для целевой системы, но ещё и для надсистемы, и для подсистем. В системном мышлении

обсуждается появление новых свойств на каждом уровне. Проявление синергии — это резкое изменение старых свойств, у которых резко меняются их значения — скажем, «мощность 100 Вт» вдруг станет «мощностью 1500 Вт», но это всё та же мощность, старое свойство. А проявление эмерджентности — это новые свойства. Скажем, принципиальной схемы резистора, конденсатора, транзистора обычно не делают, там нечего моделировать (но могут посчитать сопротивление и ёмкость, исходя из их материалов и выбранной конструкции), но если собрать их некоторое количество, то получится усилитель — и у него новые свойства, уже усилителя, и для их описания нужны новые характеристики (например, коэффициент усиления, коэффициент нелинейных искажений, к подсистемам усилителя их применить нельзя). В силу эмерджентности нужны будут разные модели::описания на каждом системном уровне, для усилителя уже уместна принципиальная схема и какие-то расчёты по оптимизации выбора его комплектующих. Это нужно, чтобы согласовать интересы пользователей усилителя, разработчиков усилителя, закупщиков комплектующих и самых разных других проектных ролей. Модели для разных системных уровней обычно сильно различаются, даже если это «функциональные модели», или «конструктивные модели», или какие-то гибриды. Модели кирпича, модели стены и модели здания — это совсем разные модели.

Модели специфичны для проектных ролей каждого системного уровня, и если выбрать неверный системный уровень, то можно скатиться к редукционизму или холизму, а это ошибка. Редукционизм — это пробовать объяснить сложную систему взаимодействием не её подсистем на ближайшем системном уровне, а каких-то частей системы на существенно более низких уровнях, даже если в этих неадекватно выбранных уровнях выделить что-то на этих уровнях важное и опустить при описании всё на этих уровнях неважное. Да, человек состоит из атомов, это правда — но это неправильный системный уровень для объяснения того, чем человек отличается от роботов и кошек, неправильно моделировать человека как набор атомов, если хочется, например, его научить арифметике. Если захочется отремонтировать поломанный экскаватор, то моделирование экскаватора из атомов, или даже молекул для ведения этих ремонтных работ будет крайне неверным решением. То же самое относится к холизму: объяснению того, что в системах всё зависит исключительно от их надсистем. Если это принять к сведению, то поломка унитаза будет быстро приводиться к зависимости от макроэкономической ситуации в стране.

Нет, модели должны быть удобны для деятельности на каждом системном уровне как уровне масштаба систем, они должны позволять проводить объяснения (обсуждать причины и следствия происходящего в предметной области интереса), а не абстрактно «научно правильными».

Повторимся: **системное мышление — это не чистая математика, не чистая логика, его модели и рассуждения с их использованием определяются интересами в выполнении работ по каким-то методам, опора тут на методологию как дисциплину интеллект-стека. Если какие-то модели и рассуждения по ним формально правильны, но не помогают что-то сделать в физическом мире, то они бесполезны, их не нужно делать.**

Для создателей и их работ, которые они ведут по каким-то методам, тоже нужны модели. Множественность описаний означает множественность моделирования. Удерживать внимание проектных ролей на главном (определяемом как типы мета-модели, в свою очередь определяемые как типы мета-мета-модели), и не

привлекать внимание к неглавному (то есть не отображать в модели случайные объекты произвольно выбранных типов — скажем, цвет корпуса при прочностном его расчёте, причёску человека при проектировании метода его обучения какому-то мастерству) нужно сразу для многих методов работы в проекте. Для этого и служит культура моделирования (культура — метод, признаваемый многими): отражение главного, типы которого берутся из культуры (культура — все знания человечества). Культура, конечно, помогает, если кто-то с этим уже работал до вас. Но вы можете высказать догадки о мета-модели (или даже необходимой мета-мета-модели) в ходе исследований, если вы первый, кто разбирается в новой предметной области и пытается понять, на каких типах объектов в ней удаётся получить объяснения, как связаны в этой предметной области причины и следствия, какие методы работы могут быть в работе с этими предметами метода и требуемыми для этих предметов состояниями.

Как узнать, что должно в проекте быть важным? Мета-моделировать, двигаться от модели к мета-модели и затем мета-мета-модели (важное в построении мета-модели тоже не произвольно, а задаётся типами мета-мета-модели).

Один из критериев хорошей модели — это удобство обсуждения причинно-следственных связей в ситуации, а для этого удобство проведения контрфактических[416] рассуждений с объектами этой ситуации. Типы из мета-модели поэтому должны быть не любыми, а пригодными для таких рассуждений, этим вопросом пригодности занимается метод создания объяснений (входит в состав интеллект-стека). Нахождением таких типов занимается метод ведения исследований, это тоже один из методов мышления, входящих в интеллект-стек.

Получается, что **системное мышление, которое занимается главным образом составлением системных описаний (системным моделированием), задействует довольно много методов из состава интеллект-стека, и для хорошего владения системным мышлением нужно быть уже достаточно образованным человеком. Именно поэтому системное мышление у людей встречается не так часто: нужно учиться не только ему, но и всем методам мышления из состава интеллект-стека. Более того: системное мышление как карта местности, она не скажет «куда идти» — она даст описание местности, в том числе укажет, где горы, а где болота, где плодородные долины, где города. Чтобы куда-то пойти по этой карте, нужно будет задействовать рациональность: предложить разные методы поведения в текущей ситуации, прокритиковать их, выбрать какой-то лучший (часто говорят «наименее плохой», ибо «лучший метод» недостижим в реальности), запланировать работы по этому методу, а потом перейти к действиям: найти ресурсы и выполнить работы.** Подробней об этом в руководствах по интеллект стеку и инженерии личности, а наставления по моделированию даются в руководстве по рациональной работе. В нашем руководстве только рассказывается, как заниматься не любым моделированием, а моделированием «систем» как правильно выбранных важных объектов окружающего мира, то есть заниматься «**системным описанием/моделированием»/«описанием/моделированием систем»/«systems modeling»/«systemic modeling»**. Напомним, это такое мульти-моделирование, при котором:

[416] https://plato.stanford.edu/entries/counterfactuals/

- Отмоделировано несколько масштабов (системных уровней), на каждом уровне новые свойства (учтена эмерджентность).
- Отмоделированы обязательные аспекты: функциональный, конструктивный, размещения, стоимостный.
- Отмоделированы создатели.
- Отмоделирован процесс создания и развития целевой системы её создателями.

Системное моделирование нужно для коллективного мышления в проекте: все эти модели нужны для того, чтобы проектные роли договорились между собой по поводу целевой системы и всего того, что нужно для её создания и развития (включая, конечно, эксплуатацию всех версий системы).

Проектных ролей много, и что модель для одной роли — информационный шум и перегрузка «думалки» играющего эту роль агента (вычислителя с конечными ресурсами: будь то мозг, мозг с компьютером или даже просто компьютер) для другой роли, и наоборот. Возникает соблазн моделировать меньше, избавиться от лишней работы, но этого нельзя делать. Нужно моделировать (и документировать модели) для удовлетворения всех интересов всех причастных проектных ролей: ошибки в описании системы дорогостоящи, а эти **ошибки обычно выявляются как расхождения/коллизии/collision** между различными моделями системы. Вы узнаете о проблемах (проблема — когда никто не знает, что делать, то есть какой метод работы для каких объектов метода применить, чтобы перевести их в желаемые состояния), ибо проектные роли предъявят претензии к системе (в физическом мире), что наверняка отразится и как претензии к моделям, а иногда и претензии к мета-моделям, а изредка — и претензии к мета-мета-модели. Например, наставления нашего руководства вполне могут устареть, скажем, поменяться состав обязательных аспектов моделирования для системной модели.

Ошибки/коллизии в системной модели (несоответствия между моделями в мульти-модели, несоответствие типов объектов модели типам мета-модели, несоответствие типов мета-модели необходимым типам мета-мета-модели, несоответствие ожидаемой степени формальности модели и предъявленной в модели, и т.д.) желательно выявлять и исправлять при описании и документировании системы перед изготовлением, а не после изготовления, и уж тем более не при эксплуатации системы. Семь раз опиши и покритикуй (и не только найди коллизии, но и обнаружь другие ошибки — например, опечатки. Впрочем, они всё одно приведут к каким-то коллизиям), семь раз исправь после этого найденные ошибки в системной мульти-модели (в том числе коллизии между отдельными моделями в мульти-модели), затем один раз изготовь по системной мульти-модели как проекту/design и далее ещё и проверь, соответствует ли изготовленная система её системной мульти-модели! И продолжай это делать постоянно, со всеми версиями системы, в том числе не только во время создания, но и во время эксплуатации.

Захватывающая время эксплуатации мульти-модель обычно называется «цифровой двойник»/«digital twin». Цифровой двойник в том числе может включать и мульти-модель/design, используемую при проектировании и изготовлении версий системы, но главное — это мульти-модель времени эксплуатации, причём эта мульти-модель задействуется для автоматизированного принятия операторских решений по эксплуатации (настройки параметров работы, индикация желательности

превентивного ремонта ввиду износа и т.д.).

Итого: в системном описании моделей должно быть много (системная модель — это мульти-модель), ибо с системами связано множество разных методов работы, выполняемых самыми разными проектными ролями, реализующими самые разные намерения, проистекающие из их предпочтений в самых разных предметах интереса. А чтобы внимание к важным описаниям не уплывало в ходе согласования действий агентов, выполняющих разные проектные роли, мульти-модели нужно документировать в виде, позволяющем ими пользоваться всем заинтересованным ролям.

Это иногда называют **«управление информацией»**: все модели, которые кому-то нужны, должны быть доступны и изменения в них должны доводиться до всех «заинтересованных сторон»/ролей/stakeholders.

Кроме моделей в составе системной мульти-модели надо в явном виде иметь ещё и мета-модели этих моделей (метафора тут — для набора карт нам нужно ещё иметь и набор легенд для этих карт). Для разных наборов данных, чтобы их прочесть, нужно иметь их модели данных. Например, для базы данных сведений о концертах в Париже осенью 2029 года нам нужна ещё и модель данных для таких сведений, а не только сами сведения: надо знать, что там будет в сведениях (скажем, только название концерта и его время, или ещё и вместимость зала, цена билета, адрес зала, чат для справок). Модель данных — это мета-модель для сведений о концертах, которые — модель самих концертов. Будем называть мульти-модель в сумме с определяющими её мета-моделями — **мега-моделью**.

Если вы встретили лист электронной таблицы, на которой нет заголовка и приведены две небольших таблички, для которых тоже нет заголовков ни самих табличек, ни их колонок — вот это и есть мульти-модель, каждая табличка там — модель. Если заголовки есть — то это мега-модель. Обычно мета-мета-модель в мега-модель не попадает (помним, что мета-модель обычно хорошо формализуется, а вот мета-мета-модель обычно выражается текстом), но формальная (например, в виде базы данных со строгой типизацией её данных) мега-модель может быть дополнена каким-то описанием того, откуда и как брали в неё типы мета-модели (как создавалась модель данных, каких типов там данные). Это описание мета-мета-модели тоже можно считать частью мега-модели.

Если наш ролевой предмет интереса географический, то он оформляется географическими описаниями — картами. Для географического описания/моделирования холма в соседней деревне нам потребуются самые разные модели (карты), составляющие целый атлас (мульти-модель), но кроме этого нам потребуется знание, какие виды моделей должны быть в атласе (например, физическая карта, политическая/административная карта, карта полезных ископаемых, карта плотности населения, карта флоры, карта фауны, карта почв и т.д.), задаваемые мета-моделями (легендами соответствующих видов карт, которые разрабатываются исследователями-географами и обычно доступны из культуры, их можно «нагуглить»).

Если нам потребуется отмоделировать холм около другой деревни, то у нас будет другая мульти-модель при сохранении всех тех же мета-моделей. Рассматривать же и обсуждать в целом можно будет только мега-модель: без обсуждения видов карт, соглашения об условных обозначениях на этих картах и прочего, относящегося к

мета-моделированию, обсуждать карты нельзя.

Если вы не договорились, на каком языке пишете проектную документацию (китайском, суахили или румынском), то вам может понадобиться её переписывать — но перед перепиской всё равно придётся договориться о языке.

Если вы не договорились, в каком формате вы предоставляете отчёт о прибылях и убытках, то вам может понадобиться его переписывать, но перед перепиской всё равно нужно будет договориться о мета-модели из финансового метода описания — какой же формат отчёта о прибылях и убытках нужно предоставить. Нельзя моделировать без знания мета-модели и договорённости о том, что она сможет отразить предмет интереса какой-то проектной роли. Не знаете о существовании типов и присвоении типов (не имеете образования, например), не нашли нужный учебник и выдумали мета-модель «из головы», «на коленке», не документировали мета-модель — и дальше будут бесконечные провалы из-за невозможности объяснений по поводу значений важных характеристик, отражённых как объекты плохих или вообще неизвестных/непонятных типов. Повторим: «-- Петька, приборы! — Восемь! — А чего восемь?! — А чего приборы?!», — эта ситуация и есть работа с моделями без мета-моделирования, желательно многоуровневого. Вот так и выглядит мульти-модель (например, файл с данными, структура которого известна читающей программе, но в самом файле не описана — данные есть, а вот «модели данных»/мета-модели нет). Но даже если в этом файле будут прямо внутри приведены не только данные по описываемому объекту (модель, например, данные о сепульке, 8), но и данные про это описание (мета-модель, например, не просто 8, а что эта мощность в ваттах и дополнительно указано, что эта мощность относится к системе «сепулька»), то возникнет вопрос о том, откуда взялась сама эта модель данных — и надо будет вспомнить о мета-мета-модели, например, в варианте нашего руководства по системному мышлению. Мы уже неявно это сделали, когда кроме мощности в файле указали, что эта мощность относится к системе «сепулька», мощность тем самым не существует сама по себе, это характеристика системы, и хорошо бы указать, какой именно (про нюансы — вид/тип системы мы указываем, или экземпляр какого-то вида, мы тут не говорим, но и с этим придётся в моделировании разобраться).

При обучении моделированию учат не какой-то модели (модель будет каждый раз создаваться новая для каждой новой системы!), учат набору типов мета-модели, объекты для которых нужно будет уметь находить в предметной области какого-то интереса. Эта мета-модель (набор типов) будет одна и та же для самых разных моделей каких-то конкретных экземпляров объектов моделирования. Но для каждого экземпляра объекта моделирования надо будет выполнить моделирование (можно считать это вариантом «измерения» объекта). Не отмоделировали (не нашли мета-модели, например, про «сепульку» не узнали её мощность) — не договорились по поводу чьего-нибудь интереса, проморгали что-то важное в проекте.

Учат методам моделирования и мета-моделирования — они будут одни и те же для разных систем, они будут помогать учитывать интересы одних и тех же проектных ролей, хотя играть эти проектные роли будут разные агенты (люди и не только) в разных проектах — знание мета-моделей помогает переносить из проекта в проект опыт о том, что является в системе важным и что удовлетворяет интересам проектных ролей. Вы один раз учитесь 3D или даже 4D-моделированию инженерных систем, один раз учитесь моделированию мастерства, один раз учитесь

моделированию чего угодно, а затем используете это мастерство моделирования во множестве самых разных проектов для самых разных экземпляров систем из той же предметной области.

Если вы даёте кому-то описание системы (документированное! Устно — это не «давать», оно легко может быть не «взято»), то вы обязаны также сообщить прямо в документе, как вы получили это описание, указать использованный метод описания, в котором задать типы важных объектов, отражённых в модели (задать мета-модель, а то и дополнительно указать мета-мета-модель). Это можно делать в разных формах: предлагать модель данных, аннотировать типами (например, давая заголовки таблицам и колонкам таблиц или указывая типы мета-модели и даже мета-мета-модели прямо в тексте рядом с объектами модели). Вы должны понимать, что это передаваемое описание какого-то объекта — результат моделирования, то есть оно должно содержать только важные факты, при этом желательно — все важные факты для той проектной роли, которой вы даёте это описание, и не должно содержать ничего другого, что будет для этой проектной роли информационным шумом. Если вы не можете сказать, каким методом вы описали систему, если вы не знаете, отражает ли этот метод предмет интереса проектной роли, вы делаете ошибку, эта ошибка вам дорого обойдётся.

Вопросы. Множественность описаний в системном моделировании

Портной пришёл к королю договариваться о новом платье. Укажите из того, что у него было с собой — что составляет модель платья:
+++ карандашный эскиз платья ### Да, это модель платья, которая описывает внешний вид платья.

+++ картина маслом, изображающая короля в платье ### Да, это модель платья, которая описывает внешний вид платья.

--- бумага эскиза и холст картины ### Нет, это не модель, и не мета-модель — это инструменты для моделирования, части «моделера».

--- карандаши, краски, кисти ### Нет, это не модель, и не мета-модель — это инструменты для моделирования, части «моделера».

--- учебник рисования, том 1 «Проекция трёхмерных предметов на плоскость» ### Нет, это не модель — это мета-модель, описание способа сделать модель платья.

+++ записка с описанием основных деталей и потребного материала (кружева на воротник, золото для пуговиц и т.п.) ### Да, это модель платья, которая описывает важные для изготовления платья основные детали/конструктивы платья, а также потребный материал.

Дана карта полезных ископаемых. Что входит в мета-модель этой карты?
+++ легенда карты (условные обозначения месторождений) ### Да, это мета-модель — описание карты.

+++ масштаб карты ### Да, это мета-модель — описание карты.

+++ указание проекции (Меркатора, гномоническая, стереографическая, и т.п.) ### Да, это мета-модель — описание карты.

--- типографские данные (число экземпляров, типография, номер заказа) ### Нет, это не описание карты как модели местности. Это описание рабочего продукта (документации) как носителя информации, т.е. совокупности бумаги и краски, проходящих обработку в типографии по методу типографской печати. Это модель типографского тиража (число экземпляров указывает на то, что моделировался не экземпляр-документ карты, а весь тираж как система).

--- инвентарный номер, код тома проекта, к которому относится эта карта ### Нет, это не описание карты как модели местности. Это описание документа — носителя информации, т.е. совокупности бумаги и краски, как объекта корпоративного учёта.

--- наименование территории, для которой сделана эта карта ### Нет, это часть модели — часть самой карты.

Описание синтаксиса языка Паскаль обычно давалось в нотации Бэкуса-Наура. Описание этой нотации в учебнике программирования по отношению к программе контроллера робота, написанной на Паскале, это:
--- модель ### Нет, модель — это исходный текст программы на Паскале, который моделирует (описывает) программу контроллера робота.

--- мета-модель ### Нет, мета-модель — это описание синтаксиса языка Паскаль в нотации Бэкуса-Наура.

+++ мета-мета-модель ### Да, это описание нотации Бэкуса-Наура (мета-мета-модель), которая описывает синтаксис языка Паскаль (мета-модель), который описывает текст программы на Паскале (модель), которая описывает актуальную программу контроллера робота как объект физического мира (физический процесс выполнения программы в компьютере).

--- мета-мета-мета-модель ### Нет, это было бы описанием описания нотации Бэкуса-Наура (скажем, грамматика русского языка, на котором велось описание нотации Бэкуса-Наура)

--- мета-мета-мета-мета-модель ### Нет, это слишком большой уровень мета-описаний, трудно представить, что ему соответствует (на практике число уровней мета-моделирования редко превышает три или четыре).

Диаграмма Ганта в проекте сооружения жилого дома является моделью:
+++ системы создания жилого дома ### Да, эта модель описывает создателя дома, а именно — его работы. Создатель тут — строительная организация, которая создаёт/строит дом своими работами. Диаграмма Ганта описывает не дом, но создателя — в части работ этого создателя.

--- системы в окружении надсистемы жилого дома ### Нет, окружение рассматривается на стадии эксплуатации. В нём для жилого дома будут такие системы, как районная электрическая подстанция, система городских дорог, или эксплуатирующая организация. Но не строительная организация.

Николай прислал файл с моделью Варе. Варя не смогла с ним разобраться. Что ещё забыл прислать Николай Варе?
--- печеньки ### Нет, это не имеет отношения к делу.

+++ мета-модель ### Да, модель без мета-модели не может быть понята: если метод описания отсутствует, в самом описании невозможно разобраться. Если Варя не знакома с мета-моделью (не проходила модели такого типа в институте) — ей надо дать мета-модель (например, учебник).

--- документ — бумажное описание системы ### Нет, вид носителя информации обычно не важен, если Варя не знакома с мета-моделью — распечатка ей не поможет. Файл вполне документирует модель.

Инженеру поручили минимально описать холодопырку как систему. Он долго думал, потом принёс начальнику принципиальную схему холодопырки - и получил выговор. Что он, скорее всего, сделал не так?
--- принципиальная схема была сделана некачественно ### Нет, просто принципиальной схемы для минимального описания системы будет мало.

--- для принципиальной схемы был выбран неправильный метод описания ### Нет, просто принципиальной схемы (функционального описания) для минимального описания системы будет недостаточно, какой бы метод ни был выбран.

--- не был указан метод описания для принципиальной схемы ### Нет, просто принципиальной схемы (это функциональное описание) для описания системы будет мало, даже если указать метод её описания.

--- начальник хотел в качестве описания системы чего-то другого, инженер не угадал, чего именно ### Нет, просто принципиальной схемы как функционального описания для минимального описания системы будет мало, тут в угадайку не сыграешь. Надо добавить конструктивное описание, описание размещение, стоимостное описание.

--- перед описанием нужно было узнать предмет интереса начальника - узнать, какую роль сейчас исполняет актёр с полномочиями начальника ### Нет, его просили описать систему — то есть сразу было понятно, что одной принципиальной схемой не обойдёшься, в руководстве по системному мышлению говорится о том, какие есть обязательные виды описаний для системного описания.

+++ нужна была мульти-модель, одного вида описаний начальнику было мало ### Да, описание системы должно включать несколько видов описаний, обязательны четыре: функциональное описание (та же принципиальная схема подойдёт, но это только начало), конструктивное описание (например, состав модулей), физическое размещение/компоновка, и оценка совокупной стоимости владения. А если надо было принести несколько описаний — это уже мульти-модель. Ещё правильно включить кроме описаний «прозрачного ящика», то есть концепции системы, концепцию использования, то есть описания «чёрного ящика». И лучше бы ещё включить мета-модели всех описаний, то есть принести не просто мульти-модель, а мега-модель.

--- нужна была мега-модель принципиальной схемы ### Нет, мега-модель только принципиальной схемы (только функционального описания) не подойдёт. Мега-модель, в которую входят только принципиальная схема и мета-модель принципиальной схемы всё равно не спасли бы ситуацию. Кроме принципиальной схемы нужны другие описания системы (минимально — конструктивное, пространственное и стоимостное). Так что нужна мульти-модель (и даже не любая, а с минимальным набором аспектных описаний для системного подхода) в составе мега-модели.

Инженер-проектировщик заявил, что по окончании проектирования в описании системы у него будут только конструктивное описание и компоновка — этого вполне хватит, технологам завода больше и не надо, а он отдаёт результаты своей работы технологам. Прав ли инженер-проектировщик?
--- да, этого обычно хватает ### Нет, в описании системы должны быть и другие виды описаний — стоимостные/финансовые, разбиения работ и т.д. При этом надо не забыть ещё и концепцию использования! В проекте явно больше ролей, с которыми надо согласовать все описания системы, чем только технолог завода.

+++ нет, нужно больше описаний ### Да, в описании системы должны быть и другие описания — функциональные, стоимостные/финансовые, разбиения работ и т.д. Так что инженер не прав, этих описаний точно не хватит, и не забывайте ещё о концепции использования, без которой вообще непонятно, что описываем — какую систему!

Инженер включил насос в список комплектующих и присвоил уникальный тег (код функции комплектующего), закупщик назвал его в переписке предметом снабжения с номером по каталогу производителя, при установке монтажник внёс его в список установленного оборудования с инвентарным номером. Чей список главный, чей идентификатор нужно использовать всем остальным?
--- Инженера ### Нет, одного основного идентификатора в описаниях нет, все используются на равных — они удобны для разных проектных ролей, использующих их для разных деятельностей. Описание инженера

удобно для инженера, но нельзя заставлять пользоваться всех остальных только им, считая только его правильным. При этом помним, что функциональное описание (в задаче это тег) — «первое среди равных». Но среди равных.

--- Снабженца ### Нет, одного основного идентификатора в описаниях нет, все используются на равных — они удобны для разных проектных ролей, использующих их для разных деятельностей. Описание снабженца удобно для снабженца, но нельзя заставлять пользоваться всех остальных только им, считая только его правильным.

--- Монтажника ### Нет, одного основного идентификатора в описаниях нет, все используются на равных — они удобны для разных проектных ролей, использующих их для разных деятельностей. Описание монтажника удобно для монтажника, нельзя заставлять пользоваться всех остальных только им, считая только его правильным.

--- Какой-то ещё ### Нет, одного основного идентификатора в описаниях нет, все используются на равных — они удобны для разных проектных ролей, использующих их для разных деятельностей. Описание кого-то ещё будет удобно для этого «кого-то ещё», нельзя заставлять пользоваться всех остальных только им, считая только его правильным.

+++ Все идентификаторы используются на равных ### Да, одного основного идентификатора в описаниях нет, все используются на равных — они удобны для разных проектных ролей, использующих их для разных деятельностей.

Инженер готовится передать заказчику мега-модель тиокармы и для этого собирает нужные файлы. Что он должен не забыть включить в состав мега-модели:
+++ квантовомеханическую модель основной реакции ### Да, модели включаются в мега-модель.

+++ мета-модель использованного варианта квантовой механики ### Да, мета-модели включаются в мега-модель.

+++ 3D-модель тиокармы ### Да, модели включаются в мега-модель.

--- письмо автора 3D-модели, указывающее на использование в этом моделировании новейших компьютерных САПР, ускоривших разработку модели ### Нет, это письмо не входит ни в модель, ни в мета-модель тиокармы, это описание создателя тиокармы.

+++ указание на версию компьютерного формата представления 3D-модели ### Да, это указание на мета-модель для 3D-модели, она включается в мега-модель.

+++ текст стандарта, в котором подробно описывается формат представления 3D-модели ### Да, этот стандарт — мета-модель для 3D-

модели, он включается в мега-модель. Если текст стандарта легко найти, то можно просто дать на него ссылку.

--- тексты четырёх решений научно-технического совета о переносе даты рассмотрения мега-модели тиокармы ### Нет, эти решения не входят ни в модели, ни в мета-модели и, тем самым, не части мега-модели тиокармы. Они характеризуют систему создания тиокармы (даты работ по рассмотрению в этой оргсистеме мега-модели тиокармы).

+++ результаты измерений параметров реакции при трёх уровнях температур ### Да, результаты испытаний (они ведь модели) включаются в мега-модель.

Инженер готовится передать заказчику мега-модель тиокармы и для этого собирает нужные файлы. Что он должен не забыть включить в состав мега-модели:
+++ статьи, где описана параметризация реакции ### Да, мета-модели включаются в мега-модель, а параметризация реакции - это мета-модель для модели реакции в тиокарме. Может, статьи (если их легко найти) нужно включить не текстом, а просто дав на них ссылку.

+++ фотографию тиокармы в сборе и под нагрузкой ### Да, это модель тиокармы, она включается в мега-модель.

+++ все пять диаграмм и четыре таблицы концепции тиокармы ### Да, это просто девять моделей тиокармы, которые все включаются в мега-модель.

+++ разъяснения по используемым способам описания концепции тиокармы ### Да, это мета-модель для моделей описания тиокармы, использованных в концепции тиокармы, и они должны быть включены в мега-модель.

--- инструкция по установке моделера диаграмм для концепции тиокармы ### Нет, это не модель тиокармы и не мета-модель для этой модели. Это описание моделера, системы создания тиокармы. Конечно, можно объявить сам этот моделер мета-моделью для диаграмм, но обычно это не так: мета-модель диаграмм описывается в каком-нибудь стандарте, а потом можно выбрать любой моделер, поддерживающий этот стандарт. Передавать, следовательно, нужно описание диаграмм, а не моделер для этих диаграмм (хотя часто бывает, что описания диаграмм нет, а передавать надо моделер, а лицензия на него очень дорого стоит — стандартная «неожиданность» при передаче системных описаний, если они делаются на платных диаграммных моделерах).

+++ учебник с описанием основ тиокармостроения ### Да, с большой вероятностью этот учебник содержит мета-модели описаний тиокармы — поэтому его правильно включить в мега-модель (может, не целиком, а просто дав на него ссылку).

+++ требования к тиокарме в виде Excel-файла ### Да, это модель/описание тиокармы, и поэтому она должна быть включена в мега-модель. Но помним, что «требования» — это из старой инженерии, и от них отказались не зря. Попробуйте найти и включить также сценарии использования, на основе которых создавались требования, там может обнаружиться много любопытного, а сами требования сегодня — анахронизм.

+++ ссылка на использованный стандарт описания архитектурных решений ### Да, это ссылка на мета-модель для архитектуры, поэтому эту ссылку нужно включить в мега-модель.

Вы хотите подключить к проектированию системы дружественное конструкторское бюро. Для этого вы передаёте им:
--- мульти-модель системы ### Нет, этого будет мало, может не хватить информации о том, как сделаны эти модели, и их не смогут прочесть.

+++ мега-модель системы ### Да, к мульти-модели нужно добавить набор мета-моделей для этой мульти-модели (чаще всего это «настройки САПР и PLM» и их не так-то легко передать). Это и будет мега-модель.

Ваши конкуренты утащили мульти-модель вашей системы и надеются теперь узнать все ваши секреты. Ваши модели хитрые, до вас никто такого сорта моделей не делал. Смогут ли ваши конкуренты раскрыть секреты?
--- увы, да ### Нет, без мета-моделей они ничего не смогут понять в моделях, составляющих мульти-модель.

+++ к счастью, нет ### Да, у конкурентов будут проблемы, без мета-моделей они ничего не смогут понять в моделях, составляющих мульти-модель.

Ваши конкуренты утащили мега-модель вашей системы и надеются теперь узнать все ваши секреты. Ваши модели хитрые, до вас никто такого сорта моделей не делал. Смогут ли ваши конкуренты раскрыть секреты?
+++ увы, да ### Да, если известны мета-модели, то можно прочесть и модели.

--- к счастью, нет ### Нет, неверно: если речь идёт только о мульти-модели (о каких-то описаниях, но не о методах описания), то конкуренты бы не справились. Но тут утащили мега-модель, так что все модели конкуренты поймут.

Понятие конфигурации системы

Конфигурация/configuration системы — это текущее актуальное состояние системы (воплощения системы: всех частей на всех системных уровнях) и её описаний в их соответствии. Онтологически это сама система, но по факту — темпоральная часть, отвечающая какому-то состоянию её конструкции в плане наличия тех или иных версий конструктивов. Обычно в ходе разных проектов порождается (если «в металле», то даже изготавливается) множество самых разных

вариантов возможных частей воплощения системы разной степени неудачности, и только одна из них находит своё место в целевой системе в момент эксплуатации, остальные так и остаются вне системы — или запчастями, или просто выкидываются за моральным старением. Может существовать и много вполне удачных воплощений для подсистем, но в целевой системе использованы не все (например, насосов было изготовлено десяток, но в насосную станцию попал только один из них — остальные были изготовлены, но часть проданы, а часть лежит на складе как возможная замена насосу, который сейчас работает в составе станции, если тот насос поломается).

А ещё в ходе проектирования обычно разрабатывается множество вариантов самых разных видов описаний системы и её частей, относящихся к разным моментам времени, эти разработки делаются самыми разными агентами. В проекте нужно понимать: какие варианты изо всех изготовленных частей системы в конце концов вошли в текущее воплощение системы, а также какие варианты описания являются актуальными для текущей. Ведь отнюдь не все изготовленные части будущей системы идут в дело, некоторые остаются неиспользованными. Отнюдь не все варианты описаний системы идут в реализацию: некоторые отвергаются в пользу других, подходящих для успешности системы, дающих системы более функциональные, более надёжные, более дешёвые, более быстрые в изготовлении и т.д. Если вы прошивку телефона к прошлой версии примените к модели телефона следующей версии — не факт, что она будет работать. Такие **конфигурационные коллизии** не должны происходить, все версии как частей системы, так и описаний системы должны соответствовать друг другу. Поэтому понятие конфигурации как состояния версий самой системы распространяется и на описания — нам нужны только те описания, которые описывают текущую конфигурацию системы, возможно, только будущей системы (это иногда отдельно называют **конфигурацией описания системы**).

В современном производстве принята эволюционная идея, при которой о системе думают как о виде систем: примерно так же, как об эволюционирующем виде животных или растений.

Во-первых, вид систем или животных (это множество, тип) представлен экземплярами (уникальными физическими объектами), которые все генетически немного разные, но принадлежат одному виду. Скажем, система Boeing-737::система при этом мыслится как «вид» из множества экземпляров слегка разных уникальных экземпляров физических авиалайнеров. Как определить, системой называют «вид» или «конкретный самолёт с бортовым номером»? Из контекста, по-другому никак. Если вы говорите о компьютере Dell XPS 13, то вы говорите о нём как обо всех компьютерах серии Dell XPS 13 и их развитии (они будут довольно разные конструктивно, модели каждого года имеют весьма различающиеся конфигурации), или о текущем компьютере с серийным номером, но и тут могут различаться тип центрального процессора, объём памяти, разрешение экрана — это можно выбрать при покупке. И всё это будет Dell XPS 13, в чём-то эти компьютеры будут схожи. То же самое — операционная система Windows, софт в вашем компьютере. Это или все похожие друг на друга версии системы Windows в их отличии от, например, iOS, или же конкретная версия какого-то года, или даже конкретная уникальная версия в вашем компьютере. Как узнать, о чём идёт речь? Из контекста, а если непонятно — переспросить!

Во-вторых отдельно думают о мемоме (все описания системы, они живут и

развиваются в организациях-создателях, аналогично геному животных, который содержится в каждой клетке организма животного) и отдельно о феноме (совокупность свойств разработанных и воплощённых систем, у животных это совокупность свойств взрослого организма). В инженерии принят сегодня принцип «непрерывного всего»/ «continuous everything». Прежде всего это «непрерывная разработка»/«continuous delivery», которая не останавливается с выпуском воплощения первой версии системы в качестве MVP, minimal viable product, но главное — **«непрерывное введение в эксплуатацию»/«continuous delivery»**, не путайтесь тут с переводом, это не просто «поставка клиенту», это всё-таки ещё и прошедшее уже разворачивание у клиента, «непрерывное разворачивание»/«continuous deployment» и далее непрерывные испытания/тестирование»/«continuous testing». Непрерывное введение в эксплуатацию означает, что создаются последовательно разные версии системы с разными её конфигурациями (наборами физических материальных частей и соответствующих им описаний, проходящие какие-то альтернативные маршруты по инструментам создателей — какую-то версию на одном станке делали, какую-то на другом), а ещё эти версии имеют разные описания. Понятно, что свойства всех систем получаются при этом разные. И даже эти вроде как конечные описания ещё и изменяются каждое по нескольку раз после доработок, исправлений ошибок, добавления новых возможностей/features в описываемую систему.

Как определить, какие из версий этих описаний должны быть использованы изготовителями для воплощения системы? А если часть изготовителей изменили воплощение системы так, что она уже не соответствует этим описаниям, а часть изготовителей работает «как договаривались» — можно ли быть уверенным, что из изготовленных частей можно будет собрать работающую систему? И какой именно экземпляр работающей системы? Автомобили Tesla разрабатывают сейчас в рамках производственного процесса, где разработчики и завод работают на одной производственной площадке, изменения в конструкцию вносятся без перерывания выпуска, операция на заводе может проводиться в двух альтернативных вариантах по образцу A|B testing[417], только сравниваются не два варианта всего автомобиля, а множество раз по паре вариантов в разных местах конвейера. Каждый автомобиль Tesla тем самым имеет уникальную конфигурацию, он не похож на другие автомобили той же марки, различаясь во множестве мест. Впрочем, и авиалайнеры делаются так же — чуть разные модификации двигателей, чуть разные конфигурации кресел, чуть разные системы управления, но нет двух одинаковых Boeing-747 в небе. При этом учитываем, что конфигурация системы это одно из конкретных состояний в плане версий частей системы самой «системы как вида» — эта «конфигурация» бегает по дорогам в случае автомобилей Tesla и летает в воздухе в случаях Boeing, это не описание конфигурации, которое не бегает или летает, а документировано в базе данных. Ближайшее к ней приближение — это пока ещё не бегающие и не летающие автомобиль или самолёт, которые уже собраны и проверенные стоят на заводе, они «в металле», но ещё не в эксплуатации. Это только программисты путают конфигурацию своей «проектной документации» (их исходных кодов) с конфигурацией самой системы (их уже работающей в машинных кодах программы). Инженеры «железных» систем точно должны понимать, что конфигурация системы, а то, что там есть ещё и описания без конфигурационных коллизий — это очень хорошо, но недостаточно для

[417] https://en.wikipedia.org/wiki/A/B_testing

разговора о конфигурации, конфигурация описаний системы — это не конфигурация системы в момент её эксплуатации.

Ошибки, связанные с тем, что некоторые части системы и их описания не известны, или даже известны, но не соответствуют в своих текущих версиях друг другу, весьма распространены — эти ошибки и называют конфигурационными коллизиями (на инженерном сленге конфигурация с такими ошибками называется «разваленной конфигурацией», это очень неприятная ситуация, у «железных» инженеров обычно говорят «система не собирается», ибо все детали изготовлены — но собрать из них систему невозможно, «вот это круглое в одной части не лезет вот в это квадратное в другой части — непонятно, как такое получилось»).

Конфигурация системы — это сами части воплощённой системы и их описания, в современной инженерии конфигурации каждого экземпляра уникальны. Сама конфигурация может быть описана (defined) и, соответственно, документирована (described). **Документация конфигурации** (configuration description, иногда говорят и «конфигурационное описание») в речи часто сокращают до просто «конфигурация» — и разницу между конфигурацией (составом самой системы) и документацией конфигурации нужно определять из контекста так же, как разницу между архитектурой и архитектурной документацией, когда для обеих используют слово «архитектура».

Управление конфигурацией (configuration management) — отслеживание, что конфигурация воплощения системы и описания системы известны и соответствуют друг другу. Эта дисциплина лежит посредине между менеджерскими и прикладными инженерными дисциплинами. Конфигурация составляется из **конфигурационных единиц** (configuration item) — самых мелких частей, на которые делится система в части её логистики. Речь идёт о единицах передачи частей системы и описаний этих частей от одного исполнителя работ к другому, от одной обработки к другой.

Инженеры целевой системы часто имеют более детальные описания частей системы, чем это требуется для организации перемещения рабочих продуктов. Например, каждый из многих миллиардов транзисторов на чипе уникален, но конфигурационной единицей в цеху электроники служит сам чип, а не эти транзисторы, ибо отдельно транзисторы никому не передаются.

Чтобы не потерять эти конфигурационные единицы при учёте, иметь возможность на них сослаться в разговоре и различных описаниях системы, им даются индивидуальные обозначения (designations). Стандарт IEC 81346[418] представляет собой стандарт, в котором предлагается типовой способ такого описания, учитывающий системный подход — наличие нескольких альтернативных структур системы (функциональной, модульной/продуктовой, размещений, стоимостной и т.д.).

Версия (version) системы — это её конфигурация по состоянию на какой-то момент времени. Версии системы дают имя. Отслеживание смены версий в результате **изменений** (changes) нужно для того, чтобы точно указывать, какая из многочисленных возникающих и исчезающих по ходу проекта конфигураций системы имеется в виду. То, что не из любых версий можно собрать работоспособную систему, часто называют **конфигурационными**

[418] IEC 81346-1:2022, Industrial systems, installations and equipment and industrial products — Structuring principles and reference designations — Part 1: Basic rules, https://www.iso.org/standard/82229.html

зависимостями/dependances.

В программной инженерии тоже есть варианты стандартов описания конфигураций, но они обычно не учитывают системный подход и опираются главным образом на идею о версиях исходного кода. Популярны «системы именования» частей кода Semantic Versioning[419], где делается попытка менять имена версии на основании того, большие изменения вносятся в исходный код, или это маленькие изменения, но потихоньку выясняется, что и большие, и маленькие изменения могут поломать систему в случае коллизии одинаково, поэтому предложили вариант попроще и понадёжнее: Calendar Versioning[420], где просто отмечается дата последнего изменения версии — так что можно по имени версии сразу сориентироваться, давно ли что-то в ней менялось.

Документация системы (сегодня практически 100% это электронные форматы, часто даже не классические «документы», а базы данных) всегда является документацией каких-то конфигурационных единиц (читаем: описания — это всегда описания каких-то систем или описания содержания, а иногда и формы других документов, версии документации — это версии описаний каких-то версий систем, версий других документов), и её обозначения обычно создаются путём приписывания к названию конфигурационной единицы вида описания. Если для каких-то целей вдруг потребовалось описать три или четыре конфигурационных единицы, то скорее всего речь идёт об описании какой-то системы из этих единиц и нужно просто создать из них новую конфигурационную единицу, которой и будет соответствовать описание.

Базис (baseline, конфигурационный базис) — это проверенная на целостность и утверждённая административно версия, но всё чаще и чаще этот термин считают свидетельством анахронизма в разработке: проверки на целостность происходят сейчас буквально для каждой версии, а административного утверждения версии при выпуске нескольких версий продукта в день — нет.

Управление изменениями (change management) часто включают в состав управления конфигурацией. Так и пишут: «управление конфигурацией и изменениями», но обычно это отдельная дисциплина. Её не нужно путать с управлением организационными изменениями — как нужно менять организацию, чтобы это не вызывало сопротивления людей и вело к положительным результатам. Инженерное управление изменениями — это про то, как принимать решения по изменению конфигурации, чтобы минимизировать конфигурационные ошибки. Для этого в конфигурации раньше различали 1. обычные рабочие версии с более простыми внесениями изменений, и 2. базисы, внесение изменений в которые связано с дополнительными инженерными и административными проверками на целостность конфигурации. Сегодня это седая старина, используют другие подходы к управлению изменениями, например, trunk-based development[421] (trunk/ствол — только ствол дерева версий, без ветвлений). Появилось много интересных наблюдений, например, что качество систем, для изменений базисов которых собирался какой-то совет по внесению изменений по факту не отличалось от качества систем, для которых базис не проходил какой-то коллективной оценки: любое «замораживание конфигурации» замедляло разработку, но не приводило к

---

[419] https://semver.org/
[420] https://calver.org/
[421] https://trunkbaseddevelopment.com/

улучшению качества. Скажем, «заморозить внесение новых фич в конфигурацию, но не заморозить исправление в ней багов» повышало качество, но вот чтобы что-то там «замораживалось» в части описаний или самой конструкции — вот это оказалось вредным. Всё живое, всё меняется к (потенциально, не всегда получается) лучшему, а если ошиблись — это покажет интенсивное тестирование, которое проводится буквально для каждого экземпляра системы (один экземпляр — одна версия системы, как мы говорили), каждой версии каждого описания.

В любом случае, у вас должно быть налажено управление конфигурацией, вы в любой момент должны понимать, какие именно (из всех возможных вариантов) части системы стоят в изготовленной системе, по каким описаниям они производились — и тут важно, чтобы все они были без коллизий, в них не было ошибок и противоречий при объединении частей системы в систему и частных описаний в системное описание. Очень нехорошо получается, когда команда А разрабатывает свою часть системы с учётом результатов совещания, где размер части был принят 220 мм, а команда Б продолжает разрабатывать корпус, где для этой части предусмотрен старый размер 180 мм — ибо про совещание эта команда ничего не знала. Потом при сборке эта части встретятся: вот она, конфигурационная коллизия: ожидаемый корпусом размер части и размер изготовленной части не совпадают на 40 мм, система «не собирается». Лучше бы это найти до того момента, когда команда А уже изготовила эту часть системы, а команда Б — изготовила корпус. Вот этим и занимается управление конфигурацией и изменениями, чтобы такого в принципе не могло бы получиться.

Функциональные описания: принципиальные схемы и сценарии

У очень и очень многих проектных ролей область интересов включает важные характеристики системы времени работы/operations, когда система готова и эксплуатируется, в run-time/operations. Объяснить, как система работает (то есть описать причины и следствия во взаимодействиях подсистем), можно при помощи функциональных описаний. В них мы объясняем назначение (функцию/метод работы) каждой функциональной части системы и вклад этой части в достижение общего назначения системы, общего поведения системы в её окружении (run-time).

Принципиальные схемы — это диаграммы, показывающие соединения функциональных частей друг с другом и удобные для объяснения принципа функционирования системы (отвечающие на вопрос «как работает система» — как взаимодействуют между собой функциональные части, чтобы дать требуемую вовне функциональность системы в целом). Вот несколько типичных примеров принципиальных схем:

INPUT
6SN7
6SN7
2A3
250VDC
ACT
GND
EARTH
CHASSIS

3 Phase Inlet
PC
PT
PV
Vapor Outlet
Demister pad
LC
LT
Weir
Vortex
Breakers
LT
LC
LV
Oil Outlet
LV
Water Outlet

1
2
3
4
5
6
7
8
9
10

В ходе работы/функционирования функциональные части системы взаимодействуют друг с другом через **потоки**, проходящие через **порты** (ports) этих функциональных частей. В принципиальных схемах функциональные части обычно изображаются графическими элементами разной формы, а потоки — линиями между этими графическими элементами. Порт — это место присоединения соединительной линии к графическому элементу функциональной части.

Конечно, все эти красивые картинки в диаграммном моделере хранятся (а чаще всего и редактируются) в каком-то абсолютно другом формате, например, в формате табличного представления, и только на экран выводятся уже как диаграмма. Помним, что картинки в инженерии хороши только очень простые — и для показа уже готовой работы (это «архивные документы»), когда изменений можно уже не ожидать, ибо изменения в них вносятся трудно, а сами моделеры неожиданно дорогие во всех смыслах (в том числе дорогие в эксплуатации — на одно изменение обычно слишком много и долго надо кликать куда-то мышкой, в тексте или таблице сделать изменение во много раз быстрее). Вот в нашем руководстве картинки диаграмм есть, они очень простые, обычно это цитирование диаграмм других авторов, а когда речь идёт о наших собственных диаграммах, то это исключительно в иллюстративных целях и не предполагает изменения этих диаграмм в ходе какого-то проекта (помним про «непрерывное всё»).

В компьютере как источнике слайдов есть порт для присоединения слайд-проектора, и через этот порт в проектор текут данные слайда. Это мы дали функциональное описание, оно показывается на диаграмме. В физическом мире мы увидим конструктивную часть «ноутбук» и в нём конструктивную часть, реализующую давно устаревший интерфейс USB третьей версии (ещё не USB Type-C) или интерфейс HDMI, через который кабелем присоединяется проектор.

Конструктивный/материальный «ноутбук» реализует/воплощает функциональный «источник слайдов» (но конструктивно источником слайдов мог бы быть и смартфон), конструктивный проектор реализует функциональный слайд-проектор, кабель реализует соединение источника слайдов и проектора, интерфейс реализует межпортовое соединение. «Реализует» тут означает, что конструктивный (физический) объект выполняет роль функционального объекта. В функциональном описании обычно присутствуют термины предметной области времени эксплуатации («показ», «слайды»), а в конструктивном описании этих слов нет. Молоток и микроскоп не знают, что они будут забивать гвозди, в их описаниях

поэтому слов предметной области гвоздезабивания нет, равно как в предметной области компьютеров-ноутбуков слов про «слайды» нет, а вот слайд-проектор вполне может содержать такое слово, если он специализирован для показа слайдов, или не содержать этого слова, если он ещё и видеофильмы может показывать. И тут мы, конечно, описали маленький кусочек концепции системы — то, каким образом мы представляем себе разбиение системы на разные виды частей и как эти части связаны друг с другом, что там конструктивное что материальное воплощает. Дальше надо бы добавить пространственное описание и стоимостное описание, но нас пока будет волновать функциональное описание системы — но как всегда, мы ничего не можем сказать про какой-то объект (в данном случае функциональное описание), пока не расскажем про его связи с другими объектами (например, другими видами описаний), чтобы уточнить значение («значение — это употребление») и не выполним заземление/grounding (дадим примеры физических ситуаций, набор употреблений), чтобы не оказаться в наших объяснениях чересчур абстрактными. Так что само появление этого абзаца в руководстве — это просто норма работы по описанию (в данном случае описываем функциональное описание, часть мета-мета-модели, материал нашего руководства).

Функциональные части физичны (в момент, когда они воплощены конструктивными/материальными объектами), они выбираются для системы так, чтобы удобно было объяснять работу системы во время её эксплуатации/функционирования (run time, operations). Потоки — это чаще всего логические/ролевые/функциональные объекты, которые тоже физичны, выделяются из мира на основе постоянства пути, а не постоянства вещества (скажем, в реке путь воды одинаков, но сама вода в любом сечении — в каждый новый момент времени разная), и через которые функциональные части взаимно влияют друг на друга. Потоки могут быть самых разных объектов, которые проходят один и тот же путь: могут быть и потоки света, и электрический ток/поток/current, и потоки энергомассопереноса (в тепловых машинах), и даже потоки данных (в программных системах).

Потоки выделяются из физического мира на основе постоянства сохранения пути во времени, равно как функциональные объекты выделяются на основе постоянства сохранения функции во времени, а конструктивные — постоянства материала при изменениях, пространственные — постоянства места. Это идеи из онтологии/мета-мета-модели ISO 15926-2, мы уже обсуждали это в нашем руководстве более подробно в разделе про методы/функции/практики, в подразделе про ролевые объекты/роли и «игру по роли»/функцию/назначение в окружении.

В любом случае, мы приводим нашу мысль всегда в физический мир, но помним, что на принципиальных схемах изображается **поток**, а не конструктивный/материальный объект, играющий роль проводника этого потока: например, электрический ток может течь по проводнику, по корпусу, по болтовому соединению, но в принципиальной схеме это разнообразие сред не будет изображено. В электрической цепи в устройстве/изделии совершенно необязательно иметь ровно такое же количество проводов, которое изображено на принципиальной схеме этого устройства/изделия. Но между реализующими функциональные части продуктами/модулями ток всё-таки должен иметь возможность течь, чтобы система работала, «провод» — лишь один из вариантов реализации потока/соединения/связи/взаимодействия.

Точно так же «труба» на **гидравлической схеме** (принципиальная схема, где

потоки будут — потоки жидкости) будет только одним из способов реализации потока/взаимодействия — играющие роль функциональных частей системы конструктивы могут быть соединены друг с другом непосредственно, фланец во фланец, вообще без трубы, или вместо трубы жидкость может идти по какому-нибудь жёлобу, вариантов тут не счесть. Главное, что на диаграмме изображены соединения функциональных частей друг с другом через порты так, что можно отследить их взаимодействие через потоковую передачу какого-то вещества или энергии, или данных. При этом данные в конечном физическом итоге будут тоже переданы через передачу вещества, например, перевозку грузовика с магнитными дисками, или энергии, например, через оптоволоконную связь, или через электрический ток между частями компьютера — всё это сводится к потоку как объекту с постоянством/устойчивостью пути в ходе неминуемых изменений.

Помним, что функциональные описания могут показывать не только функциональные/ролевые части системы, но и непосредственно поведение/функции этих частей — и это поведение тоже будет осуществляться над потоками: предметами методов, проходящими по постоянному пути. В методах (впрочем, и функциях, это же синонимы — помним, что функции чаще употребляются для неодушевлённых простых агентов, а методы — для интеллектуальных агентов, которые ещё и планировать что-то могут) обычно говорят о входе (входной поток), обработке/функции/процессе (преобразование входного потока в выходной) и выходе (выходной поток). В этом случае говорят не о **принципиальных схемах**, как для технических устройств, а о «**функциональных диаграммах»/«процессных описаниях».** Конечно, диаграмма — это описание на языке типизированных объектов (тип объектов часто задаётся их формой — овал, кружок, квадратик или рядом рисуют картинку, в простейших таких диаграммах используют только один тип объектов) и отношений, чаще все указываемых стрелочками (они тоже могут быть с окончаниями и оперениями разных видов, в простейших вариантах — один вид стрелочек). «Процессные описания» могут быть не только диаграммны, но историческая традиция — ждать их всё-таки как диаграммы. Тут мы дадим диаграммы исключительно как иллюстрации, сегодня это не основное представление. Ожидайте в жизни увидеть не картинки, а тексты и таблицы.

Вот пример такой функциональной диаграммы (function model[422]) в давно морально устаревшей, но чрезвычайно распространённой и часто встречающейся нотации IDEF0. Слева от каждого элемента его входы, а справа — выходы, на стрелках написано, что именно передаётся от выхода предыдущей функциональной части на вход следующей. Какой тут самый распространённый тип ошибки при чтении таких диаграмм? Это функциональные части и «логическое время метода работы», но часто это читается как «физическое время протекания экземпляра работы», и вот тут будет проблема: можно попасться в западню ситуации «ножниц», когда функциональное и конструктивное разбиения будут различными и «логическое/функциональное время» не будет совпадать с «физическим/материальным временем». Для материального времени используются другие диаграммы, на них показывается часто синхронизация — это, например, тоже давно устаревшая нотация IDEF3 (и в этом и проблема, что вместо неё используют нотацию IDEF0, девять раз из десяти ситуаций когда методы/функции и работы относятся как 1:1, это ОК, но в десятой ситуации они расходятся как в примере с ножницами, где ножевой блок и рукоятка не совпали с изготавливаемыми в конструктиве двумя половинками ножниц). Так что вы в жизни обязательно увидите эти IDEF0 диаграммы, будьте аккуратны к тому, что на них пытались изобразить — это не всегда будут функциональные объекты и потоки.

Ещё один вид функциональных описаний — это **сценарии** (scenario), иногда говорят о **сценариях использования** (use cases, переводить нужно не как

[422] https://en.wikipedia.org/wiki/Function_model

«варианты использования», а именно «сценарии использования» — оригинальный термин придумал Ivar Jacobson по-шведски, и это был как раз «сценарий», case был признан им неудачным переводом на английский[423]).

Пример такой диаграммы[424] иллюстрирует действие активных частей системы, показываемых фигурками человечков. На картинке это проектные роли, но в жизни это вполне могут быть функциональные/ролевые части, которые играются не-людьми. В этих диаграммах используется «аристотелева физика» («палец давит на стол, но стол не давит на палец»), то есть показываются не взаимодействия, а именно действия: функциональные элементы обозначаются человечками-ролями/actors/деятелями, их действия (функции) моделируются отдельными кругами, а вся диаграмма отражает «логические» сценарии (не в физическом времени! В «причинно-следственной» последовательности объяснения происходящего) как методы/функции, которые выполняют функциональные части (роли, в терминах use case — это акторы/actors). Тут акторы могут быть и

[423] Подробнее в разделе History https://en.wikipedia.org/wiki/Use_case
[424] https://en.wikipedia.org/wiki/Use_case_diagram

неживыми, это необязательно «интеллектуальные агенты», принимающие решения. То, что в use case называют actor, не «агент-исполнитель роли», а собственно роль, причём ещё и безотносительно агентства (то есть без свободы воли, без возможности принимать решения, молоток в забивании тоже будет actor). Нужно всегда помнить, что слов в языке мало, и у каждого слова есть много разных значений, и термины «actor», «актор», «актёр» могут в разных дисциплинах разных методов работы обозначать совсем разные понятия, хотя иногда они и оказываются похожими до неразличимости. Так что не теряйте бдительности.

Режимы работы/функционирования какой-то системы обычно рассчитываются именно по функциональным описаниям, они ведь привязаны именно ко времени работы системы, а не ко времени её создания. Мультифизическое моделирование делается именно для функциональных описаний: ищутся оптимальные характеристики функциональных объектов для заданных режимов работы. Конечно, система моделируется в её окружении (это ведь run-time/operations-time, и работа системы существенно зависит от состояния её окружения). Будьте осторожными: мультифизическое моделирование иногда называют в инженерии «системным моделированием» и привязывают к специфическому классу софта, поддерживающему какой-нибудь физический моделер с описаниями на акаузальном языке моделирования физики, например, Modelica. Но нет, это не полноценное системное моделирование, это моделирование функционального описания (например, расчёт значений характеристик функциональных объектов по какой-то принципиальной схеме — часто это называют вообще 1D моделированием, ибо моделируем только одну какую-то характеристику при заданных остальных), а не всех описаний. Как это понять? Из контекста, как всегда: если слышите «системное моделирование» и уж тем более «системный моделер», просто из контекста поймите — вам рассказывают о системном моделировании в рамках системного подхода, или просто этим словом обозначают мультифизическое (механика, плюс электрика, плюс гидравлика и т.д.) моделирование какой-то системы.

Иногда функциональные диаграммы дополняют ещё и совсем уже процессными диаграммами, в которых объектами являются сами функции/поведение, называемые глаголами или отглагольными существительными — «переноска», «охлаждение», «освещение». Но процессные диаграммы встречаются реже «принципиальных схем» с функциональными объектами как элементами диаграммы, а не поведением функциональных элементов как на процессных диаграммах. Скажем, на процессной диаграмме может быть изображено поведение «повышение давления жидкости», а на принципиальной схеме будет «насос» («объект-повышатель давления», а не его действие «повышение давления»).

Конечно, нужно помнить, что большинство описаний в реальных проектах «гибридны»: люди (а сейчас и AI) и в диаграммах, и в таблицах, и в текстах вполне могут уточнить, что функция повышения давления жидкости реализуется модулем-насосом, а не перепадом высот и модулем-жёлобом, или даже явно указать марку и технические характеристики оборудования, которое нужно закупить!

Основная проблема функциональных описаний: их обычно понимают только узкие специалисты, функциональное/ролевое/«по назначению в надсистеме» разбиение системы на части-- контринтуитивно. Но **если вам задаётся вопрос о том, как именно работает система (время использования), то без какой-то хотя бы упрощённой принципиальной схемы или процессной диаграммы вам не**

обойтись. Обязательно готовьтесь к таким вопросам: в конце концов системы делаются для того, чтобы они работали/использовались, и большинству людей интересно, как они работают (что происходит во время использования, что система меняет в своём окружении, какие потоки чего проходят в этот момент между подсистемами системы) — и только после разъяснений на эту тему с ними можно будет обсудить, как и из чего система сделана (время разработки, конструктивы), как скомпонована, сколько стоит. Если непонятно, что система делает в своём окружении (какие объекты из каких состояний переводит в какие состояния в ходе работы) — непонятно, как оценивать её конструкцию, компоновку, стоимость. Хороши они или плохи — кого спрашивать, кто будет заинтересован в системе? Всё зависит от того, что система делает! Функциональное рассмотрение системы первично!

Системное мышление не учит тому, как читать и строить функциональные описания и документировать их. Системное мышление говорит, что функциональные описания должны быть сделаны (и документированы!). Это мета-мета-модель — даже не карта, на которой можно строить маршрут и дальше идти по территории, ориентируясь на этот «маршрут на карте», даже не легенда карты, а «легенда легенда карты». Но если вы хотите куда-то идти, то самая идея о том, что у вас будет легенда карты, потом надо сделать карту, потом проложить по ней маршрут — и пойти, она будет полезной. Лучше бы вы сделали именно так, сделали бы себе карту, нежели просто «взяли и куда-то пошли, по наитию» (и закончили тем, что оказались бы через пару шагов в болоте).

Как делать функциональные описания (с функциональной декомпозицией), как потом делать модульный синтез (принимать решения о том, какими модулями будут реализованы какие функциональные части) — это говорит системная инженерия. А если вы хотите стать профессионалом, то вам потребуется изучить ещё и прикладную дисциплину, занимающуюся конкретно какой-то инженерной подпрактикой (например, освоить пару-тройку учебников, где объясняются use cases и как их использовать в проектах для создания моделей использования в какой-нибудь «разработке приложений для смартфонов» или «разработке электромобилей»).

Конструктивные/продуктовые описания

Когда предмет интереса для какого-то метода работы относится не ко времени работы системы, а ко времени построения системы — что-то, что интересно для модульного синтеза, закупки конструктивов/продуктов/модулей/изделий/блоков, сборки системы как **конструкции** из уже закупленных конструктивных частей, отгрузки уже созданной системы/продукта/изделия, то необходимо использовать конструктивные/продуктовые описания. Эти описания отвечают на вопрос «как сделана система» (ср. с функциональным «как работает система») и они относятся ко времени создания (design-time, construction), а не использования (run-time, operations). Помним, что в design-time мы включаем не только проектирование, но и изготовление, и модернизацию, и уничтожение. Но не включаем время работы/operation/функционирования/эксплуатации/использования.

Как и в случае функциональных частей, есть разные графические (диаграммы), текстовые (на естественном языке, на полуформальном «псевдокоде», на формальных языках) и табличные форматы (например, в электронных таблицах

Excel, Coda.io, Notion.so, Airtable) представления описаний конструкций.

На продуктовых/конструктивных/модульных диаграммах показываются чаще **конструктивы/модули/блоки/изделия** (в случае блоков часто уточняют, что это конструктивные блоки, ибо бывают ещё и функциональные блоки как части функциональных разбиений) соединяемые через **интерфейсы** (interfaces, буквально — «междумордия», «то, что между»). Интерфейс обычно описывается каким-то стандартом, описывающим как свойства соединения, так и события, происходящие в ходе взаимодействия модулей через соединение. Принято говорить о сборке конструктивов для их взаимодействия через интерфейсы как об **интеграции**. Описывающие двустороннее взаимодействие стандарты называют **протоколами.**

Часто конструктивные части называют по-разному в зависимости от их уровня в конструктивном разбиении: детали (если не предполагается сборки-разборки, альтернативное название — элемент, «далее неразбираемое на части»), подсборки/subassembly, сборки/assembly, иногда даже подсистемы/subsystems (но тут легко перепутать с подсистемами в их функциональной ипостаси), и далее — система (тоже легко перепутать систему-конструктив с системой в её функциональной ипостаси, системой-ролью-в-окружении). Практика закрепления отдельных имён типов для уровней сборки — плохая, ибо в рамках эволюционирования системы вы заменили три детали и восемь винтов одной деталью сложной формы, изготовленной на 3D-принтере, а то и напечатали подсборку целиком — и всё, надо довольно много менять в плане именований в проекте. Помним: если у вас развесистая иерархия по какому-то одному типу отношения с одним типом объекта (одинаковые операции с этими объектами на каждом уровне, в том числе операции мышления об этих объектах), избегайте давать разные имена типов объектам разных уровней каких-то разбиений: число уровней наверняка поменяется в ходе создания и развития системы, вам придётся долго всё переименовывать.

Интерфейсы конструктивов похожи на порты ролей в том плане, что это именно места присоединения (при этом часто и на диаграммах это выглядит как «место в квадратике, куда присоединяется какая-то стрелка одним из концов»), то есть **интерфейсы сами — не конструктивы, они не занимают объёма в пространстве**, хотя у них и может быть форма «границы в пространстве». Вилка и штепсель, гнездо и штеккер, кабель USB и гнездо USB: для всех этих пар интерфейс — «то, что между», «междумордие/interface», а не сама «морда/face».

А как назвать сам физический конец электрического шнура, оформленный в соответствии со стандартом вилки, штеккер для какого-то гнезда, разъём в компьютере для USB кабеля? Это всё будут **интерфейсные модули**, главное назначение которых — реализовать какие-то интерфейсы в физическом мире, то есть быть парой к другому такому интерфейсному модулю. И у этого модуля не один интерфейс обычно, а два интерфейса: один целевой, а другой — присоединяющий его к какому-то модулю, для которого нужен целевой интерфейс. Конечно, «модуль» тут только один из возможных синонимов. «Изделие с интерфейсом», «интерфейсный конструктив»: любое слово, которое относится к конструктивам, времени сборки.

Это очень частая ошибка: путать интерфейс и интерфейсный модуль. Не делайте эту ошибку (в том числе студенты: не удивляйтесь отсылке на пересдачу,

если вы не дочитали до этого места в руководстве и называете интерфейсом какую-то часть конструкции, а инженеры-менеджеры — не удивляйтесь, если у вас в рабочем проекте будет возникать постоянное непонимание на ровном месте). Скажем, мембрана клетки — это интерфейс клетки с окружающим миром, или интерфейсный модуль? Поскольку мембрана состоит из молекул, занимающих пространство-время, то это интерфейсный модуль, конструктив. Мембраны, корпуса — это интерфейсные модули с окружением, а не интерфейсы! У них по два интерфейса: внутрь системы и вовне системы.

Вот пример такого старинного интерфейсного модуля (который в просторечии называют USB-интерфейс, что неверно — у него есть ещё и сигнальный интерфейс к плате, и отдельный интерфейс к питанию и даже интерфейс к человеку: светодиод, который мигает, когда идёт передача данных и кнопка включения, а ещё есть механический интерфейс крепления к корпусу или другой плате):

А как же соединения, которые нужны для работы — все эти трубы, кабели, волноводы? Это тоже конструктивы/модули/продукты/изделия/артефакты, и у них есть свои интерфейсы: они находятся между ними и теми модулями, которые они соединяют. Что проходит через эти интерфейсы, и как оно связано с работой всей системы?

Неизвестно, ибо речь идёт о конструктивных единицах в design-time: функции из run-time тут не определить, для этого нужно выходить за пределы модульного описания в область инженерных решений модульного синтеза по реализации функциональных частей системы конструктивными её частями (концепция системы как раз отсылает к такому соответствию в части внутреннего устройства системы, а концепция использования — про то, как там обстоит дело с системой в её окружении).

Решения модульного синтеза (изобретения в порядке создания концепции системы):

- определяют, роли каких именно функциональных частей будут играться теми или иными конструктивами,
- определяют, какие между этими конструктивами будут интерфейсы, и через эти интерфейсы какие будут проходить потоки между функциональными объектами (их мы обсуждали в предыдущем разделе про функциональные описания),
- учитывают при этом и все остальные аспекты системы (прежде всего аспекты размещений и стоимостные, но и многие другие, которые важны для учёта интересов многочисленных других ролей. Описаний в системных описаниях много, не только основные их виды).

Часто речь идёт о приёме-переработке-выдаче модулем какого-то проходящего через входные и выходные интерфейсы этого модуля физического потока. Этот физический поток будет реализовывать функциональный поток из функционального описания (например, принципиальной схемы), который идёт через функциональную/ролевую часть, на исполнение роли которой назначен модуль. Так, через USB-интерфейс может передаваться тот или иной поток данных (на принтер, на слайд-проектор, на внешний монитор), физически это будет соответствовать прохождению какого-то профиля электрического тока через контакты интерфейсных модулей — но на принципиальной схеме/функциональной диаграмме/dataflow diagram это будут потоки предметных из прикладной предметной области (domain) данных между функциональными частями системы, то есть потоки страниц для принтера, слайдов для проектора, экранов для внешнего монитора.

При интеграции (часто используют это слово) разных конструктивов в целую работающую систему важна не просто механическая сборка из готовых конструктивов, но ещё и согласование работы этих конструктивов после соединения их через их интерфейсы. Монтажник/сборщик должен убедиться, что соединение через интерфейс между парой интерфейсных модулей установлено, соответствует стандарту, описывающему интерфейс, и тем самым модуль сможет играть свою функциональную роль во время работы/функционирования/operations/run-time.

Для интерфейсов известны правила, по которым устанавливается соединение через интерфейс, но неизвестно, что именно и зачем передаётся через этот интерфейс — это будет известным только из функционального описания (например, принципиальной схемы). Метод/способ работы системы (функциональность) и его предметы появляются на принципиальной схеме, она появляется из разбирательства с функционированием, а не разбирательства с конструкцией и того, как именно собрана система. «Спальня» появляется из понимания того, что в спальне делают (методы работы), то есть спальня определяется не в конструктивном, а в функциональном мире. В конструктивном мире, мире конструктивов/продуктов/модулей это будет просто «комната». «Спальня» реализуется «комнатой» (а не, например, коридором или балконом), затем в ней нужна ещё как-то (кроватью, диваном, лавкой с мягким покрытием, спальным мешком) реализованная кровать (а в коридоре или балконе — не нужна, но если там оборудовать кровать, то можно назначить и конструктивы «коридор» и «балкон» исполнять роль «спальни»).

Например, принтер и компьютер связаны через USB интерфейс, но какая информация идёт принтеру, это интерфейсу неизвестно. Утюг к электросети подсоединён через интерфейс между евровилкой и евророзеткой, но этому интерфейсу неизвестно, какой через неё пойдёт ток и что этот ток будет делать в run-time. Но известны не зависящие от функциональности предельные значения тока, который может пройти через этот интерфейс, равно как предельное количество информации, которое может пройти через USB-интерфейс в единицу времени. **Задача модульного синтеза при проектировании системы — это выбрать такие интерфейсные модули, интерфейсы которых смогли бы выдержать потоки с их рабочими режимами, предусмотренные функциональным описанием системы.**

Вот ещё пример конструктивного/модульного/компонентного (в IEC 81346-1 предлагается компонентом называть заказной продукт, который приходит в проект

от других создателей) описания, в этом случае речь идёт просто о списке комплектующих (ещё одно название продукта/компонента), которые нужно купить для изготовления iPhone 2008 года, при этом приведены цены — и тут же это становится гибридным описанием, ибо включает и стоимостные оценки[425]:

3G iPhone Virtual Teardown Results—Major Cost Drivers (8Gbyte Version)*	
Pricing in U.S. Dollars	Part
$22.80	8Gbyte NAND Flash Memory
$20.00	Improved Touch screen
$20.00	Display
$13.50	Application Processor
$5.00	1Gbit SDRAM - Mobile DDR
$15.00	HSDPA Digital Baseband
$7.00	2 Megapixel Camera Module
$4.00	WLAN chipset
$4.00	Battery
$3.60	GPS
$2.00	Motion Sensor / Accelerometer
$4.25	RF Transceiver
$1.89	64Mbit NOR Flash Memory, 32Mbit PSRAM
$1.80	Charger (USB)
$2.00	Bluetooth 2.0 + Enhanced Data Rate (EDR)
$126.84	Total of Above Items
$37.16	Other Costs
$164.00	Total Bill of Materials (BOM) Costs (Direct Materials Only)
$173.00	Total BOM & Manufacturing Costs
$199	Initial retail price, 8Gbyte Version

** Doesn't include other costs, including software development, shipping and distribution, packaging, and miscellaneous accessories included with*

Обратите внимание, сколько тут упоминается разных стандартов: GPS, HSPDA, DDR, Bluetooth — перечисление интерфейсов, описываемых стандартами, обычно для конструктивных описаний. Ведь вся суть модулей — это замена реализации какой-то функциональной части системы путём простой замены конструктивной части: модуля на стандартном интерфейсе. Быстрый прогресс в инженерии смартфонов и ноутбуков определяется именно этим: тщательным описанием интерфейсов.

Вместо одного принтера через интерфейс USB к компьютеру можно подключить другой экземпляр принтера той же марки, или даже принтер другой марки, или не принтер, а какое-то другое устройство (скажем, сканер, или даже дополнительный дисплей) — без стандартизации интерфейса это было бы невозможно. Дальше будет работать конкуренция: цена и качество сервиса/функции/метода работы, получаемого через один и тот же интерфейс у конкурентов-поставщиков модулей будет разная, но смена модуля будет происходить легко для инженеров, им ничего не нужно будет менять в проекте, если они использовали **стандартный интерфейс**!

Стандартные детали (со стандартными интерфейсами) стоят дёшево, ибо их цену двигает вниз «конкуренция реализаций стандарта». Если вы хотите увеличить цену изделия, то уберите из этого изделия стандартные интерфейсы — и сразу получите внутри проекта огромную работу по согласованиям интересов самых разных проектных ролей, которую в случае использования стандартов провели разработчики стандартов. Этим часто пользуются в государственных проектах, чтобы обосновать их высокую стоимость, просто прописывают нестандартный интерфейс в контракте — и дело сделано! Если стандартная деталь стоит в любом магазине 3 рубля, то нестандартная деталь с той же функцией будет стоить огромные деньги: это всё равно как вместо одежды из магазина взять вдруг одежду

[425] https://www.zdnet.com/article/isuppli-iphone-3g-costs-173-to-make/

из дорогого ателье! Проблема в том, что замена дорогой одежды из ателье на покупную будет чаще всего незаметна, а качество дешёвой вещи может оказаться даже выше, ибо её изготавливают массово и проверяют лучше! В инженерии всё то же самое.

Метод борьбы с завышением стоимости заказной разработки — требовать, чтобы межмодульные интерфейсы были прописаны не в контракте, а в открытом стандарте, контракт же содержал только ссылку на открытый широкой публике стандарт. Тогда больше шанс, что возникнет конкуренция и цены на закупаемые модули станут ниже. Этот метод борьбы за низкие цены комплектующих называется **открытая архитектура**/open architecture[426]. Но регулярно архитектуру (решения по нарезке системы на какие-то модули с определением интерфейсов по связи этих модулей, то есть решения по особенностям модульной части модульного синтеза, в котором увязываются решения по функциональной декомпозиции и затем предлагается синтез целой системы из модулей) не открывают из соображений защиты от конкурентов, а также для того, чтобы проще было управлять конфигурацией. В закрытой архитектуре, чтобы поменять интерфейс взаимодействия модулей, нужно договориться меньшему числу людей. Но если интерфейс записан в открытом стандарте, то для изменения стандарта нужно много больше времени. Архитектура iPhone как раз относится к этому классу закрытых архитектур. И это не самое дешёвое устройство, хотя переговорная сила Apple достаточно велика, чтобы держать низкими цены на комплектующие: просто там большие объёмы выпуска каждого модуля, и эти большие объёмы — мощный фактор на переговорах.

Все эти рассуждения относятся и к предприятиям/организациям, но для них функциональные части — это оргроли, функции — методы/способы работы, конструктивные части — это оргзвенья, выполняющие работы по оговорённым в функциональном описании методам. Конечно, между оргзвеньями должны быть определены какие-то интерфейсы, и как они должны работать, определяется орг-архитектурой. Об этом будет рассказано подробней в руководствах по методологии и системному менеджменту, понятие архитектуры подробно разбирается в руководстве по системной инженерии.

Вопросы. Функциональные, конструктивные, пространственные, стоимостные части — 3

Чтобы вызвать Бэтмена, надо изобразить на облаках силуэт летучей мыши. Что реализует интерфейс для вызова Бэтмена?
--- Луч прожектора ### Нет, его недостаточно для появления силуэта летучей мыши на облаках.

--- Прожектор и луч прожектора ### Нет, силуэту будет не на чем проявляться, сигнал не будет передан.

--- Луч прожектора и облака ### Нет, лучу неоткуда взяться.

---

[426] https://en.wikipedia.org/wiki/Open_architecture

+++ Прожектор, облака и луч между ними ### Да, прожектор, облака и луч реализуют интерфейс для вызова Бэтмена.

--- Человек-пингвин ### Не имеет отношения к делу.

--- Что такое Бэтмен?! ### Не имеет отношения к делу.

Главное управление разведки РККА внедряет Всеволода Владимировича Владимирова в VI отдел РСХА, с позывным «Юстас». Сведения в ГУР РККА своему куратору «Алексу» он передаёт с помощью связной Кэт, у которой есть передатчик, шифр и протокол связи. Когда говорят про интерфейс между «Юстасом» и «Алексом», то имеют в виду:
--- Связная Кэт, с передатчиком, шифром и протоколом связи ### Нет, реализующие интерфейс конструктивы (связная Кэт, передатчик) — это не интерфейс.

--- Связная Кэт, с передатчиком, шифром и протоколом связи, а также автомобилем, в котором она выезжает к месту проведения сеансов связи ### Нет, реализующие интерфейс конструктивы (связная Кэт, передатчик, автомобиль) — это не интерфейс.

+++ Шифр и протокол связи ### Да, шифр и протокол связи описывают интерфейс — то, как взаимодействуют «Юстас» и «Алекс».

--- Передатчик, шифр и протокол связи ### Нет, реализующий интерфейс конструктив (передатчик) –не входит в интерфейс.

В цехе производства кроссовок на линии изготовления подошв поставили вторую машину для литья под давлением, такую же, как уже работающая, и отрегулированную точно так же. Цех также используется для обучения персонала технологии литья под давлением, новая машина не будет использоваться в процессе обучения. Машина 1 (M1) и Машина 2 (M2):
--- M1 и M2 являются конструктивными частями в системе производства и функциональными частями в системе обучения ### Нет, M2 не является функциональной частью в системе обучения.

+++ M1 и M2 обе являются конструктивом в системе производства, и ещё M1 является конструктивом в системе обучения ### Да, ибо M2 не является конструктивом в системе обучения.

--- M1 и M2 является единым конструктивом M1M2 в системе производства и таким же конструктивом в системе обучения ### Нет, M2 отличается — это не конструктив в системе обучения.

--- M1 и M2 являются функциональными частями как для системы производства, так и для системы обучения ### Нет, M2 не является функциональной частью в системе обучения.

Если рассматривать ассортимент магазина (совокупность товаров, находящихся на складе онлайн-магазина) как целевую систему, то интерфейсным модулем, реализующим интерфейс к ассортименту, будет:
--- Модуль управления корзиной покупателя на сайте магазина ### Нет, этот модуль взаимодействует с покупателями, но не взаимодействует с ассортиментом.

--- Модуль системы управления товарными остатками, генерирующий распоряжения об отгрузке заказов на основании данных, получаемых с сайта магазина. ### Нет, этот модуль входит в цепочку взаимодействия покупателя с ассортиментом, но непосредственно с ассортиментом не взаимодействует.

+++ Операторы склада, комплектующие и отгружающие заказ покупателю. ### Да, именно модуль «операторы склада» физически соприкасается с ассортиментом, он реализует интерфейс непосредственного взаимодействия с ассортиментом.

--- Протокол HTTPS, по которому покупатели соединяются с веб-сайтом магазина. ### Нет, протокол HTTPS — это вообще интерфейс, а не интерфейсный модуль.

--- Ничего из вышеперечисленного ### Нет, один из ответов вполне описывает интерфейсный модуль к ассортименту — с ассортиментом физически работают операторы склада.

Мальчик Петя приехал в гипермаркет, расположенный в торговом центре, и обнаружил, что тележки для покупок в торговом центре теперь прикованы друг к другу замками, и ему нужна монетка, чтобы открыть замок тележки и укатить её в гипермаркет. Пете нужен:
--- Конструктив в системе создания гипермаркета ### Нет, монетка ни в каком смысле не «делает» гипермаркет, она не часть создателя гипермаркета.

--- Функциональная часть в создателе гипермаркета ###Нет, монетка ни в каком смысле не «делает» гипермаркет.

--- Конструктив гипермаркета ### Нет, монетка не входит в состав гипермаркета как элемент конструкции.

--- Функциональная часть гипермаркета ### Нет, монетка не входит в состав гипермаркета как функциональная единица.

+++ Конструктив в окружении гипермаркета ### Да, монетка является конструктивным элементом системы выдачи тележек из операционного окружения гипермаркета.

--- Функциональная часть в окружении гипермаркета ### Нет, монетка — это конструктив, а функциональный объект называется как-то по-

другому, по функции — например «средство залога за тележку» или «жетон/token для выдачи тележки».

--- Интерфейс к создателю ### Нет, ни о каких интерфейсах к каким-то создающим гипермаркет системам речь не идёт. И монетка — это конструктив, а не интерфейс.

--- Интерфейсный модуль к системе создания гипермаркета ### Система создания — это которая в каком-то смысле создаёт (проектирует, строит, ремонтирует) гипермаркет. Ни о каких интерфейсах к такой системе речи не идёт.

Инженер «ГИПРОПуть» Петя разрабатывает проект укладки рельсового пути (3D-модель с допусками и посадками) и проводит расчёты его механической прочности. Инженер НИИ «Вагон» Маша разрабатывает обод колеса трамвая (3D-модель с допусками и посадками). В своей работе и Петя, и Маша, в том числе, руководствуются стандартами прокладки трамвайной колеи, стандартными длинами рельс и вагонов. Эти стандарты являются описаниями:
+++ Интерфейса между конструктивами трамвая и пути ### Да, именно так — стандарты описывают взаимодействие рельса как конструктива пути и колеса как конструктива трамвая.
+++ Интерфейса между системой трамвая и системой пути ### Да, стандарты описывают взаимодействие между трамваем и путём (системы — это не только функциональные части надсистемы, но и конструктивы тоже, просто это разные их ипостаси).
--- концепции использования трамвая ### Нет, эти стандарты не имеют отношения к концепции использования трамвая: в них не приводится описание поведения трамвая для соответствия своему назначению, не даются сценарии использования трамвая.
--- Ничего из вышеперечисленного ### Нет, стандарты описывают интерфейс между трамваем (колёсами) и рельсом — размеры, допустимые уровни давления и т.д.

Инженер Петя купил японский ноутбук Тошиба в производственном исполнении (работает от -50 до +85 градусов Цельсия, выдерживает падение с высоты человеческого роста, класс защиты от пыли и воды IP67), но обнаружил, что не может использовать его, т.к. блок питания ноутбука на 12 В рассчитан на входное напряжение 110 В, 50 Гц, штекер типа А. Для решения проблемы он покупает трансформатор, который может подключаться к бытовой электросети РФ 220 В и у которого есть питающая розетка 110 В. Петя подключает ноутбук к трансформатору и работает. Петя изменил:
--- Интерфейс питания ноутбука ### Нет, у самого ноутбука сохранился тот же интерфейс для его питания, 12 В.

+++ Воплощение системы питания ноутбука ### Да, воплощение системы питания было изменено: добавился трансформатор.

+++ Интерфейсный модуль питания ноутбука ### Да, интерфейсный модуль питания ноутбука был изменён: туда добавлен трансформатор.

«Интерфейс USB представляет собой маленькую печатную плату с закреплённым на ней USB-разъёмом». Как исправить эту фразу?
+++ Заменить жаргонизм «интерфейс USB» на «интерфейсный модуль USB» ### Да, интерфейс бывает «между модулями», а реализует его какой-то модуль — так что печатная плата — это не интерфейс, а интерфейсный модуль.

+++ Заменить жаргонизм «интерфейс USB» на «интерфейсный модуль USB» и добавить описание интерфейса подключения печатной платы к шине компьютера ### Да, интерфейс бывает «между модулями», а реализует его какой-то модуль — так что печатная плата — это не интерфейс, а интерфейсный модуль. Кроме того, этот модуль имеет какой-то интерфейс подключения печатной платы к шине компьютера, так что можно указать и его.

--- Убрать слова про USB-разъём, ибо и так уже ясно, что это интерфейс USB ### Нет, печатная плата - не интерфейс, интерфейс бывает между модулями, но реализующий интерфейс модуль должен называться «интерфейсный модуль», а не «интерфейс». Если убрать слова про USB-разъём - ошибка останется. Нужно писать «интерфейсный модуль» или «интерфейсный конструктив».

--- Указать тип микросхемы, преобразующей входной сигнал в сигналы стандарта USB ### Нет, печатная плата - не интерфейс, интерфейс бывает между модулями, но реализующий интерфейс модуль должен называться «интерфейсный модуль», а не «интерфейс». Если указать тип микросхемы, эта ошибка не будет исправлена.

«Вчера произошла блокировка автоматики наземной системы управления. Программа полёта была прервана на короткий срок. Проблема была в том, что не прошло подтверждение нормальной работы систем по одному из параметров. Мы в ручном режиме убедились, что системы работают нормально. По всей видимости, причина в дефекте одного из кабелей, кабели были заменены». О замене чего идёт речь?

--- О смене функциональной части ### Нет, ошибок в определении функциональности элементов не было, принципиальная схема не менялась. Проблема была в конструктивном дефекте, менялся экземпляр модуля.

+++ О смене конструктива ### Да, заменили конструктивы/модули (кабели, конструктивные единицы), их функции остались те же.

--- Об изменении топологии ### Нет, топология не имеет отношения к делу.

--- Об изменении размещения ### Нет, компоновка не поменялась.

--- О смене интерфейса ### Нет, способ соединения конструктивов не поменялся.

«В функциональной единице «дрон-носитель камеры», входящей в состав системы мониторинга стройки, заменили интерфейс подключения зарядки с проводного (через USB) на беспроводной.» Всё ли нормально с данным утверждением?
--- Да, утверждение сформулировано корректно ### Нет, у функциональных частей обычно говорят про соединения/связи, а интерфейсы определяют у конструктивов.

+++ Нет, в утверждении термины использованы сомнительно ### Да, сомнительно использование термина «интерфейс» в отношении функциональной части.

«В модуле компьютерного зрения дрона изменили интерфейс к системе распознавания изображений - теперь там используется API облачного сервиса вместо обращения к собственной библиотеке модуля.» Всё ли нормально с данным утверждением?
+++ Да, утверждение сформулировано корректно ### Да, говорится о связи модуля компьютерного зрения с разными модулями, реализующими функциональную часть «система распознавания изображений». Вместо модуля библиотеки через интерфейс используется облачный сервис, и соответственно был изменён интерфейс модуля компьютерного зрения.

--- Нет, в утверждении термины использованы сомнительно ### Нет, термины использованы корректно: речь идёт не о функциональных частях и их соединениях, а о конструктивных единицах (модулях) и их интерфейсах — говорится о связи модуля компьютерного зрения с разными модулями, реализующими функциональную часть «система распознавания изображений». Вместо модуля библиотеки используется через его интерфейс облачный сервис, и соответственно был изменён интерфейс модуля, у функциональных частей интерфейсов нет, есть порты и их соединения/связи/взаимодействия.

Платформы и платформенные/инструментальные стеки

Если рассмотреть конструктивное/продуктовое/модульное разбиение (PBS, product breakdown structure в отличие от функционального разбиения SBS, system breakdown structure), в нём можно увидеть на каждом уровне иерархии по отношению композиции какие-то наборы мелких конструктивных «кирпичиков», которые предоставляют через свои интерфейсы сервисы для сборки на их основе конструктивных «кирпичиков» более высокого уровня, которые могут выдать потенциально обширный набор функций во время их использования в каких-то ролях. Вот такой согласованный по предоставляемым ими совместно сервисам (**сервис** тут чуть-чуть уточняется как «функция, порт функционального объекта которой доступен в момент использования как интерфейс какого-то конструктива») набор модулей с их предопределёнными интерфейсами называют **платформой.**

Платформа — это всегда конструктивное/продуктовое/модульное рассмотрение системы с подчёркиванием стандартности получения ожидаемого в момент использования сервиса через опубликованный интерфейс.

То, что платформа — это конструктивное рассмотрение, а сервис — рассмотрение поведения роли (функционального объекта) как создателя, которую выполняет платформа в момент использования, не должно смущать, помним о 4D экстенсионализме: конструктив и роль — это один и тот же физический объект в момент использования. А «сервис» вместо функции/метода просто подчёркивает, что речь идёт об изменении какого-то внешнего предмета метода работы, выполняемого платформой — с новым уточнением: для платформы сервис выполняется по опубликованному интерфейсу, а из описания интерфейса понятно, где его искать, как обращаться за сервисом.

В инженерии

- за интерфейсы отвечает архитектор (именно архитектор говорит, конструктивы/компоненты/модули программы будут общаться путём вызова программ, или пересылкой сообщений, или будут использовать «шину», то есть все конструктивы в момент их работы будут получать все сообщения ото всех и отбирать себе нужные самостоятельно, если это компьютерная аппаратура, то именно архитекторы скажут, будет ли интерфейс USB Type-C или Bluetooth, или ещё какой. Архитекторы прописывают, как будет устроено взаимодействие на логистическом уровне, в отрыве от тех портов подсистем как функциональных объектов, которые будет реализовывать тот или иной конструктив этими интерфейсами. Их забота — правильная сборка системы из конструктивов, оптимизация (главным образом — минимизация) взаимодействий между ними.
- Разработчики берут за должное интерфейс и определяют, какие порты он будет реализовывать во время использования. Их задача — реализовать функциональность в рамках заданных архитектором ограничений (не всякий интерфейс сможет во время использования системы обеспечить работу порта какого-то функционального объекта).

Понятно, что роли архитекторов и разработчиков имеют абсолютно разные интересы (при общем интересе получить успешную целевую систему, но у них свои расходящиеся обычно представления, каким методом работы можно этого быстрее и дешевле добиться, разные критерии оптимизации). Подробней об этом можно узнать в руководстве по системной инженерии. Понятно, что орг-архитектор и орг-проектировщик тоже имеют такой же «рабочий конфликт», об этом подробней в руководстве по системному менеджменту.

Для программных конструктивов (компонент, модулей, приложений, библиотек — термин зависит от предпочтений школы программной инженерии) этот интерфейс обычно называется **API** (application program interface), программный интерфейс приложения/application program. Тем самым приложение, которое публикует описание своих интерфейсов, по которым можно получить его сервисы путём его вызова откуда-то извне, становится тоже платформой.

Если речь идёт не о программной системе, то можно говорить просто о прикладном интерфейсе (application interface, а не application program interface).

Платформа, если реализована библиотекой программ или «железными» модулями,

то можно выбирать в зависимости от проекта:

- входит как конструктив в состав приложения или продукта/изделия в ходе его работы, является его подсистемой, выполняющей какие-то роли,
- входит в состав окружения, предоставляя сервисы для каких-то частей системы (платформа предоставляется сервис-провайдером),
- входит в состав создателя (и тут тоже платформа предоставляется сервис-провайдером).

Слово «платформа» тем самым становится абсолютно универсальным, и им начинают называть что угодно, если понятно какой сервис на каком интерфейсе можно от этой платформы получить.

«Внутренняя платформа разработки»/«internal development platform» в platform engineering[427]. Сервис-провайдер для этой платформы создаётся инженером внутренней платформы разработки (это одна из ролей в инженерии), а сервисы этой платформы через опубликованный интерфейс получают разработчики — платформа изготавливает своими сервисами из программного кода работающую на компьютерном железе готовую программу. Термин «внутренняя платформа разработки» появился в программной инженерии. Но в «железной» инженерии это завод, который используется разработчиками. Подробней — в руководстве по системной инженерии. Аналогичной платформой будет администрация компании, которая предоставляет сервисы (финансовый, юридический и т.д.) для менеджеров-организаторов компании, она создаётся и поддерживается администрацией. Подробней — в руководстве по системному менеджменту. В учебных организациях поток студентов проходит обработку в деканате, который выполняет ту же роль администрации — это платформа, которая создаёт «учебную организацию» из студенческих групп, тот самый «завод», на котором делают из студентов мастеров. Подробней — в руководстве по инженерии личности.

Маркетплейс тоже называют платформой, ибо маркетплейс публикует интерфейсы сервисов для продавцов и покупателей, а иногда и платёжных систем. Иногда создание таких платформ называют уберизацией/uberisation, а для французской википедии приводят и синоним — платформизация/plateformisation[428] — в честь фирмы вызова такси Uber, которая создала платформу гарантированной «поставки против платежа» с двумя опубликованными интерфейсами: сервиса для водителей такси и сервиса для пассажиров такси. В принципе, сервис поиска каких угодно клиентов для каких угодно сервисов или товаров — это поисковая система, прежде всего Гугл. Но при уберизации платформой выполняется главный сервис: гарантирование поставки против платежа, платформа-маркетплейс выполняет роль страховщика и биллинг-агента как для клиента, так и для провайдера сервиса (а проекты, где этой страховки не происходит, обычно более чем скромны по масштабу — за просто сводничество много денег не дают, без страховки всегда можно найти поставщика сервиса через поисковик, а вот дальше надо будет разбираться со страховкой и биллингом).

В уже упоминавшихся социальных танцах (танго, кизомба, форро, сальса и т.д.) тоже есть интерфейсы, они обычно стандартизуются на уровне «базы»: есть условные сигналы, которые подаются leader/ведущим follower/следующему (это безгендерная терминология для «партнёра» и «партнёрши», на которую сейчас

---

[427] https://platformengineering.org/blog/what-is-platform-engineering
[428] https://fr.wikipedia.org/wiki/Ub%C3%A9risation

перешёл весь мир). Скажем, рука follower находится у вас (leader) на плече — и вы не можете крутить follower, эта рука будет мешать. Вы как leader даёте небольшой толчок своим плечом руке (это и есть «сигнал ведения» из «протокола взаимодействия») — и рука follower взлетает вертикально вверх. Теперь возможны повороты, эта рука не будет мешать, не будет за вас цепляться. Вот это и есть один из интерфейсов в социальных танцах (есть и другие): между плечом leader и рукой follower. В данном абзаце приведён кусочек стандарта с протоколом взаимодействия между конструктивами leader и follower в паре танцоров социальных танцев.

Протокол — это описание какие в интерфейсе возможны взаимодействия, какие, когда и в каком направлении передаются потоки (сигналов, энергии, вещества) и что вообще в каком порядке происходит. Протоколы могут быть

- внутренними для проекта (опубликованы в рамках коллектива разработчиков),
- открытыми (опубликованными от какого-то официального источника в **публичном документе**, не подразумевающем проверку выполнения или **стандарте**, подразумевающем проверку выполнения какой-то реализацией),
- передаваемыми в устной традиции (как древние былины и сказания), как протоколы ведения-следования в социальных танцах (они тоже могут быть опубликованы в каком-то обычно неполном виде, но публикации разнятся и нет официального источника) или протоколы этикета при еде (при этом в разных странах они разные, тоже опубликованы частично или даже более-менее полно во многих местах, но эти публикации разнятся и по объёму, и по содержанию, а ещё не имеют обычно даты с версией — нет официального источника).

В мышлении это мало отличается от обращения к программному сервису (помним, что «сервисом» часто называют и поведение провайдера, и самого провайдера сервиса, «службу», «сервер»), или обращением за сервисом к какому-то оборудованию. В этом плане и leader и follower, следуя стандарту «базы» танца с её сигналами, представляют друг для друга платформу этой «базы», они предоставляют друг другу сервисы (leader — сервис выдачи интересных сигналов ведения, follower — расшифровки и выполнения этих сигналов, одна из интерпретаций социальных танцев — это совместная «игра в загадки»[429]). И важная особенность социальных танцев: стандартизация интерфейса, «платформенность» приводит к тому, что leader и follower, приехав на какую-то фестивальную вечеринку из разных стран и впервые танцуя друг с другом, смогут абсолютно предсказуемо и надёжно взаимодействовать друг с другом, а также «обслужить» множество партнёров. Это взаимодействие в социальных танцах в системном мышлении описывается примерно в том же наборе понятий, что взаимодействие смартфона и его сотовой станции, трамвая с его путями, веб-браузера с веб-сервером — для всех этих случаев мышление устроено одинаково, используются одинаковые типы мета-мета-модели (но, конечно, абсолютно по-разному это всё называется в каждой предметной области. Так, стандарт ведения-следования в социальных танцах никогда не назовут интерфейсом, там таких «умных слов» не знают. Проблема в том, что его вообще никак не назовут, разве что «негласное соглашение о ведении-следовании»).

Если думаешь о сборке двух конструктивов/продуктов/модулей, думай об

[429] https://vk.com/wall-179019873_1550

интерфейсе, если через этот интерфейс в момент использования конструктивов в каких-то ролях происходит общение с чем-то меняющимся внешним (а не входящим в постоянный состав), «обслуживание» отдельных экземпляров предметов метода по общему методу, то этот метод — сервис, если интерфейс опубликован и сервисов несколько — речь идёт о платформе.

Основной вопрос при обсуждении платформ — это так называемая **видимость интерфейсов**. Интерфейсы какого-то низкого системного уровня не должны быть видны снаружи конструктива, то есть **нельзя соединять конструктивы иначе, чем через предусмотренные в них интерфейсы.** Грубо говоря, если у вас коробочка с каким-то разъёмом, то нельзя воткнуть внешнее устройство не в этот разъём, а куда-нибудь внутрь коробочки, мимо этого разъёма, даже если очень хочется. В системах всё со всем связано, и по неотслеженному соединению (это ведь будет тоже интерфейс, только «неучтённый») могут пройти паразитные связи, возникнут непредвиденные ошибки. Помним: архитектор борется с этими неучтёнными паразитными связями между частями системы, а разработчик наоборот — пытается каким-то образом обойти эти архитектурные ограничения, ибо ему кажется, что он так быстрее получит результат. Если мы в качестве разработчика берём хакера, то он первым делом пытается обойти опубликованный интерфейс и сразу получить доступ к функциональности (сервисам платформы) через неучтённые связи. **С этим борются «безопасники». По сути, «безопасники» и архитекторы — разные изводы одной и той же инженерной роли.**

Например, если в ваше подразделение, где вы тщательно планируете работу сотрудников, получая заказы на эти работы через какой-нибудь трекер, могут прийти какие-то люди из других подразделений и попросить сделать работы для них «вне очереди» — и им не откажут, то последствия для организации будут фатальны: какие-то неожиданные работы будут выполнены, какие-то вроде как свободные сотрудники окажутся заняты, какие-то важные (но важность определяется надсистемой! Внутри модуля-подразделения как заказчика, так и исполнителя важность может быть неизвестной) работы не будут выполнены из-за задержек, связанных с этой внезапно появившейся «из ниоткуда» (мимо запланированного интерфейса, поддерживаемого трекером) работой. Весь этот трекер тут — только интерфейсный конструктив/модуль, предоставляющий интерфейс для сервиса заказа работ, это неважно, что это довольно большая программная система, на рассматриваемом системном уровне это просто интерфейсный модуль для заказа работ, средство коммуникации подразделения с внешним миром.

В электротехнике есть три типа неисправностей: есть контакт там, где он не должен быть, нет контакта там, где он должен быть, и плохой контакт. Это шутка как раз про интерфейсы — которые должны быть именно там, где они предусмотрены конструкцией!

Для обсуждения видимости интерфейсов используется диаграмма конструктивных/модульных/продуктовых/платформенных уровней. Каждый платформенный уровень отделён от другого уровня каким-то стандартизированным (то есть описанным и документированным!) интерфейсом. Реализации/воплощения нижестоящих уровней тем самым могут быть сменены так, что надсистемы в момент использования этих конструктивов не заметят подмены, если эти конструктивы будут соблюдать опубликованные (в случае платформ) или даже внутренние (если

речь идёт не о сервисе, а просто о какой-то функциональности) интерфейсные соглашения.

Итоговое многоуровневое продуктовое/платформенное разбиение называют **платформенным стеком** или **инструментальным стеком** (stack, стопка — диаграммы похожи на стопку подносов в столовой или стопку листов бумаги в пачке). Вот пример различных инструментальных стеков для межкомпьютерной коммуникации в интернете вещей (IoT, internet of things)[430]:

На диаграмме видно, что в разных сравниваемых стандартах связи описываются/определяются/defined пять платформенных уровней (конструктивов по отношению часть-целое), которые можно разделить на реализующие в момент использования роли для функций передачи физического сигнала (physical), передачи данных (Data Link) и т.д. При этом конструктивы передачи данных при работе обращаются к конструктивам передачи физического сигнала, используя указанные в диаграмме протоколы интерфейсов[431]. И так по всем уровням выше, вплоть до уровня целого конструктива как приложения/application: каждый уровень платформенного/инструментального стека (уровень реализующих этот стек конструктивов) даёт интерфейс для оказания сервиса наверх и использует сам при этом сервисы нижележащих модулей. Где тут пролегает граница между системами и окружением? Всё зависит от проекта: часть этих уровней может быть реализована какой-то системой, а часть — её окружением.

Неважно, что делают/как работают эти уровни платформенного стека. Для «как работают» нужно обсуждать функционирование в run-time, какие там данные с какой целью будут передаваться — это тоже про run-time и обсуждать надо с разработчиками, но обсуждение платформ — это design-time, «из чего собрано и как соединено», время создания системы, обсуждать с архитекторами.

Главное в показе этого инструментального стека то, что

- никакой конструктив вышестоящего уровня «не видит» конструктивы более низкого уровня (не имеет к ним доступа, не может туда «воткнуться», ибо нет интерфейса), чем конструктивы уровня, находящегося непосредственно под ним,
- интерфейсы реализующих сервисы этих платформенных уровней стандартизованы — как стандартизован и сам набор этих уровней.

Эти инструментальные/платформенные стеки вездесущи. В танцах вы передаёте

---

[430] http://www.slideshare.net/Techtsunami/cn-prt-iot-v1

[431] Соглашение о взаимодействии, подразумевающее какую-то последовательность/порядок действий двух систем, называют протоколом. Стандарты, описывающие обмен данными, часто тоже называют протоколами: они ведь описывают последовательность обменов данными.

сигнал от плеча leader к руке follower (то есть от части тела к части тела в целом), а не от мышц к мышцам, или даже от мышц плеча через кожу, затем кожу и дальше через рецепторы и нервы дальше к мышцам руки follower. Мышцы — это конструктивные объекты более низкого уровня, и если у «плеча» и «руки» ничего не поломалось, чтобы чинить это на более низком уровне, то лезть на этот уровень не нужно, нужно пользоваться предписанными интерфейсами. Универсальность при этом растёт с каждым уровнем вниз. Если в социальных танцах можно передавать сигнал ведения-следования от плеча к руке, то на уровне работы мышц уже можно обсуждать не только передачу сигналов, но и самые разные другие сервисы. Если вы обсуждаете самый нижний уровень связных протоколов, то поверх них можно строить самые разные варианты реализации других протоколов. Например, в связном стеке для более-менее универсального протокола IP можно предложить более высокий уровень протокола TCP, но можно использовать и другие протоколы, которые будут реализованы другими конструктивами — UDP и ICMP. Это всё изображено на диаграмме платформенного стека IoT для TCP/IP Protocol Stack. Видно, что стек назван по паре протоколов где-то из середины стека, даже не верхний уровень стека и не все протоколы из того же уровня, что TCP, на иллюстрации приведены всего три протокола уровня TCP, но их там 145 разных[432].

Есть два способа изображения таких диаграмм платформенных/инструментальных/модульных/конструктивных стеков:

- на некоторых диаграммах плашка-«платформа» именуется названием описывающего её стандарта прикладного интерфейса. Прикладного — это наверх по системным уровням, прикладность идёт в стеке сверху. Реализация этого стандарта интерфейсным конструктивом находится как бы «между уровнями» (ровно этот способ показан на диаграмме инструментальных стеков IoT, он самый распространённый)
- показ собственно конструктивов (именование платформ) будет на плашках, тогда интерфейс этих конструктивов либо считается проименованным самим конструктивом (скажем, пакет/библиотека глубокого обучения TensorFlow имеет API TensorFlow, и поэтому его не нужно отдельно прописывать — названия типа конструктива «пакет» или «библиотека» и выдача имени TensorFlow для него достаточно), либо его указывают отдельно.

Но если платформенный уровень реализует стандарт с другим именем, то он прописывается где-то в границах реализующего его модуля отдельной строчкой поближе к границе над-модуля, или даже выносится и явно прописывается «между платформами», как это сделано в картинке стека безопасности приложений[433] — стандарты указаны сбоку картинки платформенного стека, а каждая плашка обозначает слой/уровень конструктивов:

[432] https://en.wikipedia.org/wiki/List_of_IP_protocol_numbers
[433] https://f5.com/about-us/blog/articles/why-cves-should-be-given-priority-one-for-resolution-27910

Верхняя скобка для стандарта интерфейса в надсистему от custom code ведёт как бы в никуда при таком способе показа, поэтому чаще используется способ предыдущей картинки, где платформа обозначается не по назначениям/именам её модулей/слоёв, а назначение указывается сбоку.

Часто оба этих способа перемешивают, получая гибридную конструктивную диаграмму конструктивных/инструментальных уровней для видимости интерфейсов, она разрабатывается архитекторами. Вот, например, диаграмма уровней компилятора искусственных нейронных сетей, где верхние строки — наименования программных пакетов/библиотек для сервисов над нейронными сетями (MXNet, PyTorch, …), а нижние — интерфейсных стандартов для какой-то аппаратуры (OpenCL, CUDA, …)[434]:

Это неважно, что вы можете ничего не понимать в предметной области, описываемой этими архитектурами (наборами решений по разбиению системы на модули и организации связей между модулями). Важно, чтобы вы могли оценить

[434] https://tvm.ai/2017/10/06/nnvm-compiler-announcement

похожесть друг на друга этих диаграмм (хотя бы визуальную похожесть), понять стоящий за ними мыслительный принцип — и лучше потом разбираться с тем, какого сорта диаграммы вам предлагает посмотреть та или иная проектная роль, или какие диаграммы вы рисуете сами. Конечно, вы можете платформенные стеки рисовать не как диаграммы, а представлять табличками, мы помним, что диаграммы менять неудобно, а таблички меняются почти так же легко, как текст.

Платформенность даёт возможность эффективного разделения труда: реализацией каждого уровня платформенного стека может заниматься отдельная команда, хорошо разбирающаяся в работе на их системном уровне (системный подход!), а поскольку речь идёт о конструктивах, воплощающих роли (функциональные объекты), то такие команды могут соревноваться за эффективность реализации требуемых функций сервисами своих модулей — на каждом системном уровне своих.

Инженеры, использующие какие-то платформы для создания своих целевых систем, могут не задумываться, как именно и из каких систем были сделаны сами эти платформы. Если вы пользуетесь каким-то сотовым телефоном, то вам всё равно, из чего и как сделан этот телефон, если он умеет связаться с базовыми станциями вашего провайдера сотовой связи (а он умеет: стандарты/протоколы взаимодействия с этим провайдером опубликованы, и изготовители базовых станций и телефонов просто их соблюдают). И вам всё равно, в какой школе танцев училась ваша партнёрша или партнёр, если соблюдается стандарт «базы» этого танца: вы сможете танцевать вместе (но тут ситуация сложнее: стандарт базы не публикуется, а передаётся в устной традиции преподавателями танцев — примерно как былины передавались сказителями. При этой передаче/репликации случаются мутации — и иногда вы получаете несовместимость части сигналов, leader и follower плохо понимают часть сигналов).

Если интерфейсные стандарты не соблюдаются, мир разваливается на плохо и непонятно как взаимодействующие (нет интерфейсов!) части. Взаимодействие происходит, но совершенно незапланированным образом, заканчивается неудачей. Поэтому нужно будет тратить много сил на налаживание взаимодействия там, где его нет, но оно нужно и убирать взаимодействие там, где оно есть, но оно не нужно.

Это налаживание взаимодействия в условиях отсутствия стандартов и когда системы уже созданы, называют «федерирование», см. стандарт ISO 11354-1:2011, framework for enterprise interoperability[435]. Как мы обсуждали в предыдущих подразделах, этот стандарт определяет три подхода к налаживанию взаимодействия/interoperability: федерированный/federated, унифицирующий/unified, интеграционный/integrated. Если разработчики взаимодействующих модулей используют при их создании стандарт взаимодействия/интерфейса, то это «унификация», а если они тщательно совместно разрабатывают оба модуля, проектируя их взаимодействие (что является более исключением, чем типовой ситуацией), то это «интеграция». При федерировании разработчики не соблюдают стандарты и не знают друг о друге, и это самый частый случай. И самый дорогой в реализации. Унификация — это ориентация на стандарты, поэтому вместо федерирования предпочитают унификацию, это обычно дешевле.

Интеграция — это очень редкий случай, и дорогой: вам нужно совместно

[435] https://www.iso.org/standard/50417.html

разрабатывать устройства и согласовывать интерфейс прямо при разработке, это довольно редкий случай. Скажем, электростанцию и электролампочку пришлось бы разрабатывать вместе, договариваясь о напряжении на выходе электростанции и входе электролампочки в рамках общего проекта. При взаимодействии через унификацию этого нет: вы просто соблюдаете стандарт интерфейса, лампочка и электростанция сопрягаются в России на 220 или 230 вольт[436] и не требуют совместной разработки, хотя в других странах сопряжение идёт на других значениях: 240 вольт, 110 вольт, и в тех сетях сопряжение тоже идёт через унификацию. Но если у вас лампочка на 220 вольт, а напряжение в городской сети 110 вольт, то вам нужно федерирование (например, нужно поставить повышающий трансформатор: адаптер для сетевого интерфейса).

Инженеры могут использовать прикладной интерфейс платформ, не интересуясь устройством этих платформ — лишь бы эти платформы осуществляли сервис, осуществляя его через описанный интерфейс. Это существенно упрощает задачу создания сложных систем, ускоряет и удешевляет её. Нужно хорошо разбираться в вариантах модулей и выдаваемых ими на их интерфейсах сервисах, но необязательно разбираться, как они устроены внутри, и как именно они внутри себя работают. А если этот прикладной интерфейс стандартизован, и есть конкуренция поставщиков модулей/продуктов/изделий/конструктивных частей, то они могут ещё и выбрать подходящий им по цене и характеристикам вариант реализации. И это происходит на много уровней вверх и много уровней вниз по инструментальному стеку, системное мышление рекурсивно.

Деньги обычно приходят от удачного и массового использования верхнего, самого прикладного уровня (конечно, для каждого уровня инструментального стека вышестоящий уровень является «прикладным», системное мышление применяется рекурсивно — то есть одинаково для каждого системного уровня, одинаково для каждой в нём системы). Посмотрите примеры явного использования рассуждений о платформе в стратегии фирмы NVIDIA — при описании структуры интеллект-стека[437], инструментального стека робо-такси[438], стека вычислительной инфраструктуры[439] и обратите внимание на тип диаграмм, который там использован.

«Перетряхивание» всего инструментального/платформенного стека, перестройка всех рынков идёт обычно тогда, когда меняется принцип физической реализации самого нижнего уровня конструктивов, то есть меняется реализация платформы самого нижнего уровня.

Например, когда в компьютерах поменялись механические или пневматические элементы на вакуумные электронные лампы, компьютеры стали масштабируемы и они начали напоминать уже функционально современные компьютеры, а не калькуляторы прошлых лет. Замена ламп на дискретную полупроводниковую технику позволила наладить массовый выпуск компьютеров и это разительно изменило компьютерную технику. Замена дискретной электроники на большие

---

[436] До 2003 года стандарт был 220 вольт, но сейчас 230 вольт, хотя в некоторых нормативных документах по-прежнему 220 вольт, ГОСТ 29322-2014 определяет стандартное напряжение 230В с возможностью использовать 220В. Электросети поставляют электроэнергию согласно действующего на сегодняшний день ГОСТ 32144-2013, устанавливающего напряжение 220В. https://ru.wikipedia.org/wiki/Сетевое_напряжение

[437] https://ailev.livejournal.com/1380163.html

[438] https://ailev.livejournal.com/1384766.html

[439] https://ailev.livejournal.com/1416697.html

полупроводниковые интегральные схемы опять перевернула весь компьютерный рынок со всеми надстройками программного обеспечения. Замена CPU на GPU перевернула рынок искусственного интеллекта. Замена людей-исполнителей на роботов-исполнителей переворачивает все производство. Эта особенность делает конструктивные/инструментальные/платформенные/модульные/продуктовые/аппаратные/софтверные стеки удобными для обсуждения стратегии компании.

При стратегировании обсуждают метод работы компании и предметы этого метода, в том числе какие из конструктивов на этих платформенных уровнях будет компания делать сама, а что закупать вовне, какие стандарты интерфейсов для этих конструктивов она будет устанавливать сама, а какие брать из уже готовых стандартов. Хороший пример такого стратегирования на системных уровнях инструментального стека для искусственного интеллекта даёт компания NVIDIA[440].

Моделирование: уровни платформенного стека

Приведите уровни платформенного стека системы в вашем рабочем проекте (не меньше трёх уровней).

Уровни стека

Прикладная функциональность против универсальной конструкции

Частой ошибкой в разработке систем является игнорирование явного функционального моделирования системы: документирование принципиальных схем, описаний взаимодействия подсистем, ответа на вопрос «как работает», «каким способом меняет состояние предметов метода». В электронике и электрике, а также работе с непрерывными/потоковыми производствами[441] (химические и энергетические предприятия) обычно принято рисовать такие принципиальные схемы (документируются чаще всего диаграммами), на которых явно обозначаются или функции/процессы, или ролевые/функциональные части/единицы системы, или даже и то и другое вместе.

Но этот метод разработки, предусматривающий обязательное рисование функциональных диаграмм, существует отнюдь не для всех видов систем. Так, в программной инженерии функциональные/потоковые диаграммы в большинстве проектов рисовать не принято, равно как не принято делать подобные текстовые описания на формальном языке моделирования, или даже на псевдокоде.

Например, программисты думают о функциональности своих программ, но принципиальных/концептуальных схем их работы программисты почему-то не делают, они держат обрывки этих принципиальных схем исключительно в уме — и поэтому в этих схемах нельзя найти пробелы, ошибки, поговорить о функционировании их систем с коллегами. Это не значит, что функциональных/потоковых нотаций не существует, например, data flow diargam[442]

---

[440] Болваны для искусственного интеллекта http://ailev.livejournal.com/1356016.html, NVIDIA как поставщик инфраструктуры для интеллект-стека, https://ailev.livejournal.com/1380163.html, NVIDIA как поставщик инфраструктуры для роботакси-стека https://ailev.livejournal.com/1384766.html

[441] https://en.wikipedia.org/wiki/Continuous_production

[442] https://en.wikipedia.org/wiki/Data-flow_diagram

и даже архитектуры потоков данных (dataflow)[443], лежащие в основе ставшего популярным так называемого «реактивного программирования» [444]/«обработки потоков»[445].

В исходном коде программ, тем не менее, эти «принципиальные схемы» с их потоками не видны: парадигма программирования чаще всего не предполагает выделенного показа функциональности частей программы, плюс эта функциональность существенно зашумлена деталями описания, достаточными для исполнения кода. В передовых кругах разработчиков софта это не так, и функциональные описания (чаще всего получаемые в рамках практики domain driven design, DDD[446]) документируются, но пока это ещё не общепринятая культура/практика, скорее приятное и редкое исключение из общей практики.

Этот взгляд на систему как некоторые потоки от входов к выходам — он оказывается важным и для предприятия как системы. Так, операционный менеджмент — это как раз функциональный (иногда говорят — логистический, «транспортный») взгляд на предприятие как потоки работ (workflow), потоки комплектующих и предметов поставки (ERP, enterprise resource planning работает именно с планированием этих потоков, если нет планирования потоков — то это не ERP в оригинальном его значении).

Неработа с функциональностью — типичная ошибка объединившихся в проекте инженеров и менеджеров. «Техпредприниматели» или корпоративные менеджеры беседуют с внешними проектными ролями/stakeholders и выясняют обычно их потребности, а вот концепцией использования, то есть основными сценариями использования целевой системы не занимается вообще никто — инженеры сразу переходят к концепции системы, которая ещё непонятно что должна делать (непонятна её функция в надсистеме). В том числе в сценарии использования входят, заметим, не только happy path, то есть поведение системы, когда всё идёт хорошо и нет неожиданностей.

Но неожиданности есть всегда, и о них нужно заботиться заранее. Инженер-проектировщик в таких случаях создаёт модульную конструкцию (то есть предусматривает в конструкции какие-то заменяемые модули с различными ожидаемыми важными характеристиками в их сервисах в момент использования, но у этих модулей будут какие-то одинаковые интерфейсы подключения). Функциональность там предусматривается «на вырост», как сервисы модулей, реализующих подсистемы, которые, по его мнению, могут дать всевозможные функции на уровне системы без дополнительных доработок продукта, если те вдруг потребуются.

Для этого инженер соединяет через типовые интерфейсы из самых ходовых и простых стандартов типовые конструктивные элементы в надежде, что через восемь итераций всё как-то само-собой с функциональностью утрясётся (будет ведь обратная связь от недовольных клиентов!), и проект в конечном итоге получится успешным. Но это неэлегантно, не lean: добавляется лишняя функциональность, система за счёт этого становится обычно дороже, изготавливается дольше, но нужной функциональности может в первых итерациях и не оказаться, первые

---

[443] https://en.wikipedia.org/wiki/Dataflow
[444] https://en.wikipedia.org/wiki/Reactive_programming
[445] https://en.wikipedia.org/wiki/Stream_processing
[446] https://en.wikipedia.org/wiki/Domain-driven_design

итерации потребуют переделок/re-work, они не будут представлять MVP.

Заметим, что примерно так и срабатывает «испорченный телефон»: беседуют с клиентом одни люди (которые не обучены системной инженерии), они формулируют и жёстко фиксируют (не просто документируют, а именно что «замораживают») требования с их обязательностью (а не в форме гипотезы о том, что нужно клиенту от системы), а разрабатывают потом систему совсем другие люди (которые обучены, но контакта с клиентом у них нет — и концепция системы поэтому оказывается плоха, гипотезу проверить удаётся только после того, как предъявили клиенту уже готовую систему с ошибочным набором функций).

Ошибку отсутствия функциональных частей системы в концепции системы тоже легко найти: в концепции системы при такой ошибке нет отражения предметной области, для которой целевая система осуществляет сервис, нет предметов сервиса, проходящих какие-то состояния в ходе обработки. Сервис назван (уж как могут его назвать люди, занимающиеся целевой системой, а не надсистемой), а предметов метода в надсистеме (что выполняет модуль как подсистема в составе надсистемы) — нет.

Например, в концепции информационной системы магазина появляется просто «база данных» (сверхобобщение для чего угодно), а не «цены товарных позиций, правила предоставления скидок, история покупок». Понятно, что всё это должно где-то храниться, но не факт, что в нарисованной программистом на первой же его диаграмме «базе данных» (часть, например, может браться через API какого-то сервиса, предоставляемого через облако, а правила представления скидок могут храниться в настроечном файле или вообще оказаться искусственной нейронной сетью), и не факт, что даже «база данных» будет одна.

Вот почувствуйте разницу:

- производитель определяет сервис насосного агрегата как «повышение давления», а клиенту нужна функция «подъём воды на верхние этажи» (и нужно ещё прикинуть, какое там повышение давления для этого нужно и что ещё нужно сделать, например, поставив два насосных агрегата, чтобы поломка одного конструктива не приводила к перерыву в функционировании),
- производитель определяет сервис молотка как «удары», а клиенту нужна функция «загнать гвоздь в доски», хоть «ударами», хоть «вдавливанием»
- производитель определяет для своей программы сервис «вычисления» (и надеется, что встроенный язык программирования даст «любые вычисления»), клиенту нужна от программы функция «расчёт калибровочной кривой», но чтобы «из коробки», без написания программ на внутреннем языке.

Функциональные диаграммы будут содержать названия функциональных частей или функций, а конструктивные/модульные/продуктовые диаграммы, замаскированные под функциональные — всё равно будут содержать названия самых общий сервисов, и указывать обычно не потоки и порты, а стандартизованные интерфейсы как места/каналы оказания сервисов.

Объяснить программисту необходимость делать функциональные описания в составе концепции системы трудно, ибо он мыслит компонентами/модулями/продуктами/фреймворками/платформами, которые, по

его мнению, абсолютно универсальны и поддержат «**любую функцию». Любую-любую?** Помним, что **вы должны удваивать любой встречающийся в описаниях систем квантор всеобщности — «каждый», «всякий», «любой», «все» и т.д. и ставить после этого вопросительный знак, сомневаясь в этой всеобщности.** Всеобщности, увы, не случится. Но разочарование будет не в начале проекта, а поближе к концу — когда окажется, что универсальных прикладных софтверных систем не бывает, так же, как и любых универсальных систем «железных», «человеческих», «социальных» и т.п.

Архитектуру (архитектурные решения, которые относятся прежде всего к делению на модули и организацию связи между модулями для поддержки приемлемых значений более-менее одинаковых для самых разных систем архитектурных характеристик) программист признаёт, а концепцию системы — нет, функциональная декомпозиция у него идёт крупноуровневая «в голове», а затем сразу в исходном коде, «рабочке», и это ошибка в методе работы. При этом важность архитектурной работы уже признаётся в крупных проектах, само понятие архитектуры претерпело существенные изменения (подробней в руководстве по системной инженерии) и работа архитектора ставится в оппозицию работе разработчика. Даже появляется новая терминология для компонент/модулей, отражающая архитектурные соображения: **архитектурный квант** как автономно вводимый в эксплуатацию конструктив, внутри которого обеспечена высокая функциональная связность[447]. Дальше мы ожидаем, что из архитектуры больших корпоративных приложений эти рассуждения про то, как делить систему на конструктивы, в том числе архитектурные кванты — перейдут и в «железную» инженерию, и ещё много куда. Грубо говоря, автомобиль с прицепом имеет довольно много конструктивных частей, но вот автомобиль и прицеп тут будут архитектурными квантами, их будут вводить в эксплуатацию автономно.

Даже если отвлечься от архитектурной работы (выполняемой отдельной ролью архитектора по методам работы архитектора, а не разработчика, причём это работа над конструкцией, а не над функциональностью), у программистов-разработчиков обычно случается много ошибок как раз по линии учёта функциональности. Разработчики как бы действуют как «недоархитекторы», налегая на «универсальные модульные конструкции», но забывая детально разобраться с функциональностью — но за учёт функциональности в конечном итоге ответственны именно разработчики, а не архитекторы.

Вот пример, иллюстрирующий происходящее в голове (или компьютерах, или головах и компьютерах, ибо не все разработчики сегодня — люди) у разработчиков на примере более понятной работы с «железной» системой — но того же класса, что и программы: кажущейся суперуниверсальной по форме, но весьма конкретной и уникальной по содержанию. «Коммерсанты» на заводе подтащили какие-то свои идеи продукта инженеру-разработчику в «техническом отделе», а инженер-разработчик подумал-подумал и выдал какие-то идеи, которые «коммерческий отдел» оформил как предложения для нового проекта:

> У нас стоит задача тонкого химического синтеза диметилфторидметилхлорпентилбензолтитана для задач фармацевтической промышленности. Известно, что его трудно синтезировать, и при синтезе

[447] https://www.oreilly.com/library/view/building-evolutionary-architectures/9781491986356/ch04.html

получается много примесей, от которых потом люди умирают. Поэтому мы предложим аптекам такой чистый продукт, от которого люди не будут умирать, а маркетинг будет оригинальный: через уборщиц медицинских учреждений, которые будут подбрасывать наши буклеты врачам, а также задушевно беседовать с пациентами в очередях ко врачам. Для того, чтобы получить чистый диметилфторидметилхлорпентилбензолтитан, мы будем использовать четыре стальных реактора, соединённых трубами диаметром 56мм, последняя труба с тефлоновым покрытием. Второй реактор закажем внешним контракторам. Чистоту получающегося диметилфторидметилхлорпентилбензолтитана будем отслеживать при помощи независимой экспертной службы. Все четыре реактора должны будут пройти проверку на выдерживание давления в 156 атмосфер.

В этом описании нет никаких идей, какие именно примеси имеют максимально вредный эффект, как получить чистый именно от этих примесей (а не от любых — безвредные ведь не мешают) диметилфторидметилхлорпентилбензолтитан, какие нужны исходные вещества и доступны ли они, или их тоже нужно синтезировать в рамках проекта, далее химические реакции, начиная с доступных веществ — т.е. подробное разбирательство того, что происходит в реакторах.

Теперь представим инженера-разработчика в техническом отделе, которому пришедшие к нему представители коммерческой службы (маркетологи, эккаунт-менеджеры) говорят, что ему нужно будет решить задачу химического синтеза, но непонятно ещё какую. И они когда-нибудь потом, но вот не прямо сейчас наймут химика, который всё уточнит. Инженер соглашается и берётся за дело: ему сразу понятно, что нужны будут реакторы и трубы. Реакторов непонятно сколько, ибо непонятно, сколько там в синтезе будет химических реакций. Поэтому лучше бы этих реакторов было побольше. Он выбирает цифру 4, как «не очень мало, но и ещё не очень дорого», догадывается, что в этом синтезе может быть хитрая эндотермическая реакция, но сам он для такой реакции реактор вряд ли спроектирует. Так что этот «хитрый реактор» он отдаёт проектировать внешним контракторам и честно об этом пишет. Хотя внешние контракторы уже спрашивали, что там с массами и температурой, ответ пока никто не может дать, поэтому контракт планируется без разговора о сценариях использования — но контракторы, разумеется, от заказа не отказываются они говорят, что «сделаю любой реактор». Слово «любой» никого не отпугивает, ибо нет химика, который бы сказал, что же будет в этом реакторе происходить. Есть только инженеры по железу, хорошо разбирающиеся в марках стали и сварке, и маркетологи, беседующие с врачами и фармацевтами. Итог работы инженер и выражает в вышеприведённом абзаце весьма и весьма абстрактное описание системы, которое вряд ли приведёт к успеху, почти все принятые впопыхах решения потом придётся принимать заново: нельзя обсуждение конструкции ставить перед обсуждением функциональности.

В случае железа реактора и труб всё тут очевидно: вероятность выпуска чистого диметилфторидметилхлорпентилбензолтитана в этой ситуации равна нулю. Но **в случае софта почему-то кажется, что задачу так решать можно. И вот пишут про «сервера» и «базы данных», но каким образом там будет происходить обработка какой информации, и с чего бы эта обработка вдруг оказалась даже не лучше конкурентов, а вообще возможной, этого вопроса не возникает. Ровно как в предыдущей задаче, где нет химика, есть только сварщики и спецы по корпусам высокого давления. Они точно**

сделают софтверный «универсальный реактор», но вот что там с чистотой «информационного вещества» или даже просто запуском нужной «информационной реакции» (слова «обработка информации» или «расчёт» так и пишутся, без малейших уточнений, какой именно), и получить исходные данные (или исходную информацию? данные! ибо информация что-то означает, а данные тут как сырьё — можно игнорировать их содержание) — это не к ним.

В проекте получения чистого диметилфторидметилхлорпентилбензолтитана химика пока нет, про химию разговор поддержать некому, равно как сообразить, как же эта химия влияет на примеси, а примеси на здоровье пациента. **За разговорами про конструктивы реакторов и труб к этому моменту уже можно забыть, что сама ситуация про медицину, потребности тут в здоровье пациента (целевая система тут в пациенте: окружают её в момент работы живые ткани, а реакторы и трубы — в системах создания, «сбоку» целевого системного разбиения/окружения, в другом времени, во времени создания).** Делать, конечно, приходится железную конструкцию «нашей системы», но работать-то нужно функциональной системе, которая должна выполнить свою функцию в графе создателей! Нужно протянуть строгую логическую цепочку от целевой системы (синтезируемого вещества) к нашей системе (из труб и реакторов) — через функциональное описание, то есть химические реакции, которые описывают функциональность целевой системы (как она себя ведёт в пациенте, время использования), и далее к ведущим к появлению вредных примесей отклонениям от режимов проведения реакций в реакторе (время создания). При этом предметная область функциональных (химических) рассмотрений времени эксплуатации — по факту клиентская, а итоговые конструктивные (труб и реакторов) рассмотрения — предметная область изготовителя-контрактора, времени создания. По сути дела, это рассуждение — даже не очень полный проход по системной мантре.

Возвратимся к программированию и типичным ошибкам разработчиков, которые недостаточно озабочены функциональными описаниями. Увы, с первого раза получить функциональное описание, в котором рассказывается о том, что содержательно происходит с клиентской информацией в корпоративных информационных системах, обычно не удаётся. Нужна пара-тройка итераций с ролевыми/функциональными частями системы и их назначением/поведением (функциями), но зато после её получения уже вполне осмысленно и существенно корректируется структура компонент/модулей/платформ/продуктов/микросервисов, этой структурой занимается уже главным образом архитектор.

В случае «железных» проектов тоже нужны итерации с постепенным уточнением описаний на базе получаемой в этих итерациях дополнительной информации. В нашем примере после подробного разбирательства с функциональными описаниями будет изменено число реакторов, уберутся лишние трубы, будут проведены дополнительные исследования по наличию на рынке дешёвых датчиков чистоты продукта, найдут контрактора, который пишет управляющий софт для закрытия-открытия задвижек на реакторах, чтобы управлять ходом синтеза на основе показаний датчиков и т.д.

Сначала нужно разобраться с функционированием: что происходит в тот момент, когда система работает/функционирует (слово «функционирует» тут не случайно!

Интересуют функции и функциональные части! Это предмет интереса тех ролей, которые отвечают за работу системы!), а не когда её собирают из размещаемых где-то конструктивных частей продуктов/модулей/изделий/платформ. Потом в порядке модульного синтеза всё равно нужно будет рассмотреть и конструктивные части, и размещения. Но это всё потом, а для начала нужно сделать качественную функциональную декомпозицию, функциональный/системный анализ.

Рассмотрение функционирования сначала, а конструкции потом — это важная часть системного мышления: сначала нужно обсудить подробно, зачем эти конструкции и как они будут работать, а уже потом рассматривать, из чего они будут собраны.

В современной инженерии, тем не менее, не принято делать функциональную декомпозицию сверху-вниз, ибо при этом очень тяжело преодолевать последствия тех решений, которые придётся менять, когда за дело примутся специалисты по предметной области той или иной функции. Поэтому принято обычно визионерам (эта роль чаще всего исполняется на должностях продукт-менеджеров) приходить с какими-то идеями о том, что и с каким приоритетом надо добавлять к сервисам/функциям системы в её окружении, а затем за дело берутся архитекторы, которые по-грубому прикидывают, какие конструктивы вероятнее всего могут поддержать в их взаимодействии эту функциональность — но не делают сами функциональную декомпозицию. Далее они приглашают для детального рассмотрения функциональности сразу на нижнем уровне функциональной декомпозиции людей из тех команд, которые занимаются отобранными ими модулями. И итерации идут уже с ними. Тем самым грубая функциональная декомпозиция и модульный синтез делаются архитекторами и визионерами (в лице продуктовых менеджеров) на верхних уровнях, а вот детальные рассмотрения сразу идут на нижнем уровне, силами людей, хорошо разбирающихся в предметных областях этих функций/методов/сервисов, знающих особенности предметов этих методов.

Конечно, эти рассуждения относятся к инженерии чего угодно, необязательно «железа» или софта. Например, можно рассмотреть применение этих рассуждений к танцам. Просто махание ногами и руками (с запасом! Побольше махов, и сами махи пошире!) никого не устроит. Каждый мах должен быть понятен, зачем — какая там выражается эмоция или сигнал ведения: что этим махом хотят сказать, что продемонстрировать, какой ожидают результат. Тогда и мах будет подобран соответствующий (резкий, плавный, объёмный или короткий, рукой или ногой — ровно под необходимую функцию, а не «стандартный с бесчисленными вариантами из-за параметризации»). Если вам не нужно делать для вашего танца мах прямой ногой к уху (да ещё «с запасом под любую хореографию, с которой вы можете встретиться», как делается при создании и развитии танцоров классического балета, это примерно десяток лет тренировок), то вы можете сэкономить много лет жизни. Менее универсальные махи «под функцию», с учётом предметной области конкретного танца, тренируются много быстрее. Универсальных танцоров, музыкантов, инженеров, менеджеров, учителей не бывает — поэтому сначала надо задаваться вопросом «что будем делать», и только потом готовить не только «вообще», но и специализировать подготовку для выполнения конкретной функции. Так, в системном фитнесе можно выделить 4 проекта, которыми он

занимается[448]:

- Проект-1: Обучение изготовлению тела — стажировка по системному фитнесу (1 месяц), создание «тренера::создателя себя»
- Проект-2: Изготовление тела - регулярные/специальные занятия (самостажировка, 12 месяцев), эксплуатация тренера, создание заготовки «тело как универсальный модуль».
- Проект-3: Использование тела — изучение спорта/танца/борьбы (6 месяцев), используется универсальный конструктив, продолжение работы сам-себе-тренера, создаётся прикладной конструктив, выполняющий функции в прикладной предметной области.
- Проект-4: Отдельная задача в выбранном спорте (3 месяца), развитие прикладного конструктива сам-себе-тренером до новой потребной приоритетной функциональности (вплоть до того самого «мах прямой ногой к уху с запасом», хотя тут может и не хватить 3 месяцев, но уж точно это будут не 10 лет, минимум впятеро быстрее).

Польза от первого проекта проявляется не сразу, поэтому от стажёра требуется инвестиция: вложение ресурсов «здесь и сейчас», когда имеются риски не дойти до успешности «тогда и там». Но ровно это же происходит в любых других проектах: заказывать систему приходится «здесь и сейчас», вкладываться в её проектирование и изготовление, но всегда есть риски не дойти до успешности «тогда и там».

Эта совместная поэтапная работа над универсальными (сырьё) и прикладными (результат применения метода) конструктивами и их универсальной и прикладной функциональностью важны как общий принцип для инструментальных стеков. Так, можно рассмотреть ещё один учебный проект, на базе интеллект-стека. Так, освоение системного мышления можно представить в похожем виде прохождения четырёх похожих проектов с переходом от универсального мышления к прикладному:

- Проект-1: Обучение созданию у себя универсального мастерства системного мышления — освоение руководства по системному мышлению (1 месяц, объём как четыре пятых «Войны и мира», но одолеть можно, много запомнится и даже поймётся, но устойчиво удерживаться в памяти не будет. Это уровень квалификации специалиста-аналитика: «знаю материал, но использовать не получается, ибо скорости не хватает, много не замечаемых быстро ошибок, затраты на мышление оказываются больше выгод»). По факту это создание тренера по системному мышлению для самого себя, то же самое, что и в стажировке по системному фитнесу.
- Проект-2: Изготовление сам-себе-тренером, полученном в предыдущем проекте, универсального своего мастерства системного мышления, достижение беглости — регулярные/специальные занятия («самостажировка», 12 месяцев, но на реальных проектах, а не на учебных упражнениях. Впрочем, учебных упражнений в нашем руководстве нет, всё только на рабочих проектах)
- Проект-3: Использование системного мышления для изучения какой-то прикладной инженерии, менеджмента, каких-то других методов прикладной работы (3-6 месяцев каждая). В программе рабочего развития инженеров-

[448] https://www.youtube.com/watch?v=k7oyl99mMSE&t=13020s

менеджеров это освоение работы по руководствам системной инженерии, инженерии личности, системного менеджмента.
- Проект-4: Отдельная задача инженерии-менеджмента (от нескольких минут до 3 месяцев, «подумать и сгенерировать альтернативы, выбрать лучший метод работы, спланировать ресурсы, перейти к действиям и выполнить план»).

Мыслить только про «универсальные в плане функций конструкции» нельзя, универсальных конструкций не бывает. Отсутствие мышления о прикладной функциональности нужно как-то осознавать и исправлять. А это отсутствие мышления о прикладной функциональности заметно только в случае наличия системного мышления — оно позволяет заметить то, о чём вы забыли подумать.

Системное мышление управляет вашим вниманием, прежде всего оно направляет это внимание на функциональность систем: что делает система в составе надсистемы (первый вопрос), что делают подсистемы в составе системы (второй вопрос), только третий вопрос про «из чего сделана система». И каждый раз проверяется, на каком вы уровне абстракции работаете: если нет слов о предметной области, то вы делаете что-то неправильно. Если завод выпускает «оборудование» или «изделия», то вы в беде: кто-то подумает, что это авиалайнеры, а кто-то — что это домашние хлебопечки. И дальше будут относиться соответственно, а вы будете удивляться, почему вас так неправильно понимают (а ваш завод или «завод» что выпускает? Какой сервис делает? Какая это предметная область, или вы ударились в универсализм — и тем самым уже проиграли?).

Внимательно посмотрите на свой текущий проект: не пропущены ли в нём функциональные рассмотрения? Если пропущены, то помним: **без специалистов предметной области (subject matter expert) надсистемы, а потом и целевой системы, и её подсистем, и вашей системы (если она из систем создания) вам не разобраться. Общайтесь, общайтесь, общайтесь. Системные мыслители много общаются с проектными ролями, отвечающими за самые разные прикладные практики, для этого они имеют универсальное/трансдисциплинарное/фундаментальное системное мышление, но они знают, что его результаты всё равно надо довести до какой-то прикладной системной области.**

Вопросы. Наличие описания функциональности

Программист долго рассказывает про его целевую систему как «прозрачный ящик»: она у него состоит из компьютеров, контейнеров с данными и программами, хранилищ для хранения бэкапов, баз данных, средств балансирования нагрузки и многого другого. Чего не хватает в этом описании, чтобы серьёзно отнестись к его рассказу?

--- всё в порядке, все программисты рассказывают именно об этом ### Нет, не всё в порядке: это всё описания времени создания программы, а не времени эксплуатации. Нет ни одного слова из предметной области, с которой работает система программиста!

+++ не хватает описания функциональности: для чего это всё ### Да, не хватает

описания времени работы программы, и это описание должны включать слова из предметной области, с которой работает программа как целевая система программиста (финансы, или товарные потоки, или радиация в активной зоне ядерного реактора, или ещё что-то, что выходит за пределы времени создания программы)

--- не хватает описания стоимости, ибо без указания того, сколько всё это стоит, невозможно сориентироваться. ### Нет, стоимость и её приемлемость понадобятся только после того, как будет понятно, для чего вся описываемая конструкция работает: описана функциональность. После этого можно будет оценить, уместна ли описываемая конструкция, где её размещать, и сколько это будет стоить в разработке и потом в ходе эксплуатации. Но вот что происходит при эксплуатации, хорошо бы знать с самого начала. Нужны слова про то, что делает система программиста в ходе её работы с объектами, описывающими предметную область назначения этой программы (а не предметную область программирования).

Оргзвенья

В системах из людей конструкты — это прежде всего сами люди, с их аудио, зрительным, тактильным и прочими интерфейсами. Функции людей даже не делается попыток обсуждать, равно как смысл и содержание информации или изменений окружающей среды, которые поступают к людям и уходят от них через интерфейсы человеческого тела. Люди — это универсальные создатели. Помним определение универсальных создателей/по David Deutsch, оно по факту расширяет определение универсальных компьютеров по Alan Turing: если им дать достаточные ресурсы и время, они могут выполнить любое допустимое принципами физики преобразование физического мира точно так же, как универсальный компьютер может выполнить при наличии достаточно времени и памяти любое возможное вычисление. В том числе люди могут при недостатке ресурсов реплицироваться, в том числе люди могут сначала создать инструментарий, а уже затем выполнить необходимое преобразование, работая коллективно и используя инструменты.

Это мы проводим про людей конструктивное/продуктовое рассмотрение, т.е. рассмотрение «как сделать», как составить систему из людей, уж какие они есть перед началом работы — ибо про функциональность мы ничего не говорим, считаем, что поскольку люди универсальны как создатели, они могут быть как-то обучены на выполнение нужных функций. По большому счёту, нас волнуют даже не люди, а универсальные агенты — роботы с AI достаточной силы нам тоже подойдут для рассуждений.

У интеллектуальных агентов есть особый тип отношений, который обозначает возможность распоряжения теми или иными средствами производства, или даже другими людьми: речь идёт о полномочиях собственности, полномочиях давать поручения другим людям, полномочиях принимать работу других людей, полномочиях обмениваться какими-то предметами или сервисами/услугами (включая тот случай, когда речь идёт об «универсальном предмете» — деньгах).

Чтобы получить затребованное внешнее поведение от агента, нужно иметь соответствующие полномочия, которые им признаются. Агенты, договорившиеся о полномочиях по распоряжению друг другом и их средствами производства, о взаимных обязанностях, называются **организованными**, а система из таких людей — **организацией/оргзвеном, оказывающим сервисы для других**

организаций/оргзвеньев. Термин «организация» чаще всего используется по отношению к вершине конструктивного разбиения системы из интеллектуальных агентов (прежде всего людей) и средств производства. Эти организации — предприятия/фирмы/компании. В случае государственных и общественных/некоммерческих организаций используются другие имена — органы, агентства, учреждения, ассоциации, фонды и т.д..

Конструктивы, из которых созданы организации (то есть отдельные агенты или организованные группы агентов с их инструментами) мы будем называть не «подразделение», а более общим словом — **оргзвено**.

Слово «оргзвено» лучше слова «подразделение» тем, что в конструктивах оргструктуры обычно не показываются вре́менные организованные группы агентов и их инструментарий (средства производства): проектные группы (проекты ведь по определению временны!), а также всевозможные советы, комитеты, кружки, комиссии и т.д. Но когда вам говорят, что «для запуска нового проекта нужно решение научно-технического совета», то вам нужно как-то учесть такой орг-конструктив как научно-технический совет, который явно есть в конструкции организации, но никто его не называет «подразделением». Структура «подразделений» обычно удобна для выдачи зарплаты, учёта числа сотрудников, но если сотрудник начинает работать в нескольких разных наложенных друг на друга организациях (в отделе, в каком-то совете и паре проектных групп), то это невозможно обсуждать в терминах подразделений: советы и проектные группы не входят в штатные расписания. Если мы обобщаем понятие подразделения до всех полномочных принимать какие-то решения и выполнять связанные с этими решениями действия подразделений и включим вот такие «коллегиальные органы», собирающиеся на отдельные заседания (да и те часто проводимые через почтовые рассылки или обсуждения в чате) и постоянно работающие проектные команды, набранные из сотрудников подразделений «мимо официального штатного расписания», то мы как раз получим «организационное звено»/оргзвено::конструктив — тип части, из которой сконструирована организация.

Минимальное оргзвено тем самым — один человек или один AI-агент (и даже неважно, это AI-бот, реализуемый где-то в дата-центрах, или AI-робот, у которого, заметим, тоже «мозги» могут оказаться в каких-то датацентрах, а в мире он будет представлен даже не одним телом, а несколькими, необязательно антропоморфными). Поскольку агенты не изменяют физический мир голыми руками, и не пользуются голыми каналами восприятия, то минимальное оргзвено будет включать не только агента, но и его инструменты, которыми тот полномочен распоряжаться. Учитывая то, что агенты в организации должны сотрудничать, они называются сотрудниками. Минимальное оргзвено — один сотрудник с приданным ему в распоряжение оборудованием/инструментами (помещения и мебель будем тут тоже считать оборудованием/инструментами).

Дальше эти минимальные оргзвенья/сотрудники/агенты (некоторые из них — люди) из одного агента объединяются на основе известных им полномочий в оргзвенья следующего уровня, и так — пока оргзвено самого высокого уровня не окажется организацией в целом (предприятием или учреждением, или ассоциацией, или фондом и т.д.). Иногда первичное объединение оргзвеньев называют **командой,** обычно в команде 5-12 агентов, а сейчас часто говорят о команде как «столько людей, сколько можно за обедом накормить двумя большими американскими

пиццами» (two pizza team), как тут поступать с командой роботов, людей и роботов — непонятно, скорее всего так и считать, что речь идёт о 5-12 интеллектуальных агентах, иногда команда — это 2-5 агентов (в Tesla говорят про такие маленькие бригады — mob/банда).

Иногда вводят ещё один уровень — **коллектив**, это «команда команд». Но помним, что когда разные уровни иерархии (тут — иерархия часть-целое для оргзвеньев) называют разными именами — это кажется удобным поначалу, а потом оказывается неудобным по мере нарастания изменений в количестве уровней для тех или иных ветвей этой иерархии. В риторических (для убеждения людей) целях могут и население страны назвать «командой», чтобы подчеркнуть принадлежность к «маленькой организации» и готовность выполнять согласованные планы работы (при этом желание, конечно, отличается от возможностей — всё, что мы знаем про автономных агентов-людей и аналогичных им роботов уже не полностью подходит к организациям, хотя и те, и другие — «лица» и «стороны», берущие на себя ответственность за выполнение каких-то планов. Это и означает «организованность». А вот с **сообществами** (агентами, занимающимися какими-то методами работы и обменивающимися опытом по практикованию этих методов работы, например, community of practice[449]) уже не так, и с обществами (агенты, которые компактно проживают на каких-то территориях, но вот общность каких-то методов работы даже не обсуждается) — тем более не так. Так что разговор об оргзвеньях заканчивается на не очень далёких от одной организации системных уровнях — а именно тех уровнях, где можно какой-то явной договорённостью (а не просто силовым принуждением типа «заплатил налоги — и спи спокойно») говорить о выполнении обсуждаемым конструктивом как системой работ по какой-то стратегии, то есть работ по каким-то договорённым методам по изменению состояния предметов этого метода, с учётом разделения труда между членами организации.

От сообществ и обществ этого ожидать мы не можем, но вот от производственных коопераций, бизнес эко-систем вокруг маркетплейсов как платформ, организованных политических движений и партий — можем, ибо у них есть механизмы договориться об организации труда (разделиться на агентов, выполняющих какие-то отдельные роли, то есть выполняющих работы по каким-то методам, а также договориться о том, каким образом они получают понимание того, какие работы с какими предметами методов по каким методам должно быть выполнено, чтобы в итоге этой работы получить успешную целевую систему).

На уровнях расширенных поэтому совместные действия по координации использования ресурсов осуществляются на основе прямых соглашений создания «расширенных предприятий» (extended enterprise, организации, которые сами состоят из организаций, договорившихся о выполнении каких-то общих работ). Например, атомную электростанцию сооружает расширенное предприятие — сотни и тысячи предприятий-контракторов, а не только одна строительно-монтажная организация («сооружение» — это и есть «строительство» плюс «монтаж сложного оборудования»). У строительно-монтажной организации атомной электростанции есть контракты с её подрядчиками, в этих контрактах определено, кто за что отвечает в сооружении атомной станции: не на уровне отдельных людей, а сразу на уровне организаций.

---

[449] https://en.wikipedia.org/wiki/Community_of_practice

Вся организационная (конструктивная) структура и интерфейсы между оргзвеньями (конструктивами организации) обсуждаются не в связи с тем, как что-то в оргсистеме сделать получше в прикладной предметной области (время работы организации/оргзвена), а именно в связи с тем, «как организоваться» — кто чьи поручения будет выполнять, что и когда будет закуплено в качестве средств производства и кто этим будет распоряжаться. Предметом модульного/организационного обсуждения в организации/оргзвеньях является организовывание (достижение состояния организованности, когда понятны полномочия всех сотрудников). Почему так? Оргзвенья рассматриваются как агенты и коллективные агенты, причём универсальные. Это означает, что можно им назначить другие функции, поручить выполнение работ по другим методам, перенаправить к ним поток каких-то других предметов метода.

Сервисы оргролей, назначаемые на оргзвенья (как штатные подразделения, так и «коллегиальные органы», временные проектные группы — проектные команды, рабочие группы, подразделения, советы, комитеты, комиссии и т.д.), оказываемые ими на их организационных интерфейсах, обсуждение интерфейсов и интерфейсных модулей/каналов (почта, issue tracker, окошко кассы, лента конвейера и т.д.) для взаимодействия с ними, являются типичным предметом обсуждения в организациях. Эти сервисы в хороших организациях пытаются стандартизовать (регламентировать: подвести под стандарт уровня предприятия, стандарт опубликовать), сделать из них какую-то оргплатформу с опубликованным интерфейсом. Основная тут проблема: в крупных организациях централизация такой регламентации приводит к тому, что люди работают без соблюдения этих регламентов — просто потому, что им неудобно пользоваться документами и операционным софтом, которые достаточно далеко лежат от их рабочего места. Решением тут является децентрализация работы с регламентами и чеклистами. Подробно как это делать — в руководствах по методологии и системному менеджменту. Впрочем, и всё остальное про оргзвенья — это руководство по системному менеджменту, оно как раз о том, как создавать организации, менеджмент — это создание и развитие (включая эксплуатацию) организаций, то есть инженерия организаций.

Организация/предприятие буквально составлено, собрано, сделано из своих оргзвеньев — это конструктивные, а не функциональные части предприятия/организации. Из обсуждения организационной диаграммы не скажешь, как работает предприятие, какие функции выполняют оргзвенья предприятия. Но скажешь, «из чего сделано» — из каких подразделений.

Именование оргзвеньев — это рассмотрение «как сделано, из чего собрано» отражает этот конструктивный, а не функциональный аспект. В именах самые общие представления об оказываемом сервисе, вне связи с назначением/функцией этого сервиса. Цех №5, палата №6 — простые номера в именах, из которых не следует функций, являются типичными указателями на конструктивное рассмотрение. Самые общие слова без указания предметной области (как «база данных» и «сервер» в информационных технологиях) тут такой же признак («инжиниринг», «юридический отдел», научно-технический совет, проектная группа). Примерно так же могут быть в именованиях уточнения аспекта размещения — но всё равно ничего нельзя сказать о функциях: «Вологодский филиал» понятно, что оргзвено, и понятно, что в Вологде. Но чем занимается? Каким методом какие предметы метода он меняет, какие у него сервисы как у

системы? Это всё надо узнавать из других аспектных описаний.

Функций, исполняемых конструктивными элементами предприятия/оргзвеньями, может быть множество, и самых неожиданных. С другой стороны, если «подразделение переезжает в другое здание», то это описание связи ипостасей конструктивной и размещения. «Подразделение будет работать новыми методами» — это обсуждение связи функциональной и конструктивной ипостасей подразделения.

Моделирование: назначение оргзвеньев на оргроли, выполняющие практики

Какие функции/сервисы/«методы работы» каких оргролей выполняются какими оргзвеньями вашего предприятия? Заполните табличку по примеру первой строчки, десять практик.

Номер п/п	Метод работы	Оргроль	Оргзвено
Пример	Бухгалтерский учёт	Бухгалтер	Бухгалтерия
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			

Необходимость хорошей модульности

Система должна собираться из модулей по принципу подводной лодки, в которой все отсеки делаются оптимально автономными — если один из них будет затоплен, это не будет означать затопления подводной лодки. Но если сделать полную автономность (например, не предусмотреть люков между отсеками), то это уже не сможет быть подводной лодкой — не взаимодействующие через интерфейсы части конструкции не породят эмерджентности, новых свойств, которых не было у отдельных частей (например, отсеков подводной лодки). Чересчур связная система «тонет», а совсем уж мало связная — вообще перестаёт быть системой, проявлять ожидаемые свойства. Это верно и для «железных», и для программных, и для организационных, для всех систем.

В системе все части системы взаимосвязаны, в системном окружении они ведут себя совершенно не так, как могли бы вести себя без системного окружения. Если у нас нет чётко выделенных и хорошо описанных интерфейсов, через которые мы контролируем все взаимодействия в системе, то любая попытка улучшить модуль

может привести не только к улучшению этого модуля, но и дать побочный эффект: ухудшить систему в целом через влияние на другие модули по неотслеживаемым связям. Если интерфейсов не хватает, то системы вообще не получится.

Вот графики компьютерной симуляции зависимости падения стоимости разработки от улучшения n отдельных модулей, при d связей каждого из них[450]:

Видно, что если связь одна, то стоимость разработки с каждым улучшением падает быстро по экспоненте, слабо зависящей от числа модулей. Но если связей больше, то ошибки от локальных улучшений распространяются по этим связям и снижение стоимости разработки от улучшений в отдельных модулях идёт плохо. Каждая связь имеет свою системную цену, и не было бы этой цены, мы не имели бы даже в живой природе «органов», «клеток» и других конструктивов/модулей[451].

Модульность человеческого организма получена эволюцией, и она не идёт ни в какое сравнение с модульностью современного компьютера или смартфона, собираемого из стандартизованных деталей с чётко прописанными интерфейсами. Поэтому слабомодульный человеческий организм не очень понятно, как улучшать или даже просто лечить (но и тут есть модульность! Пересадка органов работает!). Компьютер или смартфон улучшать конструктивно и лечить-ремонтировать много проще, ибо модульность компьютера или смартфона тщательно продумывается.

Интерфейсами нужно управлять: тщательно их проектировать, учитывать их наличие, выявлять паразитные (то есть незапроектированные) интерфейсы и принимать меры по их устранению. Через эти паразитные интерфейсы в системах может распространяться вредное, ошибочное взаимодействие. **Оптимальная модульность, понимаемая как оптимальная сборочная структура из конструктивов, соединённых через качественные интерфейсы — это залог возможности автономного улучшения отдельных модулей, залог качества работы всей системы.**

---

[450] http://www.pnas.org/content/108/22/9008.full
[451] http://arxiv.org/abs/1207.2743

Интерфейс — это не просто соединение, это соединение через какой-то вполне определённый физический канал/набор интерфейсных модулей по вполне определённому протоколу (лучше бы — определяемому открытым стандартом, а не только внутрипроектным соглашением, хотя это и не всегда соблюдается), так что это взаимодействие можно отследить, и ошибка не распространится по системе. Более того, можно просто заменить модуль на альтернативный (это проще, если интерфейс стандартный) — и система продолжит работать. Даже оргзвено бухгалтерии можно заменить: «своё» подразделение заменить внешней бухгалтерской службой! Если же какие-то связи (в случае модулей говорят о «зависимостях») будут проходить не через интерфейсы модулей, а система будет эти связи использовать для реализации своей функции, то замена модуля приведёт к исчезновению этих связей и нарушениям в работе системы.

Есть множество методов, которые помогают уменьшать связность модулей в создаваемой системе. Одним из наиболее известных таких методов является DSM (design/dependency structure matrix)[452].

Оптимальная (минимальная из возможных для нормальной работы системы) связность модулей нужна и в «железных» системах, и в программных системах, и в организационных, и даже в танцах. Хорошей модульностью занимается архитектура::метод, роль агента, который выполняет архитектурную работу — архитектор (в случае организационных систем — орг-архитектор).

Хорошая модульность и минимизация связей между ними совсем не означает, что части системы/подсистемы не находятся в тесном взаимодействии друг с другом. Это означает, что эти **взаимодействия частей конструкции ставятся под жёсткий архитектурный контроль**, и подсистемы можно улучшать, а в случае стандартизации интерфейса даже заменять на принципиально другие по конструкции и принципу действия, не рискуя при этом ухудшить работу всей системы в целом.

Системный подход обычно называют холистическим, ибо он обращает внимание на систему в целом. Но нет других подходов, которые так бы помогали разбираться с разбиением на самые разные части, как системный подход. Суть системного подхода не только ко вниманию к целой системе, но и одновременному вниманию к частям системы. Системный подход — это про многоуровневость: осознанное удержание внимания всегда минимум на трёх системных уровнях: надсистемы (куда встраиваем нашу систему), целевой системы (что делаем), подсистем (из чего собираем). Это верно и для конструктивного/продуктового/модульного разбиения.

В руководстве по системной инженерии будет подробно рассказано об архитектурной работе, её месте в ряду других методов инженерной работы, даны ссылки на литературу по методам оптимизации межмодульных зависимостей, метрикам количественной оценки таких зависимостей.

Борьба со сложностью в мышлении

Edsger Dijkstra в 1974 году ввёл[453] понятие **разделения предметов интереса**

---

[452] http://www.dsmweb.org/

[453] http://www.cs.utexas.edu/users/EWD/transcriptions/EWD04xx/EWD447.html

(separation of concerns, разделение важных характеристик/предметов интереса/озабоченностей) как способ упорядочивания человеческих мыслей о каком-то предмете. Этот принцип гласит, что нужно обсуждать сложные ситуации по одной важной характеристике (concern) за раз, удерживая внимание самых разных ролей на каком-то одном аспекте, одной ипостаси этой ситуации. Это не значит, что игнорируются все остальные. Просто это удерживание во внимании и одного прямо сейчас обсуждаемого аспекта проблемы, и одновременно удержание внимания на объемлющем обсуждении всей проблемы. По словам Edsger Dijkstra про такое мышление, «It is being one- and multiple-track minded simultaneously».

И это разделение мышления по предметам интереса проводится рекурсивно, в понимании системного мышления — это одинаковыми методами мышления на каждом из многочисленных системных уровней. У каждой системы на каждом системном уровне есть много предметов интереса/важных характеристик, и они там обсуждаются по очереди. Ввиду эмерджентности на каждом системном уровне системы, подсистемы и надсистемы проявляют разные наборы свойств, имеют разные характеристики, и нужно обсуждать и эти разные системы, и их разные характеристики. Предметов интереса много для каждой системы на каждом системном уровне, плюс ещё есть предметы интереса к создателям, там уже отношение не часть-целое, а отношение создания — но это ведь тоже системный подход, начиная со второго его поколения.

Сложность описания/обсуждения системы тем самым уменьшается за счёт разделения одного запутанного сверхсложного обсуждения на отдельные не очень запутанные и не очень сложные обсуждения по двум направлениям:

- деление полного обсуждения всей системы на **обсуждение её отдельных частей по системным уровням.** Каждая часть системы проще, чем система в целом, поэтому обсуждение проходит более-менее единообразно на каждом уровне, а каждая часть системы описывается **концепцией её использования** (ConOps или OpsCon с точностью до множества нюансов их сходства и различий, которые отличаются в каждой инженерной школе мысли), то есть описанием системы как «чёрного ящика». В ней игнорируется внутренняя структура системы и её частей, поэтому обсуждение начинается всегда с целых систем в составе надсистемы, или целых подсистем в составе системы, и так «до самого низа» и «до самого верха». Само системное мышление как бы стоит «сбоку», оно трансдисциплинарно, удерживает внимание на всём системном разбиении, чтобы ничего не забыть. А уж системы каждого отдельного системного уровня в их разбиении могут обсуждать или прикладные специалисты одной команды, или даже разных команд.
- деление полного обсуждения каждой системы на одном системном уровне на **обсуждение с помощью множественных описаний разных аспектов** этой системы как прозрачного ящика. Важнейшие из этих описаний — это **концепция системы** (systems concept), включающая и функциональное, и конструктивное, и пространственное описание, и стоимостное, равно как и множество других. Тут разделение идёт по основным инженерным ролям (визионер, разработчик, архитектор, инженер внутренней платформы разработки), которые тоже специализируются на отдельных видах систем какого-то одного системного уровня.
- Отдельно **обсуждаем создателей в их графе**, то есть проходим уже не по

отношению часть-целое, а по отношению создания. Системное мышление заставляет удерживать во внимании и создателей!

Детальное и в подробностях обсуждение огромных сложных систем и графов их создателей принципиально (в силу самой сути системного подхода) может быть разбито на достаточно маленькие части, и ни одна часть этого обсуждения не будет забыта, ни одно описание не будет пропущено. Как съесть слона? По кусочку за раз!

Системное мышление по своей природе коллективно, оно позволяет разделить и умственный (по проектированию), и физический (по изготовлению и эксплуатации) труд, задействовать множество разных живых и не очень живых исполнителей для самых разных проектных ролей — и при этом детально продумываются части системы на всех системных уровнях для учёта всех известных ролевых интересов на каждом системном уровне.

Но как договориться о том, как склеивать все эти самые разные описания самых разных частей системы в одно целое системное описание? Как договориться всем этим ролям и их исполнителям? Все эти описания можно совместить, если понимать, что они описывают одно и то же место в пространстве-времени, относятся к одному и тому же воплощению системы. В центре любых системных обсуждений будет физическое воплощение системы, рассматриваемое в момент эксплуатации (run-time), к нему будут привязаны все остальные рассуждения.

Без системного подхода сложные проекты с задействованием большого количества разных специалистов выполнить в срок и вовремя невозможно: система будет неуспешна, ибо чьи-то интересы не будут соблюдены (то есть забыты, потеряны: на них не хватит внимания команды!), и в системе тем самым будет ошибка, люди не будут довольны её работой, или не будут довольны работой проекта, который создаёт систему (не будут довольны системами создания). Банально: если вы забыли обсудить какую-то систему, подсистему, надсистему «нашей» системы, или целевой системы, или даже создателя — быть беде, «если что-то может пойти не так, оно пойдёт не так». Неизбежны коллизии, что-то придётся переделывать/re-work, тратить незапланированные ресурсы.

Системы обсуждаются по одной части за раз, одному описанию части за раз — ни на секунду не теряя из вида целой системы в её окружении, а также её создателей.

Создатели («системы создания», убираем слово для типа мета-мета-модели) — тоже системы, только нужно не забывать, что всё их существование связано с целевой системой и поэтому не нужно говорить, что они сами и есть целевые. Но

они могут вполне быть «нашей системой» для менеджеров как инженеров предприятия. Если менеджер вдруг говорит, что «моё предприятие — это и есть моя целевая система», то это нарушение коллективности в мышлении, менеджер ни с кем не сможет договориться. Он должен честно сказать, что целевая система — это то, что делает его предприятие (даже если предприятие предоставляет сервис), а само предприятие — это для него «его система» (а не общая для всех система!).

Если вы поставляете какой-нибудь продукт для использования в составе какого-то предприятия или оргзвена (или его части, если речь не идёт обо всём предприятии или оргзвене в целом), то оно будет надсистемой для вашего продукта. Часы, которые вы поставили автомобильному заводу, могут быть использованы заводчанами (а не «пользователями часов») либо в выпускаемом заводом автомобиле (и заводчане тогда не «пользователи часов!»), либо в заводском цеху (и тогда заводчане сначала выступают в роли разработчиков цеха как надсистемы для часов, а потом и в роли интересующихся временем, «пользователи часов»).

В случае инженерии предприятия все системные рассмотрения остаются, но меняется терминология и практики. Так, модулями предприятия называются **подразделения** (тут реже говорят про **оргзвенья**, но системное мышление обращает внимание и на коллегиальные органы типа рабочих групп, советов и комиссий), функциональными частями предприятия являются его **оргроли**, функции/методы работы оргролей часто называют **рабочими процессами**, а управление конфигурацией предприятия называется **учётом**. А ещё важно, что оргзвенья не **разрабатывают**, а **организуют**, при этом само изготовление — это не **производство**, а **лидерство** (людям, которые вменяемы, объясняют, как устроено их оргзвено, и какое оргместо они там должны занять, сотрудничая с другими людьми — это и есть лидерство). Подробней — в руководстве по системному менеджменту.

Если речь идёт об организационных системах, то само понятие **инженерии предприятия** (enterprise engineering) как-то стало употребимым пару десятков лет назад, но терминология там не стала «железно-инженерной», а также не стала общесистемной, уровня мета-мета-модели. Если речь идёт о каких-то других видах систем (медицина, правоохрана, танцы, политика и т.п.), то терминология может отличаться от терминологии инженерии и системного мышления ещё больше. В образовании (инженерии личности) речь идёт об инженерии мастерства, и там выделяются все те же роли: визионер — это культуртрегер, проектировщик — это методолог, инженер-технолог — это методист, и т.д. (подробней об этом — в руководстве по инженерии личности).

Слова не важны, важно общее устройство мышления (на что обращать внимание во всех этих проектах: системное мышление занимается управлением вниманием). **Слова важны**: без терминологии какой-то предметной области вы вообще не поймёте, о чём идёт речь. А ещё важны типы мета-мета-модели: с ними вы сможете увидеть общее в самых разных ситуациях, самых разных проектах, самых разных системах самых разных видов.

- **Вам потребуется в каждом проекте сопоставлять «яблоки из учебника» с «яблоками из жизни»: отождествлять понятия, соответствующие предметам самых разных предметных областей (domains, понятия из мета-модели) с фундаментальными понятиями системного мышления (мета-мета-модели). Помним, что это просто**

ориентирование на местности: вы быстро поймёте, где находитесь. Дальше надо будет:

- **Стратегировать, то есть разобраться, каким методом с какими предметами метода вы будете работать — даже если ни вам, ни всем остальным не ясно, что же надо делать, чтобы добиться успеха. Для этого предложить несколько наиболее вероятных для успеха всего дела методов, а затем при помощи методов принятия решений выбрать тот, который не удаётся раскритиковать.**
- **На основе выбранного метода спланировать ваши действия, привязать к конкретному времени и ресурсам.**
- **Принять планы всерьёз: выделить (или даже добыть, если поначалу их нет) ресурсы и выполнить эти планы, даже если речь идёт о пошаговом планировании.**
- **И дальше продолжать системно мыслить, то есть продумывать то важное, на что указывает системное мышление, надёжно удерживать это важное в коллективном внимании участников проекта, не давать вниманию рассеиваться.**

Задания: системное моделирование

Поставьте отметку о выполнении:

1. Написан пост по итогам прохождения материала и занятий моделированием в текущем разделе руководства. В посте приведены форматы (заголовки всей таблицы, заголовки колонок, содержание таблички необязательно) трёх табличек, которые вы разработали для вашего рабочего проекта сами (не надо копировать в пост таблички, которые вы взяли без изменений в наставлениях нашего руководства).

12. Вместо глоссария: онтология системного подхода третьего поколения

Онтология системы как набор описаний для понятийного наведения внимания

Часто просят дать **глоссарий** (краткие словарные определения основных понятий) нашего руководства, игнорируя замечание начальных разделов о том, что **определения — это гробик для уснувшей мысли**. Так что мы закончим наше руководство не глоссарием, а коротким текстом, излагающим онтологию системного подхода (мета-мета-модель), в котором напомним про понятия третьего поколения системного подхода и дадим литературу по теоретическим основаниям системного подхода. Обо всём этом в руководстве говорилось на многих страницах, но для продвинутого понимания мы крайне советуем посмотреть упоминаемую литературу.

Если вам нужно не продвинутое (исследовательское) понимание, а «обычное студенческое», то можете опустить изучение материала этого раздела и переходить к изучению руководств по методологии, системной инженерии, инженерии личности, системному менеджменту (в таком порядке). Если нужно продвинуться по «аспирантскому пути», то попробуйте пойти не по прикладным руководствам инженерного цикла программы рабочего развития, а по руководству по интеллект-стеку фундаментальных методов мышления программы исследовательского

развития.

По необходимости в этом разделе (впрочем, и в предыдущих разделах руководства по системному мышлению) затрагивается и материалы руководства по методологии. Когда речь идёт о **методах** и **работах**, которыми заняты **создатели**, то приходится обращаться к понятиям **создания и развития**, рассматривать **методы** создания и **предметы методов**. Частично затрагивается и онтология системной инженерии. **Системная инженерия** как **метод/способ/культура/практика изменения мира к лучшему** базируется **на онтологии системного подхода** (иногда говорят «системная онтология»/systems ontology, включая в неё все понятия системного подхода).

Есть несколько современных инициатив по **онтологическим посылкам**/commitments такой онтологии, которые мы считаем **третьим поколением системного подхода**, учитывающим **термодинамически обусловленную эволюцию**. Мы предлагаем пути гармонизации этих подходов к созданию системной онтологии.

Gruber, Borst и Studer определили **онтологию** как **явную спецификацию разделяемой концептуализации** (an ontology is a formal, explicit specification of a shared conceptualization)[454]. Определим **концептуализацию** как задание важных для надёжного **active/embodied inference**[455] объектов восприятия в мире. Это позволяет сформулировать задачу создания онтологии в терминах **управления вниманием**, а «явная формальная спецификация» тут говорит о том, что эти объекты внимания выделяются не спонтанно, а по какой-то явной модели этих объектов (онтологии), заданной при помощи другой модели-формализма (foundational ontology). В качестве **foundational ontology** мы выбираем **объекты и отношения**.

Мы решаем вопрос создания онтологий и объяснений на их основе (появления и улучшения объяснительных теорий, использующих для объяснения альтернативную нарезку мира на объекты) в соответствии с **поппериа**нской **эпистемологией**[456]: **объекты** в онтологии появляются путём **догадок**, приемлемость этих догадок для суждений о мире подвергается сомнению, но догадки, пережившие **критику**, **принимаются всерьёз**. То есть хорошие догадки о том, что **мир состоит из систем** (догадки о том, какие есть понятия в онтологии системы) и суждения о том, что эти догадки полезны для изменения мира к лучшему, можно взять из литературы, и если нам не известна их критика и мы не можем сходу предложить **фальсификации** этих догадок рассуждением и/или экспериментом, то мы считаем их **нашими лучшими теориями** (SoTA, state-of-the-art) о мире, принимаем их всерьёз и далее учим **интеллектуальных агентов** (**реплицируем мемы** этих **теорий/знаний/объяснений/дисциплин/алгоритмов**), чтобы эти догадки об объектах были **общими**/shared.

Конечно, мы не просто сообщаем агентам (людям, AI-агентам, организациям) онтологию системы, но мы учим их выделять в мире **абстрактные** и **физические**

---

[454] Nicola Guarino, Daniel Oberle, and Steffen Staab, What Is an Ontology?, in S. Staab and R. Studer (eds.), Handbook on Ontologies, Springer-Verlag, 2009

[455] Karl Friston, Embodied inference: or "I think therefore I am, if I am what I think" In W. Tschacher, C. Bergomi (Eds.), The implications of embodiment: Cognition and communication (pp. 89–125). Imprint Academic., https://www.fil.ion.ucl.ac.uk/~karl/Embodied%20Inference.pdf

[456] Karl Popper, https://plato.stanford.edu/entries/popper/

объекты, описываемые явной и по мере понижения уровня абстрактности всё более формальной спецификацией. Верхние абстрактные уровни системной онтологии чаще всего представлены в **распределённых/нейросетевых представлениях**, поэтому изложение системного подхода идёт не в формальном (математическая логика) виде, а в тексте на естественном языке, понятия вводятся не определениями, а их **употреблением в разных контекстах**. Но после перехода к прикладным специализациям в различных **предметных областях/domains** представления становятся **локальными/знаковыми** и выражаются в формализмах, удобных для обработки классическими информационными технологиями (то есть для обработки в реляционных, графовых и иных базах данных).

Конечно, если появляются проблемы в использовании этой онтологии, это означает её фальсификацию, но это просто повод поправить онтологию, исправив выявившиеся ошибки — онтология системного подхода тем самым **развивается**/evolve, она на каждый момент времени представляет собой лучшее, что нам известно на этот момент. Мы[457] разработали программу рабочего развития инженеров-менеджеров, в которой дали набор руководств по методам ведения современных рабочих проектов. Эти руководства наставляют выявлять в мире объекты, описываемые/**типизируемые онтологией на уровне абстракции мета-мета-модели мира.** Основная последовательность руководств в части их освоения сейчас реализует подход сначала полностью осознанного, а затем интериоризированного «автоматического» **понятийного наведения внимания** инженеров-менеджеров на объекты, описываемые/типизируемые онтологией системы. Инженер-менеджер обучается тому, как:

- **наводить своё внимание** и внимание других **агентов** на объекты из явной спецификации (онтологии), то есть наводить внимание понятийно, а не спонтанно, а также **удерживать** понятийно наведённое внимание на самых разных временных масштабах, в том числе включая коллективное внимание в таких агентах, как команда или целое предприятие, путём лидерства, или влияния на сообщество или общество (руководство по рациональной работе),
- наводить внимание на **системы** (руководство по системному мышлению),
- наводить внимание на **деятельности/практики**, которые осуществляют **системы-создатели** (руководство по методологии),
- структурировать **деятельности и роли**, которые обязательно удерживаются во внимании в ходе **проектов системной инженерии**, (руководство по системной инженерии),
- обучать **мастерству мышления** агентов, задействуя в том числе и локальные, и распределённые представления (руководство по инженерии личности),
- обращать внимание себя и **организации** (команды, коллектива, предприятия) на объекты, описываемые **специализацией** системной онтологии для таких видов систем, как организации (руководство по системному менеджменту).

Для того, чтобы учить инженеров-менеджеров **SoTA** (state-of-the-art, «лучшее из известного на настоящий момент») системной онтологии, мы **гармонизировали** между собой несколько частных концептуализаций понятия системы, которые

[457] Мастерская инженеров-менеджеров (ранее ШСМ, школа системного менеджмента), https://system-school.ru/

вместе представляют собой **третье поколение системного подхода**. Как обычно, каждое поколение системного подхода включает в себя все достижения предыдущего поколения, но добавляет что-то новое.

Первое поколение системного подхода: система в её окружении во время operations/эксплуатации

Первое поколение системного подхода появилось в 40е годы прошлого века главным образом в результате работ von Bertalanffy[458]. Понятие системы как предмета рассмотрения, отделённого от его окружения, появилось в физике очень давно, но системный подход как рассмотрение всего мира в виде взаимодействующих систем появился главным образом после работ von Bertalanffy по **общей теории систем**. **Подход** — это обычный термин для ситуации, когда онтологию, успешно наработанную и опробованную в одной **предметной области**/domain, начинают использовать в самых разных предметных областях. Ключевым тут было осознание, что системы представляют собой взаимодействующие **холоны** (Koestler ввёл этот термин для того, чтобы описать **часть целого**, которая сама **состоит из** частей[459]), плюс эти системы как-то появляются в мире и потом исчезают из мира, то есть проходят обычный для биологии **жизненный цикл.** Von Bertalanffy был биологом и обобщал в том числе успехи применения системного подхода в биологии, то есть жизненный цикл — это цикл рождения-взросления-размножения и отдельно — смерти (которая после размножения уже не влияет на геном, поэтому может не учитываться в эволюции, но обязательно учитывается в техно-эволюции). В результате взаимодействия **частей системы** во **время работы**/operations получаются **эмерджентные свойства** целого. Шестерёнки в часах ещё не показывают время, часы показывают время, дом с часами внутри уже не показывает время. Ключевым тут было то, что первое поколение системного мышления оперировало по меньшей мере двумя разными способами разбиения на части, зависящими от времени рассмотрения:

- **функциональное разбиение** в зависимости от назначения системы в надсистеме и назначения подсистем в системе. Оно предполагало рассмотрение системы в ходе её operations.
- **конструктивное/модульное разбиение**, нужное для рассмотрения **времени создания**/construction системы.

Трудности в освоении **системного мышления** были главным образом в том, что людям трудно было воспринять концепцию выделения динамических объектов в мире своим вниманием — прямо в ходе работы/operations системы. Чаще всего при упоминании частей-целых они в голове представляли себе «**взрыв-диаграмму**»[460], заодно теряя и **многоуровневость** такого **разбиения**. Понятно, что нельзя обсуждать эмерджентные свойства остановленной и разбитой на одноуровневые физические конструктивные части системы, ибо взаимодействие — это время operations и рассмотрение должно быть функциональным. Но и разбиение на конструктивные части важно, ибо такую систему нужно **создать**. Например, ножницы функционально состоят из режущего блока и ручки, а конструктивно из двух половинок ножниц и скрепляющего их винта. При этом пользователя **интересует** функциональное рассмотрение, а заводского инженера —

[458] https://en.wikipedia.org/wiki/Ludwig_von_Bertalanffy
[459] Koestler, Arthur (1967). The Ghost in the Machine
[460] Bellami, Nikon F3-P Parts Diagram, https://www.japancamerahunter.com/2014/11/nikon-f3-p-parts-diagram/

конструктивное.

Берталанфи включил в состав предлагаемых им системных дисциплин в том числе и **системную инженерию**, которая быстро развивалась. В системной инженерии целевые системы были физичны: это давало гарантированное **заземление**/ grounding[461] для всех описаний. При этом сами **описания как система не рассматривались**, **взаимодействие** (то есть изменения во времени под действием друг друга) частей описаний, которое давало бы **эмерджентность**, обсуждать **строго** не удавалось, **формализма** построить не удавалось — **мереотопологические** рассмотрения **ментальных/абстрактных/математических объектов** по большей части заканчивались неудачей.

Второе поколение системного подхода: целевая система в её окружении, которую создают системы-создатели в их графе

Второе поколение системного подхода опиралось главным образом на идеи системной инженерии: физические (в том числе **киберфизические**, в том числе киберфизические с людьми) системы создаются людьми, а не размножаются сами в цикле, и эти системы главным образом неживые. Это **проблематизировало** понятие жизненного цикла, который оказался не жизненный и не цикл, хотя само **понятие** осталось. Во втором поколении тем самым появились системы **создания** (enabling/constructor systems), которые вводят кроме **отношения часть-целое** для систем ещё и **отношение создания** для **создателей/constructors**, создающих и развивающих **целевую систему** (наиболее известны тут работы Peter Checkland[462] конца 70х и начала 80х). В системной инженерии появился **системноинженерный менеджмент**, предметом которого стало **создание и развитие** систем создания (появились **графы создателей** целевых систем).

Диаграммы системной инженерии подчёркивали **«водопадную» модель жизненного цикла**, в котором система **замышлялась, проектировалась, воплощалась, испытывалась, эксплуатировалась и выводилась из эксплуатации**, на чём проект обычно заканчивался. Все эти **работы** самыми разными **методами** живая система выполняла для себя главным образом сама (при этом часть работ выполнялась за систему **эволюцией**, но это обычно не рассматривалось), а вот в **инженерии** всё это выполняли создатели. Для разных **ролей** (roles, stakeholders) в системе создания проявлялись разные **интересы**/concerns по отношению к системе и её проекту, что требовало различных **аспектных описаний**/views, которые делались исходя из каких-то **методов описаний**/veiwpoints, хорошо подходящих для обсуждения самых разных интересов самых разных **проектных ролей**.

Стандарты ISO 42010[463] и ISO 15288[464] закрепили концептуализацию системы для системной инженерии, это гарантировало то, что системноинженерная онтология второго поколения системного подхода разделяемая/shared в инженерных сообществах. Формализация системной онтологии на основе **4D**

---

[461] https://plato.stanford.edu/entries/grounding/

[462] https://en.wikipedia.org/wiki/Peter_Checkland

[463] ISO/IEC/IEEE 42010, Systems and software engineering — Architecture description, https://www.iso.org/standard/74393.html

[464] ISO/IEC/IEEE 15288, Systems and software engineering — System life cycle processes, https://www.iso.org/standard/63711.html

экстенсионализма/4D extensionalism (идеи, что если два объекта занимают одно и то же **место** в **пространстве-времени**, то это один и тот же объект — и это **физический объект**) была предложена в ISO 15926-2[465] в 2003, она опиралась на идеи BORO[466], и на базе этих идей было предложено ещё несколько других похожих онтологий, главным образом для военных применений, эту работу вела IDEAS Group[467]. Особо можно выделить работу Matthew West, который предложил в 2010 основанную на тех же идеях 4D экстенсионализма онтологию HQDM[468], в которой понятие системы было сделано одним из центральных в онтологии. Все эти подходы предполагали наличие не просто спецификации концептуализации, но и формальное выражения концептуализации на **логических языках описания онтологий** (таких как очень редко используемый EXPRESS[469], но также и появившийся позже более популярный OWL[470]) для создания **моделей данных** в базах данных PLM систем[471]. Основная онтологическая посылка была в использовании 4D онтологии, что давало хорошее и компактное описание **изменений**, которые претерпевает система в ходе её создания из частей[472].

Пик этой линии работ по инженерному пониманию системы в рамках второго поколения системного подхода пришёлся на 2008-2013 годы, после чего интерес к подобному **формально-логическому онтологическому описанию** системы немного угас: создание онтологий как явная формальная спецификация разделяемых концептуализаций больше не воспринималось как продвижение по пути создания AI, и не помог даже ребрендинг формальных онтологий как **графов знаний**[473]. Web 3.0 как **semantic web** ожидался как будущий интернет-мейнстрим, но **семантические методы,** поддержанные **инструментами** на базе OWL так и остались нишевыми. В конечном итоге в **больших языковых моделях** (LLM) удалось задействовать распределённые представления для генерации объяснений, а локальные (знаковые) представления оказались удобными главным образом для критики идей, а не для тотально всего мышления. На сегодняшний день семантика включает как локальные, так и распределённые представления, хотя задача их объединения одной теорией (нейро-символические вычисления) пока не решена.

Continuous everything в инженерии как техно-эволюция

Ограничением **водопадной модели**[474] была идея однократного однонаправленного прохождения создателями жизненного цикла как набора работ,

---

[465] ISO 15926-2, Idustrial automation systems and integration — Integration of life-cycle data for process plants including oil and gas production facilities — Part 2: Data model, https://www.iso.org/standard/29557.html
[466] Peter El Hajj, Business Objects Reference Ontology, https://digitaltwinhub.co.uk/top-level-ontologies/business-objects-reference-ontology-r2/
[467] IDEAS Group, https://ideasgroup.org/
[468] https://www.elsevier.com/books/developing-high-quality-data-models/west/978-0-12-375106-5
[469] ISO 10303-11:2004 Industrial automation systems and integration — Product data representation and exchange — Part 11: Description methods: The EXPRESS language reference manual, https://www.iso.org/standard/38047.html
[470] Web Ontology Language (OWL), https://www.w3.org/OWL/
[471] What is PLM (Product Lifecycle Management), https://www.oracle.com/uk/scm/product-lifecycle-management/what-is-plm/
[472] Ian Bailey, The simplification in integration architecture that 4D supports, https://gateway.newton.ac.uk/sites/default/files/asset/doc/2105/Ian%20Bailey.pdf
[473] John Sowa, Knowledge Graphs for Language, Logic, Data, Reasoning, https://www.youtube.com/watch?v=J9_kXpZAcQY (Part of the Ontology Summit 2020)
[474] Waterfall model, https://en.wikipedia.org/wiki/Waterfall_model

проводимых по практикам, причём целевая система рассматривалась как пассивно претерпевающая это разовое создание. Поэтому в инженерии развивались **agile-подходы**, преодолевающие понятие «водопадного» жизненного цикла, в 2001 году появился Manifesto for Agile Software Development[475]. В 2017 году идея **evolvability** как основного **архитектурного свойства**, отражающего идею **«непрерывного всего»/continuous everything** закрепилась не только в передовой инженерной практике, но и отразилась в литературе[476], а **архитектура** окончательно выделилась из **разработки** в отдельную предметную область, ибо литература стала отражать неизбежные «продуктивные конфликты» архитекторов с разработчиками. Архитектура работала над нарезкой системы на минимально взаимодействующие **конструктивы/модули** для поддержания **стабильности** общего целого в **эволюционном масштабе времени** (хорошо известные инженерам **-ости/-ilities** оказались архитектурными характеристиками), а разработчики заботились о максимизации функциональности, что могло влечь временное ухудшение архитектурных характеристик как накопление **технического долга** (работ, направленных на поддержание выбранных архитектурных решений, но не затрагивающих напрямую функциональность), который оказался **архитектурным долгом**.

В инженерию пришли идеи **техно-эволюции**, но они отличались от идеи **биологической эволюции** тем, что **мемом** как аналог **биологического генома** не находился в самой изготавливаемой системе, а хранился в **цифровой** (то есть подразумевающей точное повторение при **репликации** без накапливания аналоговой ошибки) форме где-то в системах-создателях. Это позволяло существенно ускорить эволюцию, ибо для какого-нибудь технокраба не надо было ждать, пока умрёт целый краб, чтобы путём **мутаций** формирующих феном клешни генов заменить неудачный вариант клешни только вместе с крабом. Также можно было использовать **умные/smart мутации**, получая результаты, то есть **в реальной жизни** пробовать только варианты клешни, показавшие успешность при моделировании в **виртуальном мире** — и менять только клешню, но не всего краба. Это давало высокую скорость техно-эволюции и её результаты, недостижимые только разовым предложением набора новых **фич** (разовое прохождение «водопада» для одного экземпляра системы) или только эволюцией со случайными мутациями[477]. Для понимания, насколько удачна система в эксплуатации, появилось моделирование системы для времени операций — **цифровой двойник**, отражающий не только текущее состояние мемома, но и состояние **фенома** системы. Так что онтологии системы, ориентированной на два времени (1. эксплуатации и 2. создания одного **инкремента/фичи**) оказалось недостаточно. Недостаточно было и средств моделирования целой **группы систем** (**product lines**[478], иногда **system families**) как сосуществующих в один момент времени разных вариантов системы.

---

[475] Manifesto for Agile Software Development, https://agilemanifesto.org/

[476] Neal Ford, Rebecca Parsons, Patrick Kua, Building Evolutionary Architectures, 2017, https://www.oreilly.com/library/view/building-evolutionary-architectures/9781491986356/

[477] Joel Lehman, Jonathan Gordon, Shawn Jain, Kamal Ndousse, Cathy Yeh, Kenneth O. Stanley, Evolution through Large Models, 2022, https://arxiv.org/abs/2206.08896

[478] Software Product Lines matures into the next generation of Systems and Software Product Line Engineering, https://www.softwareproductlines.com/

Выход оказался в идее **непрерывного всего/continuous everything**[479] как бесконечно длящейся разработки с **непрерывным вводом в эксплуатацию**/continuous delivery всё новых и новых **версий** целевой системы со всё новыми и новыми изменениями, причём незамедление темпов разработки при растущей сложности целевой системы поддерживается как изменяющейся по ходу разработки (evolving) архитектурой, так и изменяющейся структурой **команд**/teams, которые осуществляют эту разработку (использование **обратного манёвра Конвея**[480]). В software engineering эта идея развивалась в рамках подхода DevOps/SRE/platform engineering (инженерия внутренней **производственной платформы**)[481], в «железной» инженерии эта идея обсуждается сейчас на традиционных для системной инженерии примерах аэрокосмоса, когда **испытания** двигателей следующей версии начинаются раньше, чем испытанные двигатели предыдущей версии в составе ракеты улетают в свой первый полёт, а каждый новый экземпляр ракеты имеет новые фичи и/или улучшенную конструкцию по сравнению с предыдущим экземпляром. Но уже подобным способом работают не только в аэрокосмосе, но и автомобилестроители, а также робототехники. «Непрерывное всё» означает переход инженерии к техно-эволюционным проектам, то есть умным (неслучайным) мутационным/эволюционным изменениям мемома (**информационной модели** проекта/design системы), а не только изменениям фенома как «доработке по месту напильником» экземпляра системы. В ходе эволюции система рассматривается уже не как экземпляр (возможно, промышленная серия с одинаковыми экземплярами), а как вид — где конфигурация каждого экземпляра может быть уникальной, непрерывно обсуждаются **умные мутации/smart mutations**[482]**,** с наибольшей вероятностью ведущие к успешности системы (в дарвиновской эволюции — это случайные мутации при репликации) и даже в отличие от естественного отбора ведётся промышленное A|B тестирование, (и даже больше, ибо тестируются не только два варианта, а во множестве подсистем множество вариантов сразу) как направленная селекция таких мутаций. **Развитие/evolving системы** стараются делать как можно более приближенным к бесконечному, максимально долго приспосабливая систему к неизбежным изменениям как в окружении, так и в ней самой, так и в системах создания. В биологии и науках о жизни аналогичные проблемы приближения к бесконечному по времени развитию обсуждают как **бесконечное развитие/open-endedness**[483].

Тем самым в онтологии системы появилось отражение идей эволюции и её начали развивать в рамках **третьего поколения системного подхода**, который не только учитывает системы создания из агентов в их разных проектных/деятельностных ролях, но и учитывает в явном виде с тремя временами: оперирования системы как фенома, создания инкремента фенома из мемома (**проектирование/design** и **изготовление/implementation**), изменения мемома (эволюции, continuous everything).

---

[479] The Continuous Everything (CE) concept defined, https://en.itpedia.nl/2021/06/02/het-continuous-everything-ce-concept-gedefinieerd/

[480] Nicole Forsgren, Jez Humble, Gene Kim, Accelerate, 2018, https://www.oreilly.com/library/view/accelerate/9781457191435/

[481] Aeris Stewart, How Is Platform Engineering Different from DevOps and SRE?, 2022 https://thenewstack.io/how-is-platform-engineering-different-from-devops-and-sre/

[482] https://arxiv.org/abs/2206.08896

[483] Kenneth O. Stanley, Joel Lehman and Lisa Soros, Open-endedness: The last grand challenge you've never heard of, 2017, https://www.oreilly.com/radar/open-endedness-the-last-grand-challenge-youve-never-heard-of/

Безмасштабные описания физических систем

В обсуждении эволюции в биологии системный подход проявлялся как **большие эволюционные переходы**/major evolutionary transitions — переходы **роста сложности** между большими молекулами и клетками, клетками и многоклеточными организмами. Примерно такое же усложнение систем идёт и в техно-эволюции: из транзисторов делают микросхемы, из микросхем компьютеры, компьютеры соединяют в датацентры, датацентры в компьютерную сеть. Аналогия вроде бы понятна, но требовалось получить **безмасштабную/scaleless/scale-free теорию** (описание, применимое для систем любого **масштаба**/size/scale, то есть для мира микрочастиц с учётом явлений квантовой физики, но также и для макрообъектов). Безмасштабные описания применимы для систем самого разного масштаба, в том числе живых и неживых, а также живых сознательных и коллективов живых сознательных существ (например, человечество в целом). Безмасштабность в том числе означала и **безмасштабность в четвёртом измерении**, времени: учёт времени создания системы, времени жизни/эксплуатации/использования, но ещё и **времени эволюции**.

Ключевым стал **переход к формулировкам физических явлений как информационных**, а также объяснение **феномена стабильности объектов в физическом мире** — почему какие-то объекты (например, молекула или человек) поддерживают свою **форму в пространстве-времени**. Fields, Glazebrook, Levin, Friston предложили онтологический фреймворк **панпсихизма** в форме **минимального физикализма** для описания физически стабильных систем как реализующих **принцип минимизации свободной энергии** и охватывающий всю **шкалу сложности/масштабов** от элементарных частиц до людей и обществ[484]. **Свободная энергия** определяется как информационная характеристика системы, а не традиционная энергия механической работы или работы электромагнитных сил. Системы эти усложняются от элементарных частиц через молекулы, через системы из косной материи до живых созданий, где они усложняются от одноклеточных до многоклеточных организмов и их популяций. Все эти виды систем уменьшают **байесовский** (или **расширенный байесовский/excess bayesian**, чтобы учесть **квантовоподобность**/quantum-like вычислений в биологии[485]) **сюрприз** несоответствия определяемых **порождающей/generative моделью[486]** ожиданий измерений реальным измерениям в реальности.

Эта безмасштабная онтология физически стабильных систем формализована с использованием **теории категорий** как **foundational ontology** в части **мереологии[487]** и высказана в форме, позволяющей описать **квантовоподобный**

---

[484] Fields, C., Glazebrook, J. F. and Levin, M., Minimal physicalism as a scale-free substrate for cognition and consciousness. Neuroscience of Consciousness 2021, https://chrisfieldsresearch.com/min-phys-NC-2021.pdf

[485] Yukio-Pegio Gunji, Shuji Shinohara and Vasileios Basios, Connecting the free energy principle with quantum cognition, https://www.frontiersin.org/articles/10.3389/fnbot.2022.910161/full

[486] https://en.wikipedia.org/wiki/Generative_model

[487] Fields, C. and Glazebrook, J. F. A mosaic of Chu spaces and Channel Theory I: Category-theoretic concepts and tools. Journal of Experimental and Theoretical Artificial Intelligence 31: 177-213, 2019, https://chrisfieldsresearch.com/mosaic1-pre.pdf и Fields, C. and Glazebrook, J. F. A mosaic of Chu spaces and Channel Theory II: Applications to object identification and mereological complexity. Journal of Experimental and Theoretical Artificial Intelligence 31: 237-265, 2019, https://chrisfieldsresearch.com/mosaic2-pre.pdf

active/embodied inference[488]. Эта линия **онтологической инженерии** хорошо показывает то, как думать о функционировании систем самой разной эволюционной сложности, в том числе как применить **стохастические** рассмотрения к **неэргодическим** системам, то есть системам с памятью. Одним из самых сильных подтверждений рассуждений по этой линии было создание **hybrot**[489], который научился играть в Pong: достаточно сложная система, способная к научению (например, **естественная нейронная сеть**) должна демонстрировать в силу принципа минимизации свободной энергии поведение, минимизирующее **непредсказуемость внешней среды**, что было продемонстрировано в эксперименте[490]. По факту в этой линии работ была сделана формализация и математизация основанных на физике идей онтологии первого поколения системного подхода, хотя в силу безмасштабности теории можно так же рассматривать и другие масштабы времени: жизненные циклы систем, и их эволюцию. Ключевым в этом было разработанное в квантовой физике для решения проблемы наблюдателя **понятие измерения как взаимодействия систем**, а не просто **пассивного восприятия/наблюдения**. Обратное тоже верно: взаимодействие систем, приводящее к изменению, то есть создание, описывается как обратное измерению. Это означает, что любое взаимодействие даже молекулы как системы с её окружением можно рассматривать как измерение или создание: молекула является прото-агентом, который как-то воспринимает/познаёт/измеряет и/или создаёт/изменяет окружающий мир, как-то поддерживая в нём свою стабильность в силу соблюдения принципа минимизации свободной энергии.

Эта линия рассуждений (восприятие/измерение обратно изменению/созданию и оба являются взаимодействием систем) была проведена в **теории создателей/constructor theory**, предложенной Deutsch и развиваемой разными исследователями **квантовой гравитации**, ибо там крайне нужны именно безмасштабные описания[491]. **Создатель/constructor** — это какое-то физическое устройство, которое может долго поддерживать свою собственную неизменность/стабильность, изменяя при этом окружение какими-то заранее описанными последовательностями **операций** (например, это молекула катализатора, или робот с универсальным компьютером, или живое существо — то есть достаточно продвинутые создатели могут в том числе и **реплицировать** себя, участвовать в эволюции). Безмасштабные физические теории, опирающиеся на представления о связанных с информацией **изменениях как вычислениях**, оказались весьма продуктивны. Концепция создателя стала расширением хорошо разработанного понятия **универсального вычисления** и **универсального вычислителя/компьютера** (все универсальные компьютеры аналогичны машине Тьюринга: при достаточном времени и памяти они все могут вычислить всё, что принципиально вычислимо, то есть не запрещено принципами математики) на понятие **универсального преобразования/transformation**: **универсальный создатель/constructor** может при наличии достаточного времени и материалов

---

[488] Fields, C., Friston, K., Glazebrook, J. F. and Levin, M., A free energy principle for generic quantum systems. Progress in Biophysics and Molecular Biology 173: 36-59, 2022, https://chrisfieldsresearch.com/qFEP-2112.15242.pdf

[489] Hybrot, https://en.wikipedia.org/wiki/Hybrot

[490] Brett J. Kagan, Andy C. Kitchen, Nhi T. Tran, Forough Habibollahi, Moein Khajehnejad, Bradyn J. Parker, Anjali Bhat, Ben Rollo, Adeel Razi, Karl J. Friston, In vitro neurons learn and exhibit sentience when embodied in a simulated game-world, 2022, https://www.cell.com/neuron/fulltext/S0896-6273(22)00806-6

[491] https://www.constructortheory.org/

выполнить все возможные преобразования физического мира, которые не запрещены принципами физики (возможно, для этого универсальному создателю потребуется создать многочисленные копии/реплики себя, а также создать самый разный **инструментарий/tools**).

Конструктивизм: системы выполняют операции как с физическими объектами, так и с абстрактными объектами

Следующее, что должно быть учтено в онтологии системы, опирающейся на физику — это онтология самой физики и математики. Такая онтология предлагается во множестве работ, но выделим работу Deutsch[492], в которой предложено считать **физику наукой о реальных объектах**, **математику — наукой о ментальных/абстрактных/математических объектах, а компьютерную науку — экспериментальной наукой о доказательствах того, что поведение физических объектов как-то может отражать поведение идеальных/абстрактных/математических объектов** (то есть наука об **универсальных компьютерах** как физических устройствах, способных выполнять **вычисления/рассуждения/inference** — традиционных электронных, квантовых и т.д., включая вычисления/рассуждения человека или даже человеческих коллективов вместе с их компьютерами). Эти идеи предполагают сдвиг обсуждения с **моделей** и **данных** на **универсальные вычислители/computers** как физические устройства, интерпретирующие такие данные и меняющие в зависимости от этих вычислений состояние **окружающей среды** (ввод исходных данных для вычислений и вывод результатов в символьной форме тут просто частный случай. Восприятие окружающего мира и изменение окружающего мира и/или «себя» как embodied computer/constructor — это просто более общий случай). Центральным в этом **подходе** будет понятие **создателя/constructor** — физического устройства, которое может взаимодействовать с окружающей средой, производя в ней многократно какие-то операции согласно описаниям, и при этом удерживая свою стабильность (например, молекула катализатора, или робот, или человек). Создатели дают подход к формальному описанию второго поколения системного подхода: какие-то системы разной **степени агентности/разумности** (из косного вещества, роботы, разумные агенты и всевозможные их гибриды) создают по каким-то описаниям целевые системы, а дальше уже можно обсуждать всё как в системном подходе первого поколения, то есть как целевые системы функционируют в их окружении.

Примерно такой же подход, интегрирующий физику, топологию, логику и компьютерную науку с **переносом внимания на операции с объектами** вместо рассмотрения объектов в их статических отношениях, предложили Baez и Stay в подходе Rosetta stone[493]. После перехода от описания взаимодействия систем как **процессов в сетях/networks** (электрические, гидравлические сети, а также **сети взаимодействий в системной динамике**, обычно связываемых с функциональными представлениями системы, то есть представлениями времени функционирования, первое поколение системного подхода) Baez делает предложение о переходе к использованию формализма **теории симметричных**

---

[492] David Deutsch, The Beginning of Infinity, 2011, https://www.thebeginningofinfinity.com/

[493] John Baez and Mike Stay, Physics, topology, logic and computation: a Rosetta Stone (2011), in "New Structures for Physics", ed. Bob Coecke, Lecture Notes in Physics vol. 813, Springer, Berlin, 2011, pp. 95—174, https://arxiv.org/abs/0903.0340

моноидальных категорий для описания не просто процессов, но процессов **открытых систем**, подразумевающих взаимодействие с окружающей средой[494].

Снова и снова мы видим ход на использование теории категорий (а не объектов и отношений, базирующихся на идеях классической теории множеств) как foundational ontology для онтологии системы, и основной ход — это переход от выражения онтологии в **статичных отношениях** к **морфизмам, изменениям**. Этот мыслительный ход соответствует ходу на **конструктивизм в математике**, когда от «вечных классов»/eternal classes и их отношений мы переходим к **операциям построения/construction[495]**. Это даёт возможность **конструктивистского переформулирования мереологии** как центральной онтологической дисциплины системного подхода, она как раз изучает отношения «часть-целое»[496], а поскольку тут говорится об операциях по созданию целого из частей, то конструктивизм даёт ход и на формальное выражение идей создателей (то есть позволяет описать идеи не только первого поколения системного подхода, но и идеи второго поколения).

Этот ход на конструктивистскую мереологию был предложен Fine, его мереология включает уже не только физические, но и абстрактные объекты[497]. Она предлагает **операции конструирования целого из физических и абстрактных частей** (например, конструирование множества из его элементов). Хотя Fine в явном виде этого не говорит, определяя операции конструирования (равно как и морфизмы теории категорий) выполняющимися как бы «никем», но за этими операциями легко увидеть физическое устройство-constructor из constructor theory, в том числе для абстрактных частей — физическое устройство, воплощающее «универсальный вычислитель», в том числе и квантовый компьютер, и живой математик — вычислительно они эквивалентны и физичны, чтобы как-то получить информацию о входных и выходных данных вычисления. Подход Fine позволяет ставить вопрос о понятии системы, простирающемся за пределы физически взаимодействующих частей, ибо взаимодействие физических частей, управляемое абстрактными объектами, может быть в вычислителе — определение системы как взаимодействующих частей с получением эмерджентности сохраняется, только взаимодействие идёт в системе-создателе, а не целевой системе. По этой линии можно обсуждать системность в сложных случаях сообществ, которые вроде как существуют, но которые трудно обсуждать на основе взаимодействия членов этих сообществ друг с другом: взаимодействие происходит путём проведения операций рассуждения об этих сообществах в вычислителе создателя[498].

Эта линия работ на конструктивистское переописание системной онтологии второго поколения была продолжена при создании **Core Constructional Ontology**[499], которая предлагает конструктивистский взгляд на **теорию частей, множеств и отношений** и служит одним из уровней foundational ontology для **инженерных описаний**. Для неё более низким уровнем foundational ontology является

---

[494] John Baez, Symmetric Monoidal Categories: a Rosetta Stone, https://johncarlosbaez.wordpress.com/2021/05/28/symmetric-monoidal-categories-a-rosetta-stone/
[495] Constructive Mathematics, https://plato.stanford.edu/entries/mathematics-constructive/
[496] Mereology, https://plato.stanford.edu/entries/mereology/
[497] Kit Fine, Towards a Theory of Part, 2010, https://as.nyu.edu/content/dam/nyu-as/philosophy/documents/faculty-documents/fine/accessible_fine/Fine_Theory-Part.pdf
[498] Kit Fine, The Identity of Social Groups, 2020 https://metaphysicsjournal.com/articles/10.5334/met.45/
[499] Salvatore Florio, Core Constructional Ontology (CCO): a Constructional Theory of Parts, Sets, and Relations, 2021, https://gateway.newton.ac.uk/presentation/2021-04-22/29947

математика, а выше неё — соображения по выражению **4D мереотопологии пространства-времени** в **4-Dimensionalist Top Level Ontology**[500].

Использование какой-то **математики** (чаще всего в computational ontology это какие-то варианты логик первого порядка, но всё чаще и чаще это заход на морфизмы и, соответственно, теорию категорий) в качестве foundational ontology с пропуском всех остальных (**upper ontology, middle ontology**) **онтологических уровней** обычно для **физики**. Сама физика с точки зрения системной онтологии работает с **функциональными объектами** (например, «физическое тело»), роли которых выполняют разные объекты физического мира (стулья, камни, молекулы, фотоны). Далее в **догадках** и **экспериментах** выясняются особенности поведения этих функциональных объектов, они выражаются **математическими соотношениями/формулами**. При структурном совпадении формул, описывающих какие-то разные объекты, может делаться предположение об общей природе этих объектов как **рассуждение по аналогии/analogical reasoning**. Этот приём используется и для онтологий, выраженных логически в виде **решёток/lattices**, примером тут может быть VAE (VivoMind Analogy Engine)[501]. Примером использования такого приёма физиками могут служить также работы по **теории информации** («свободная энергия» в теории информации названа так в силу того, что в теории информации и термодинамике используются похожие формулы[502]). Так что есть тенденция использовать математические объекты и формульные отношения между ними так же и для формулирования **системной онтологии третьего поколения, то есть физического объяснения эволюции, приводящей к большим эволюционным переходам.**

Вся эта формализация, конечно, связана главным образом не с направлением порождения новых знаний в ходе **системного творчества**, а с направлением критики.

Третье поколение системного подхода: учёт времени эволюции

Vanchurin, Wolf, Koonin, Katsnelson заметили, что рассуждения по аналогии в классической термодинамике и теории информации, применённые к величинам/quantities в термодинамике, **машинном обучении** и эволюционной биологии[503] ведут к неожиданной общей для них всех онтологии (**framework**). Другими словами, эволюция могла бы быть описана как **обучение/познание/learning**, а обучение носит термодинамическую природу, то есть вполне физический феномен. На базе замеченных аналогий они сформулировали теорию эволюции как **многоуровневого познания/deep learning**[504] (deep — это многоуровневость представлений[505]). В этой теории движущая сила эволюции — это **конфликты** между объектами разных системных

---

[500] Chris Partridge, 4-Dimensionalist Top Level Ontology, https://gateway.newton.ac.uk/presentation/2021-04-22/29946

[501] John F. Sowa and Arun K. Majumdar, Analogical Reasoning, 2003, http://www.jfsowa.com/pubs/analog.htm

[502] Entropy in thermodynamics and information theory, https://en.wikipedia.org/wiki/Entropy_in_thermodynamics_and_information_theory

[503] Vitaly Vanchurin, Yuri I. Wolf, Eugene V. Koonin, Mikhail I. Katsnelson, Thermodynamics of evolution and the origin of life, 2022, https://www.pnas.org/doi/full/10.1073/pnas.2120042119

[504] Vitaly Vanchurin, Yuri I. Wolf, Mikhail I. Katsnelson and Eugene V. Koonin, Toward a theory of evolution as multilevel learning, 2022, https://www.pnas.org/doi/10.1073/pnas.2120037119

[505] https://ailev.livejournal.com/1045081.html

уровней, приводящие к **неустроенностям/frustrations**[506]. Это понятие неустроенности было введено в системный язык по итогам исследований спиновых стёкол как примера поведения неэргодических (с памятью) систем и там оно означало геометрические неустроенности (невозможность устойчивой геометрии спинов в стёклах). Эволюция тем самым представляет собой процесс познания/обучения/learning, который сводится к решению **оптимизационной задачи** нахождения минимума свободной энергии за счёт изменения структуры многочисленных **системных уровней**. В ходе этого эволюция находит **квазиминимумы**, но не **абсолютный минимум**. Время от времени возникает **скачок роста сложности** (большой эволюционный переход, появление ещё одного системного уровня, ещё одного вида целых систем), что даёт резкую минимизацию свободной энергии эволюционирующей системы, но всё-таки это просто очередной квазиминимум, а не абсолютный минимум. В этой серии работ[507] была показана **физическая природа роста сложности систем** (на примере биологических систем, но рассуждения там безмасштабны, то есть биологичность систем или наличие сознания не влияют на выводы работы). Так что в системной онтологии появляются ещё и конфликты между системными уровнями, и явление неустроенностей из-за этих конфликтов, и вывод о неизбежности роста сложности системы как увеличении числа системных уровней.

Все эти исследователи контактируют друг с другом (например, были обсуждения между Vanchurin и Friston), потому как в основе большинства этих работ лежит термодинамика и понимание того, что все системы реализуют принцип минимизации свободной энергии и во время своего существования, и во время своего создания, и в ходе эволюции. Перевод традиционных формул для выражения этой онтологии (сейчас эта онтология выражена в привычном физикам формульном виде, традиционном для термодинамических выкладок) в конструктивистскую форму теории категорий — отдельная тема, но такая постановка задачи для математиков более-менее привычна и можно обсуждать программу исследований на эту тему. То же можно сказать про переформулирование **дифференциальной формы** в **квантовоподобную/quantum-like**[508], что могло бы увеличить точность физического моделирования биологической эволюции и техно-эволюции (меметической эволюции) в силу квантованности/цифровости большинства описываемых явлений, а заодно уменьшить время вычислений за счёт того, что вычисления ведутся в дискретных точках, а не в каждой точке непрерывных функций.

Третье поколение системного подхода учитывает, что эволюционируют прежде всего знания (в том числе знания о системах), а не только сами системы, понимаемые уже ближе к биологическим видам, это близко к изменению воззрения биологов на эволюцию как прежде всего эволюцию генома, а не эволюцию фенома. В основе современных биологических классификаций уже лежат не феномы организмов, а их геномы.

Таким образом, современная/SoTA (третьего поколения системного подхода) онтология системы:

---

[506] Geometrical frustration, https://en.wikipedia.org/wiki/Geometrical_frustration

[507] Yuri I. Wolf, Mikhail I. Katsnelson, and Eugene V. Koonin, Physical foundations of biological complexity, 2018, https://www.pnas.org/doi/10.1073/pnas.1807890115

[508] Irina Basieva, Andrei Khrennikov, Masanao Ozawabc, Quantum-like modeling in biology with open quantum systems and instruments, https://www.sciencedirect.com/science/article/pii/S0303264720301994

- даёт типы объектов для многоуровневого наведения внимания с целью обеспечения эволюции целевых систем (continuous everything) системами-создателями
- рассматривает минимально три времени существования системы: operations, construction/evolving of phenome, evolution/development of genome/memome — базируется на физике, математике и computer science
- рассматривает системы как стабильные сущности в рамках минимального физикализма (в том числе системы активны по отношению к себе и окружению, системы ищут минимум свободной энергии, ведя active/embodied inference, независимо от уровня своей «разумности»)
- даёт безмасштабные описания систем (феномены квантовой физики тем самым учитываются), объясняет возникновение системных уровней (рост сложности) в силу многоуровневой оптимизации для достижения минимума свободной энергии
- мереология выражается уже не через «вечные классы» и отношения между ними, а через морфизмы, отражающие операции с физическими системами в ходе их взаимодействия. Учитывается также и то, что операции с абстрактными объектами выполняются физическими системами-создателями с (универсальным в смысле эквивалентности машине Тьюринга) вычислителем в их составе, «математик, рассуждающий о математике — тоже физический объект».

Текущее развитие системного подхода прежде всего связано с развитием знаний о распределённых представлениях, ибо все три поколения системного подхода развивались в ситуации, когда природа распределённых представлений понималась плохо, это были больше поколения системного подхода, выражаемые формально в локальных представлениях и поэтому удобные для критики:

На сегодня ещё не решена задача создания общей локально-распределённой семантики, объединяющей классическую семантику локальных представлений на

базе треугольника Фреге, объёдиняющую локальные представления о знаках (семиотику), понятиях (абстрактные объекты даёт математика) и физике (объекты физического мира, отражаемые понятиями, выражаемыми в знаках, даёт физика) и нарождающуюся в работах по большим языковым моделям в AI распределённую семантику, в которой на месте семиотики стоит **обучение представлениям/representations learning** (главным образом векторным/распределённым/distributed[509], отвязанным/disentangled[510], непрерывным/continuous[511] представлениям):

Вполне возможно, что развитие теории распределённых представлений и обобщение семантики до включения не только семиотики, но и обучения представлениям приведёт к появлению **четвёртого поколения системного подхода**, но это пока только гипотеза.

Что дальше

Вот вы и завершили освоение руководства по системному мышлению. Что дальше? Конечно, переходить к освоению руководства по методологии. Если что-то и осталось непонятным, мы рекомендуем продолжать освоение руководств полной программы рабочего развития инженеров-менеджеров, ибо каждое очередное руководство в этой программе как-то повторяет материал предыдущих руководств и даёт примеры использования введённых в этих руководствах понятий — поэтому многие вопросы будут сняты. Не надо застревать на материале одного руководства, надо продвигаться вперёд и вперёд.

Тем не менее, подумайте над тем, что вы можете прямо сейчас получить пользу от освоенных методов системного мышления. Просто думайте так в своих проектах, как наставляется в руководстве («мышление письмом»: составляйте и заполняйте таблички в универсальных моделерах или даже в более простых, чем таблички, формах — списках, просто текстах, это и есть системное мышление в его текущей версии!), используйте мета-мета-модель системного мышления из нашего

[509] https://deepai.org/machine-learning-glossary-and-terms/distributed-representation
[510] https://deepai.org/machine-learning-glossary-and-terms/disentangled-representation-learning
[511] https://www.sciencedirect.com/science/article/pii/S2666651021000206

руководства.

Добейтесь беглости в системном мышлении, не откладывайте на «когда-нибудь, а то некогда». Системное мышление — это заполнение таких же табличек, какие вы заполняли при освоении работы по нашему руководству, это не что-то особенное, что вы ещё не пробовали делать. Заполняйте эти таблички для ваших текущих проектов, заполняйте для новых проектов, не бойтесь модифицировать эти таблички. Используйте какие угодно моделеры для этих табличек, вам должно быть уже легко это делать. Это соответствует «Проекту-2» из подраздела «Прикладная функциональность против универсальной конструкции», то есть вам нужно научиться рассуждать системно о ваших предметных областях (мета-модели), удерживая во внимании типы понятий системного подхода (мета-мета-модель) — но делать это бегло, без особых затрат ресурсов, длинными ходами мысли не на пару понятий мета-мета-модели, а сразу на десяток-другой понятий. Помним, что за месяц занятий музыкой или за месяц тренировок, или за месяц занятий иностранным языком результаты будут «никакими», за год — вы будете существенно отличаться от «людей с улицы», за три года — можно будет этим зарабатывать деньги, десяток лет — и можно говорить о том, чтобы побеждать в мировой конкуренции. К системному мышлению это относится в полной мере.

Для чего надо использовать наставления руководства? Для критики многочисленных идей, которые порождаете вы сами, а также ваши коллеги. Вы сэкономите много времени, если не будете делать ошибок, на которые указываем в руководстве. Вы сэкономите много времени, если будете больше думать о тихо молчащем в проекте важном и меньше о неважном, которое кричит и привлекает (можно даже сказать — отвлекает) ваше внимание. Вы уже многое знаете, так используйте это знание! Сделайте доклад на конференции о своих успехах, напишите пост в клубе (не только «мышление системным моделированием», но и «мышление письмом» существенно помогает поднять беглость системного мышления!).

Если материал оказался трудным, вы не очень поняли, о чём говорится в руководстве, то попробуйте осваивать работу по руководству с наставником, возможно, в составе корпоративной группы стажировки с мастерами мастерской инженеров-менеджеров. Варианты освоения работы по руководствам в ходе стажировок с наставниками ищите на вебсайте мастерской инженеров-менеджеров[512]. Наш опыт показывает, что освоение работы по руководствам с наставником в разы эффективней. И помним, что если вы сэкономили «прохождением мимо» стажировки по пререквизиту (руководству по рациональной работе), то вряд ли вы сэкономили, вряд ли наше руководство оказалось вами полностью понято (и это ещё мягкая формулировка!).

Повторим: после освоения работы по нашему руководству по системному мышлению мы рекомендуем не делать себе каникулы, но продолжать рабочее развитие, изучая работу по руководствам по методологии, а затем работу по руководствам по инженерным методам — по системной инженерии, по инженерии личности, по системному менеджменту. Это направление можно назвать **«дьявол в деталях»**: углубиться в изучение отдельных инженерных, менеджерских, творческих, социальных методов работы, то есть изучать отдельные прикладные методы работы. Это традиционное обучение инженеров-менеджеров инженерии,

[512] https://system-school.ru/

менеджменту, другим специальностям в их связи с реальной жизнью, результативностью в рабочих проектах. Системное мышление позволит удерживать целостность изучаемого набора кругозорных и прикладных методов работы, а также переносить накопленный опыт из проекта в проект. Это образование инженера, менеджера, предпринимателя/основателя компаний, деятеля искусств, общественного деятеля/политика и т.д.: человека, который овладел системным мышлением и использует его для изменения мира к лучшему.

Но можно предложить и альтернативный вариант, **«ангел в абстракциях»** («знание принципов освобождает от знания фактов»): обобщить предлагаемое системное мышление с целью поднятия беглости в использовании его приёмов и распространения его на самые разные виды систем — для экспансии системного мышления на новые методы работы, новые классы систем (например, системы машинного обучения и искусственного интеллекта, системы из молодёжных субкультур и т.д.). По этому направлению можно углублять свои знания интелект-стека фундаментальных методов мышления программы исследовательского развития. Это развитие человека, который занимается бесконечным познанием, бесконечным развитием, бесконечным фундаментальным образованием. Хорошим подспорьем на этом пути будет изучение литературы, которая упоминается в нашем руководстве, особенно литературы из последнего раздела руководства (этот раздел исследовательского, а не рабочего уровня).