Основные задачи и методы проектирования программных средств

Лекция 9

Роль и обязанности архитектора программных средств

Овчинников П.Е.

МГТУ «СТАНКИН»,

ст.преподаватель кафедры ИС

12.11.2020

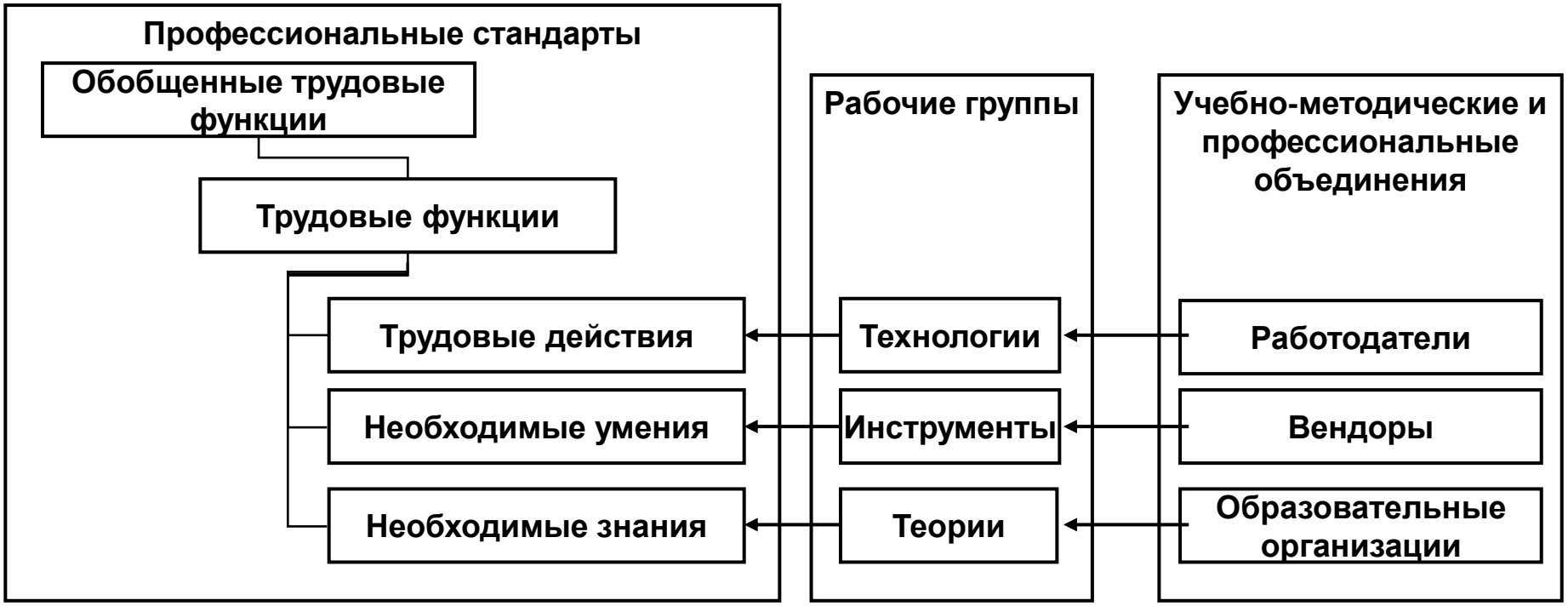
Структура профессиональных стандартов

Трудовой кодекс Российской Федерации" от 30.12.2001 N 197-ФЗ

Статья 195.1. Понятия квалификации работника, профессионального стандарта

Квалификация работника - уровень **знаний**, **умений**, профессиональных **навыков** и **опыта** работы работника.

Профессиональный стандарт - **характеристика квалификации,** необходимой работнику для осуществления определенного вида профессиональной деятельности, в том числе выполнения определенной трудовой функции.



Ограничения профессиональных стандартов

Технология

машины или **оборудование**, разработанные с использованием техники

англ. machinery or equipment designed using technology

Инструмент

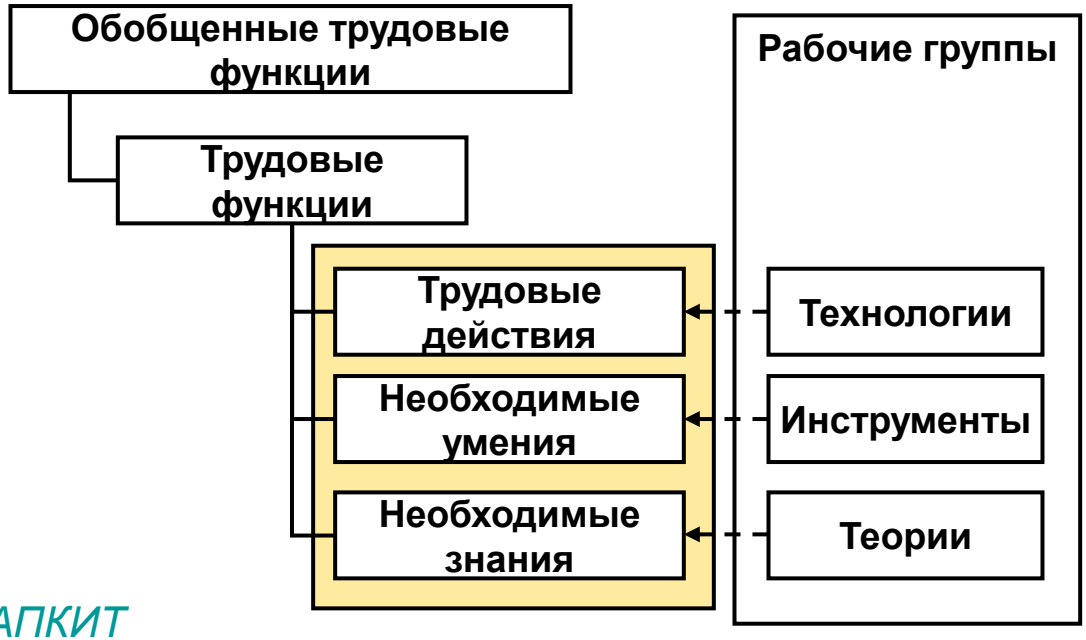
устройство или **оборудование** для выполнения работы

англ.a tool or implement

Технология (теория)

применение научных знаний для практических целей, особенно в промышленности

англ. the application of scientific knowledge for practical purposes, especially in industry



ТФ включается в ПС, если:

✔ технологии **востребованы**

и

✔ инструменты **доступны**

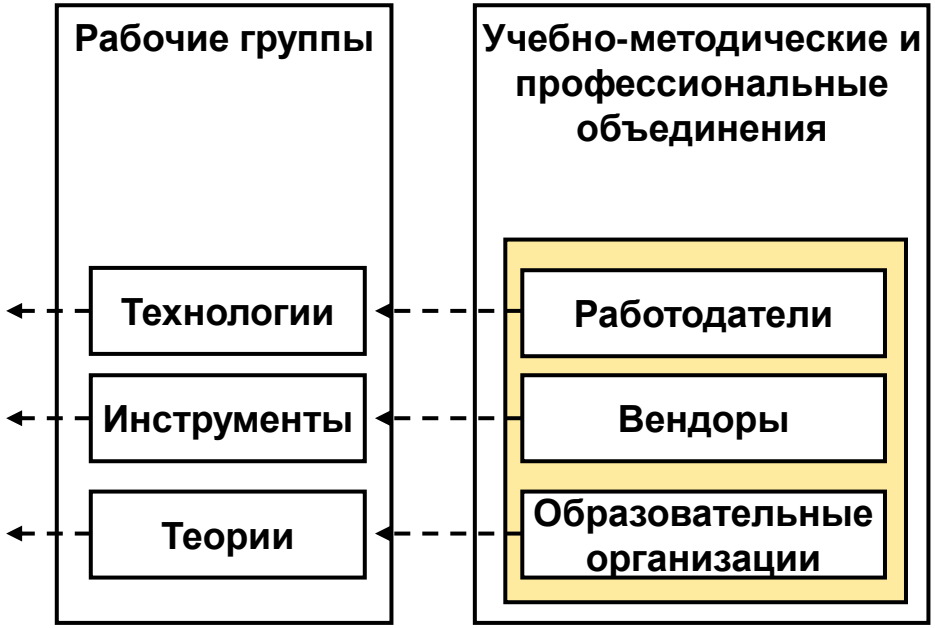
и

✔ способы применения **известны**

Источники профессиональных стандартов

Формирование требований федеральных государственных образовательных стандартов профессионального образования в части профессиональной компетенции осуществляется на основе соответствующих профессиональных стандартов (при наличии).

Профессиональное обучение направлено на **приобретение** лицами различного возраста **профессиональной компетенции**, в том числе для работы с конкретным **оборудованием**, **технологиями**, **аппаратно-программными** и иными профессиональными **средствами**, **получение** указанными лицами **квалификации** по профессии рабочего, должности служащего и присвоение им (при наличии) квалификационных разрядов, классов, категорий по профессии рабочего или должности служащего без изменения уровня образования.



ТФ включается в ПС, если:

✔ технологии **востребованы**

и

✔ инструменты **доступны**

и

✔ способы применения **известны**

Применение профессиональных стандартов

Инструкция

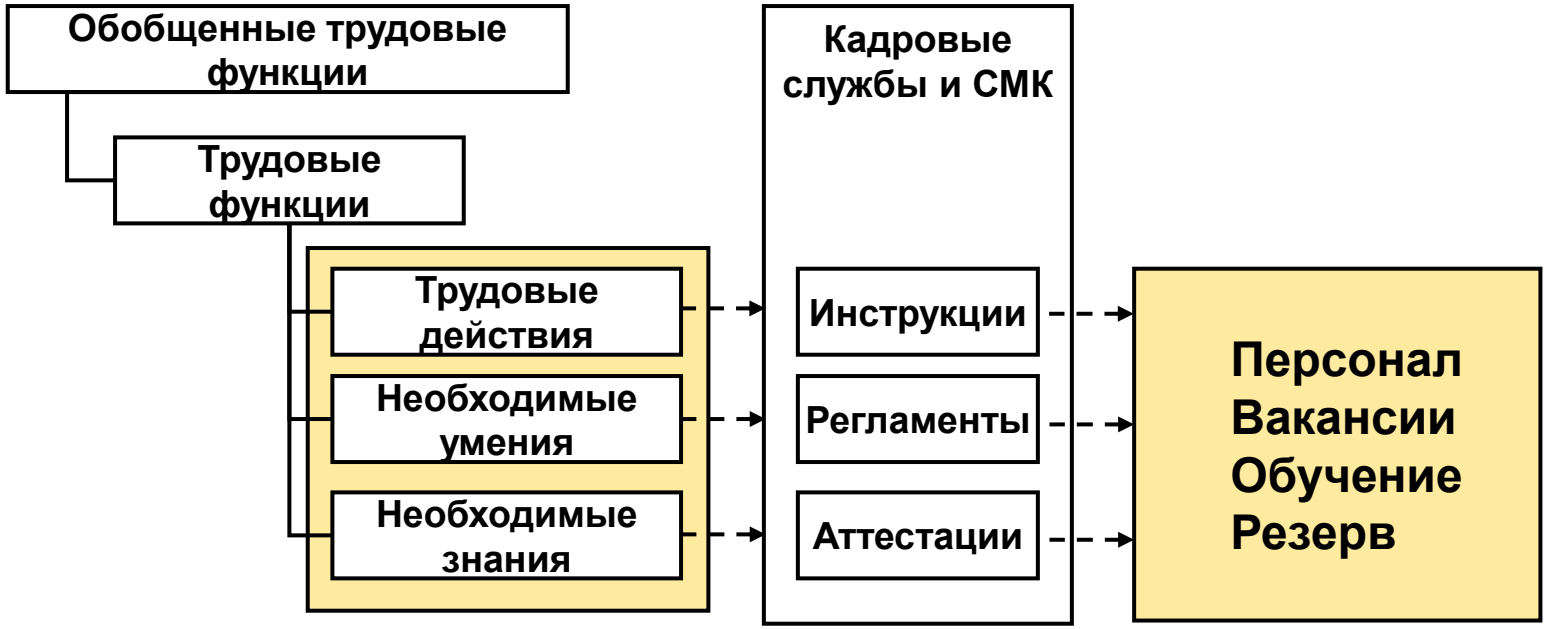
Документ, регламентирующей производственные **полномочия** и **обязанности** работника на конкретном предприятии (в организации) с учетом конкретных особенностей организации трудового процесса

Регламент

Документ, который перечисляет и **описывает** по порядку **этапы (шаги)**, которые должен предпринимать участник или группа участников для выполнения **бизнес-процесса**, как правило, с указанием требуемых сроков выполнения этапов (шагов).

Аттестация

проверка **уровня подготовки**, мастерства, квалификации работника



Профессия: архитектор

ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ

Архитектор программного обеспечения

67

Регистрационный номер

I. Общие сведения

Проектно-конструкторская деятельность | 06.003

(наименование вида профессиональной деятельности) | Код

Основная цель вида профессиональной деятельности:

Создание и сопровождение архитектуры программных средств, заключающейся
- в синтезе и документировании решений о структуре;
- компонентном устройстве;
- основных показателях назначения;
- порядке и способах реализации программных средств в рамках системной архитектуры;
- реализации требований к программным средствам;
- контроле реализации и ревизии решений

Архитектор: основные обязанности

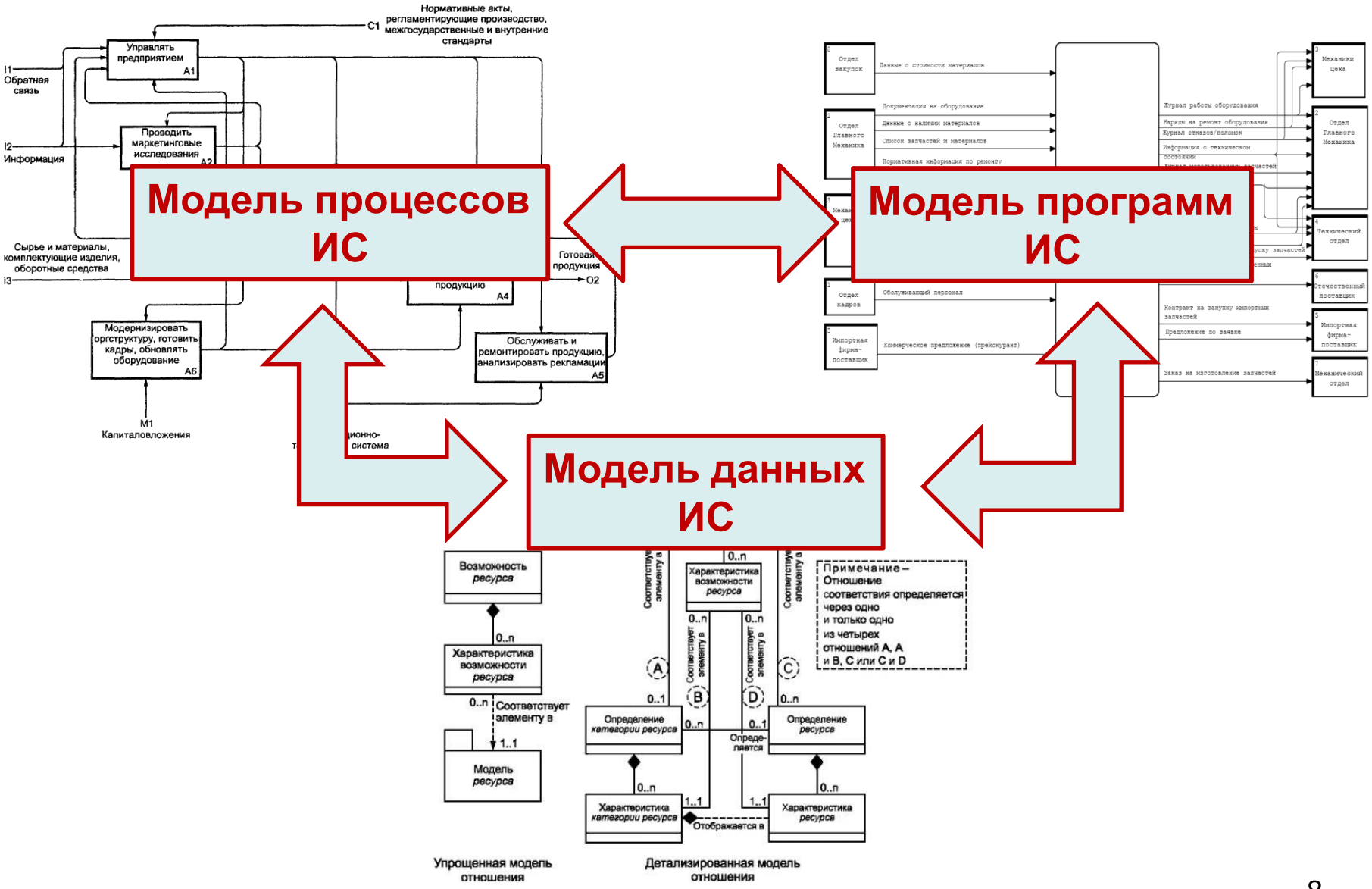
Основная цель вида профессиональной деятельности - **создание** и **сопровождение** архитектуры программных средств, заключающейся в **синтезе** и **документировании решений** о:

- структуре
- компонентном устройстве
- основных показателях назначения
- порядке и способах реализации программных средств в рамках системной архитектуры
- реализации требований к программным средствам
- контроле реализации и ревизии решений

Синтез + документирование = проектирование!

Проектирование = синтез + документирование!

Профессиональная задача: сохранение связей



Архитектор: определяет показатели назначения

ГОСТ 34.602-89 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы

2.6.1.3. В требованиях к показателям назначения АС приводят значения **параметров**, характеризующие степень **соответствия** системы ее **назначению**.

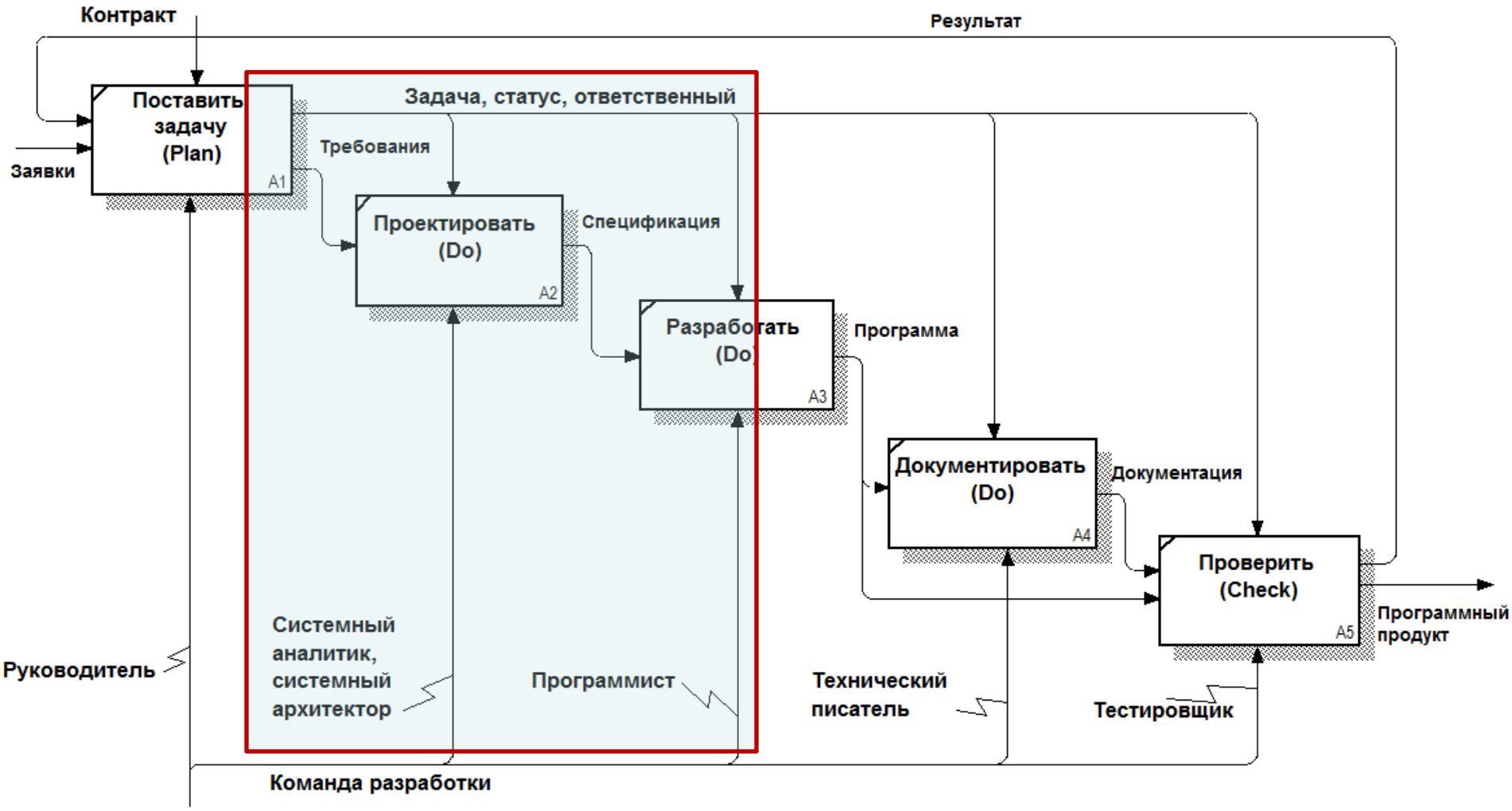
Для АСУ указывают:

- **степень приспособляемости** системы к изменению процессов и методов управления, к отклонениям параметров объекта управления

- **допустимые пределы модернизации** и развития системы

- **вероятностно-временные характеристики**, при которых **сохраняется целевое назначение** системы

Роль архитектора и модель разработки



Позиция исполнителя ИТ: «Программист»

II. Описание трудовых функций, входящих в профессиональный стандарт (функциональная карта вида профессиональной деятельности)

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
A	Разработка и отладка программного кода	3	Формализация и алгоритмизация поставленных задач	A/01.3	3
			Написание программного кода с использованием языков программирования, определения и манипулирования данными	A/02.3	3
			Оформление программного кода в соответствии с установленными требованиями	A/03.3	3
			Работа с системой контроля версий	A/04.3	3
			Проверка и отладка программного кода	A/05.3	3
B	Проверка работоспособности и рефакторинг кода программного обеспечения	4	Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	B/01.4	4
			Разработка тестовых наборов данных	B/02.4	4
			Проверка работоспособности программного обеспечения	B/03.4	4
			Рефакторинг и оптимизация программного кода	B/04.4	4
			Исправление дефектов, зафиксированных в базе данных дефектов	B/04.5	4

Позиция заказчика ИТ: «Менеджер по информационным технологиям»

II. Описание трудовых функций, входящих в профессиональный стандарт (функциональная карта вида профессиональной деятельности)

Обобщенные трудовые функции			Трудовые функции		
код	наименование	уровень квалификации	наименование	код	уровень (подуровень) квалификации
A	Управление ресурсами ИТ	6	Управление качеством ресурсов ИТ	A/01.6	6
			Управление ИТ-инфраструктурой	A/02.6	6
			Управление расходами на ИТ	A/03.6	6
			Управление изменениями ресурсов ИТ	A/04.6	6
			Управление отношениями с поставщиками и потребителями ресурсов ИТ	A/05.6	6
			Управление персоналом, обслуживающим ресурсы ИТ	A/06.6	6
			Управление информационной безопасностью ресурсов ИТ	A/07.6	6
B	Управление сервисами ИТ	7	Управление договорами об уровне предоставления сервисов ИТ (SLA)	B/01.7	7
			Управление ИТ-проектами	B/02.7	7
			Управление моделью предоставления сервисов ИТ	B/03.7	7
			Управление изменениями сервисов ИТ	B/04.7	7
			Управление отношениями с пользователями и поставщиками сервисов ИТ	B/05.7	7
			Управление персоналом, осуществляющим предоставление сервисов ИТ	B/06.7	7
			Управление непрерывностью сервисов ИТ	B/07.7	7
C	Управление информационной средой	8	Управление стратегией ИТ	C/01.8	8
			Управление программами и портфелями ИТ-проектов	C/02.8	8
			Управление формированием и внедрением системы показателей оценки эффективности ИТ	C/03.8	8
			Управление изменениями информационной среды	C/04.8	8
			Управление отношениями с поставщиками и потребителями	C/05.8	8

Понятия аспекта и сложности: система

Внешняя среда

Объект = Система

Граница системы

Субъект = Наблюдатель

Р 50.1.028-2001

Понятия аспекта и сложности: система

Сложная система (Википедия) — система, состоящая из множества взаимодействующих составляющих (подсистем), вследствие чего она приобретает новые свойства, которые отсутствуют на подсистемном уровне и не могут быть сведены к свойствам подсистемного уровня.

Американский экономист Кеннет Боулдинг предложил шкалу сложности систем, состоящую из девяти уровней

1. Уровень **статической структуры**. К таким системам можно отнести: расположение электронов в атоме, строение кристалла, анатомию животного и т. п.
2. Простые **детерминированные динамические системы**. Примеры: Солнечная система, механическое устройство, структура теории наук вроде физики и химии.
3. Уровень **управляющего механизма** или **кибернетической системы**, уровень термостата. Система характерна тем, что стремится к сохранению равновесия.
4. Уровень **открытой** или **самосохраняющейся** системы, уровень клетки. Кроме биологических объектов, к этому уровню можно отнести реки и пожары.
5. Уровень **генетического сообщества**. Примерами могут являться растения. Характерен специализацией клеток.
6. Уровень **животных**. Системы характеризуются мобильностью, целесообразным поведением, самосохранением
7. Уровень **человека**. Самосознание, отличное от простого самосохранения. Рефлексия. Речь.
8. Уровень **социальной организации**.
9. Уровень **трансцендентальных систем**, не поддающихся анализу, но обладающих структурой.

Понятия аспекта и сложности: редукция

Редукция (лат. *reductio* — сведение, возведение, приведение обратно) — логический приём преобразования каких-либо данных к более удобному с какой-либо точки зрения виду; **сведение сложного к более простому**, доступному для анализа или решения.

Декомпозиция — **разделение целого на части**. Также декомпозиция — это научный метод, использующий структуру задачи и позволяющий заменить решение одной большой задачи решением серии меньших задач, пусть и взаимосвязанных, но более простых.

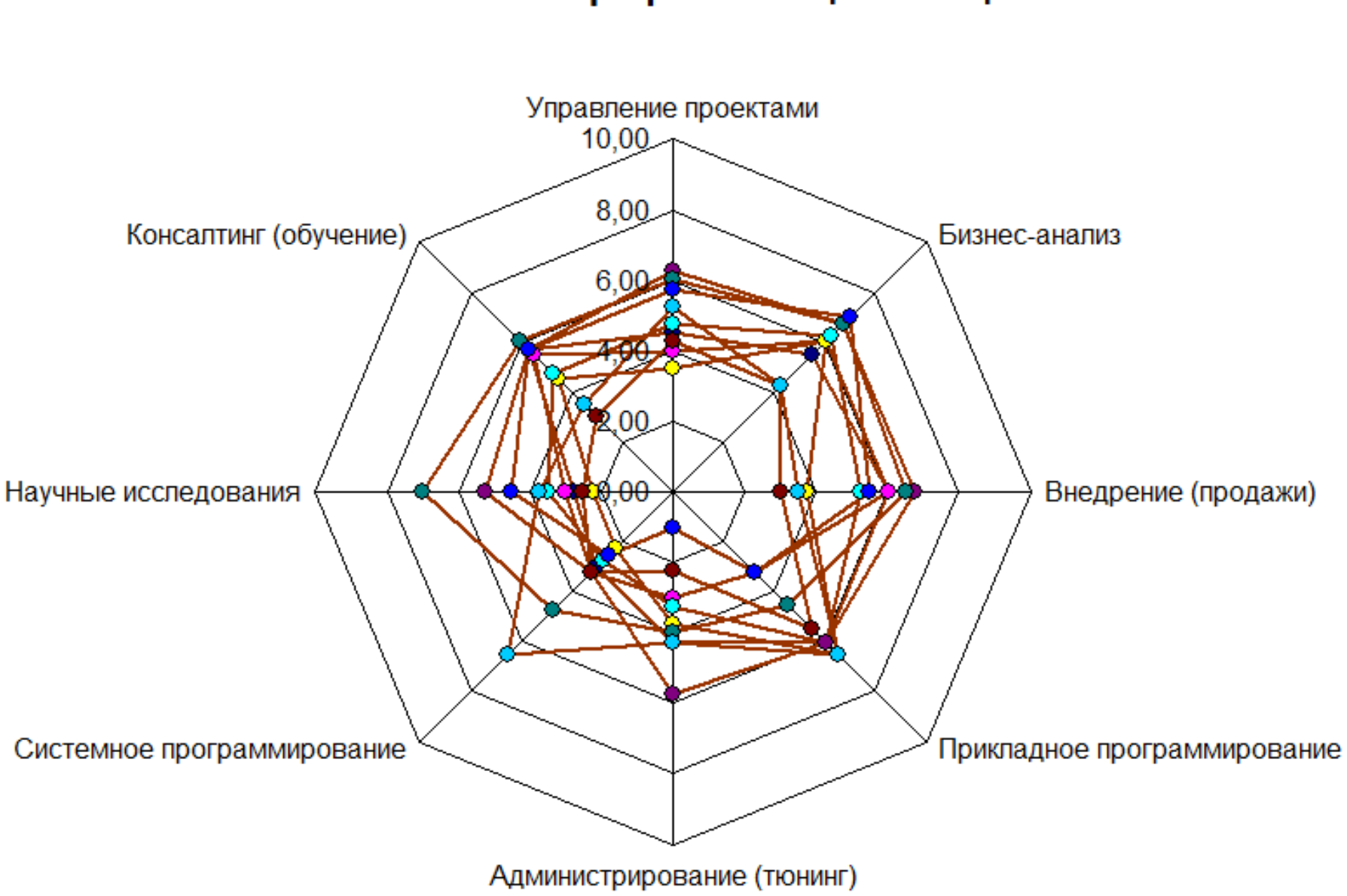
Декомпозиция, как процесс расчленения, позволяет рассматривать любую исследуемую систему как сложную, состоящую из отдельных взаимосвязанных подсистем, которые, в свою очередь, также могут быть расчленены на части. В качестве систем могут выступать не только материальные объекты, но и процессы, явления и понятия.

Иерархическая структура работ (ИСР) (англ. *Work Breakdown Structure, WBS*; иногда **Структура декомпозиции работ, СДР**)

иерархическое разбиение **всей работы**, которую необходимо выполнить для достижения целей проекта, на **более мелкие операции** и действия до такого уровня, на котором **способы** выполнения этих действий вполне **ясны** и соответствующие **работы** могут быть **оценены** и **спланированы**.

Многообразие профессий: упрощаем задачу

Реальный опыт



Выбор и распределение ролей

Понятия аспекта и сложности: архитектура

ГОСТ Р 57100-2016/ISO/IEC/IEEE 42010:2011 Системная и программная инженерия. Описание архитектуры

архитектура (системы) (architecture)

основные **понятия** или **свойства** системы в **окружающей среде**, воплощенной в ее **элементах**, **отношениях** и конкретных принципах ее **проекта** и **развития**

описание архитектуры (architecture description)

рабочий продукт, используемый для **выражения архитектуры**

структура архитектуры (architecture framework)

условности, принципы и практики для описания архитектур, установленные в пределах заданной области применения и/или объединения заинтересованных сторон

процесс архитектуризации (architecting)

процесс **понимания**, **определения**, **выражения**, **документирования**, взаимодействия, соответствующей сертификации при реализации, сопровождении и улучшении архитектуры в жизненном цикле системы

ГОСТ Р 57100-2016

Понятия аспекта и сложности: метрики

Показа́тель — обобщённая характеристика какого-либо объекта, процесса или его результата, понятия или их свойств, обычно выраженная **в числовой форме**

Крите́рий (др.-греч. κριτήριον — **способность различения**, средство суждения, мерило) — признак, основание, правило принятия решения по оценке чего-либо на соответствие предъявленным требованиям (мере)

Критерий в квалиметрии — **условие**, накладывающееся на показатель свойства предмета исследования

Ме́трика — *матем.* правило **определения расстояния** между любыми двумя точками , *техн.* вычисляемая величина, характеризующее какое-либо явление

Ме́ра — философская категория, означающая **единство** качественной и количественной определённостей некоторого предмета. Эта категория обобщает способы и результаты измерения предметов. Анализ меры исходит из важности интервала изменений количественных величин, в рамках которого можно говорить о **сохранении качества предмета**

Критерий (Википедия)

Понятие аспекта и сложности: система

Аспект - точка зрения, сторона, с которой рассматривается какое-либо явление, понятие

Р 50.1.028-2001 Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования

точка зрения:

Указание на **должностное лицо** или **подразделение** организации, с позиции которого разрабатывается модель

Для каждой модели точка зрения **единственная**

Точка зрения определяет, что и в каком разрезе можно увидеть **в пределах контекста** модели.

Изменение точки зрения приводит к рассмотрению **других аспектов** объекта

Аспекты, важные с одной точки зрения, могут не появиться в модели, разрабатываемой с другой точки зрения **на тот же самый объект**

Р 50.1.028-2001

Понятие референтной модели

ГОСТ Р ИСО 15704-2008 Промышленные автоматизированные системы. Требования к стандартным архитектурам и методологиям предприятия

архитектура:

описание (**модель**) основного устройства (структуры) и связей частей системы (физического или концептуального объекта или сущности)

модель:

абстрактное **представление реальности** в любой форме (включая математическую, физическую, символическую, графическую или описательную) для представления определенного **аспекта** этой реальности для **ответа на** рассматриваемые **вопросы**

модель предприятия:

представление о том, чего предприятие **планирует достичь** и **как оно работает**

инжиниринг предприятия:

дисциплина, применяемая для выполнения любых **работ по созданию**, **изменению** или **реорганизации** любого предприятия

ГОСТ Р ИСО 15704-2008

Понятие референтной модели

ГОСТ Р ИСО 15704-2008 Промышленные автоматизированные системы. Требования к стандартным архитектурам и методологиям предприятия

Предприятия работают в **неопределенных** рыночных и внешних условиях, поэтому процесс инжиниринга предприятия должен носить постоянный характер. Это означает, что персонал предприятия выполняет самые различные функции в рамках концепции предприятия и постоянной миссии производственных правил, производственных процессов, организационных структур, поддержки ресурсов и услуг. Из-за **высокого уровня сложности** процессов инжиниринга предприятия постоянно возникает необходимость в применении средств, обеспечивающих проведение оценки, структуризацию, координацию и поддержку процессов инжиниринга.

Стандартные архитектуры предприятия, подкрепленные стандартными методологиями, обеспечивают общие приемлемые методы для организации и координации проектов в области инжиниринга. Принимая и адаптируя в соответствии с производственными потребностями стандартную методологию и архитектуру, персонал предприятия может участвовать в инжиниринговых проектах предприятия, улучшая деятельность предприятия и применение имеющихся ресурсов.

ГОСТ Р ИСО 15704-2008

Понятие референтной модели

ГОСТ Р ИСО 15704-2008 Промышленные автоматизированные системы. Требования к стандартным архитектурам и методологиям предприятия

Среда **GERAM** (**Generalised Enterprise Reference Architecture and Methodology**) идентифицирует наиболее важную составную часть **GERA** (**обобщенная стандартная архитектура предприятия**), основные понятия, которыми необходимо руководствоваться при инжиниринге и интеграции предприятия (например сущности предприятия, жизненные циклы и истории жизни сущностей предприятия).

GERAM проводит различие между **методологиями** инжиниринга предприятия (EEMs) **и языками моделирования** (EMLs), которые используются методологиями для описания и **моделями, структурами, содержанием и поведением** рассматриваемых сущностей предприятия. Эти языки моделирования обеспечивают моделирование **человеческой составляющей** в деятельности предприятия, а также доли бизнес-процессов и их вспомогательных технологий в работе предприятия.

В результате процесса моделирования разрабатываются модели предприятия (EMs), которые представляют все или часть операций предприятия, включая его производственные или сервисные задачи, организацию и менеджмент, а также систему управления и информационную систему.

ГОСТ Р ИСО 15704-2008

Понятие референтной модели

IEC PAS 63088:2017 Smart manufacturing - Reference architecture model industry 4.0 (RAMI4.0)

Умное производство. Модель эталонной архитектуры индустрии 4.0 (RAMI4.0)

физический мир (physical world):
все реально существующие объекты и люди

информационный мир (information world) цифровой мир или кибермир:
идеи, концепции, алгоритмы, модели и совокупности **представлений физических объектов и людей** в виртуальной среде

эталонная архитектура (reference architecture):
модель для описания архитектуры (для Индустрии 4.0), которую обычно используют и признают приемлемой (носит эталонный характер)

эталонная модель (reference model):
модель, которую обычно используют и признают приемлемой (носит **рекомендательный характер**) для **получения конкретных моделей**

IEC PAS 63088:2017

Понятие идеи. Источники идей

Идéя (др.-греч. ἰδέα «вид, форма; прообраз») в широком смысле — мысленный **прообраз** какого-либо **действия, предмета, явления, принципа**, выделяющий его основные, главные и существенные черты.

В науке и искусстве идеей называется главная мысль произведения или общий принцип теории или изобретения, вообще **замысел** или наиболее **существенная часть** замысла. В этом же смысле термин идея трактуется в сфере регулирования авторского права.

Существует множество способов генерации идей (например, метод **мозгового штурма**), но на практике, как правило, все проще.

Если посмотреть на появляющиеся на рынке стартапы, то можно заметить, что лишь **малая часть** из них **является уникальной**, а остальные созданы одним из трех методов:

- **Копирование** (клонирование): как правило, копируют идею (не продукт)
- **Трансформация**: дополнение существующей идеи
- **Комбинирование**: объединение нескольких идей

Источники идей: мозговой штурм

Метод мозгового штурма (**мозговой штурм**, **мозговая атака**, англ. *brainstorming*) — оперативный метод решения задач, в котором участники обсуждения генерируют максимальное количество решений задачи, в том числе самые фантастические и глупые. Затем из полученных вариантов выбираются самые лучшие решения, которые могут быть использованы на практике. Включает этап экспертной оценки.

10 правил эффективного мозгового штурма

1. **Предварительная подготовка.** Всем участникам мозгового штурма следует готовиться к нему заранее
2. **Много участников.** Чтобы мозговой штурм прошёл максимально эффективно нужно приглашать для участия в нём как можно больше людей, предлагающих, соответственно, больше идей.
3. **Уточнение поставленной задачи.** Перед началом штурма рекомендуется отвести некоторое время на дополнительное уточнение исследуемой проблемы.
4. **Записи.** На протяжении всей «игры нужно непременно вести записи и делать пометки. Причём, делать это должен каждый участник.
5. **Никакой критики.** Ни в коем случае не отвергайте предлагающиеся идеи, какими бы нелепыми или фантастическими они не казались.
6. **Максимальная генерация идей.** Каждый участник процесса должен понять, что ему нужно предлагать как можно больше идей..
7. **Привлечение других людей.** Если, например, во время штурма есть цель составить список из 100 решений, но этот уровень никак не достигается, можно привлечь к мозговому штурму людей, которые либо не присутствуют на штурме, либо вообще не имеют к нему никакого отношения.
8. **Модификация идей.** Для получения наилучшего результата можно соединять две идеи (и более) в одну. Особенно эффективно использовать этот приём, когда имеются варианты решения проблемы, предложенные людьми различного статуса, должности, ранга.
9. **Визуальное отображение.** Для удобства восприятия и повышения результативности мозгового штурма следует использовать маркерные доски, флэш-панели, плакаты, схемы, таблицы и т.п.
10. **Отрицательный результат.** Во время поиска решения и даже по его окончании представьте, что ситуация обернулась образом, прямо противоположным требуемому,

Понятие идеи. Источники идей

Бизнес-идея — это то, с чего начинается процесс создания любого бизнеса, в том числе и технологического

Рассчитывать на успех можно, если Вы сможете шаг за шагом пройти *путь* **создания цепочки ценности**:

- **придумать** новый **товар** или **услугу**, имеющих **ценность (value)** для потребителей
- **обеспечить реализацию** данной **ценности** через создание продукта или услуги
- найти **выход на рынок** с помощью бизнес-модели, которая позволит достичь максимальной прибыли.

Все начинается с идеи

Источники идей: ТРИЗ

Теория решения изобретательских задач (ТРИЗ) — наука об общих **законах развития искусственных систем**, объектом которой являются все искусственные системы

ТРИЗ является междисциплинарной наукой, призванной объединить и систематизировать знания тех областей, которые до сих пор было принято считать различными и несовместимыми.

Данная цель достигается в ТРИЗ за счёт анализа и выявления общих принципов, подходов, законов, закономерностей и тенденций развития в процессе научного познания.

Наиболее весомыми теориями, объединяемыми ТРИЗ, можно назвать следующие:

- Теория систем
- Системные исследования
- Теория принятия решений
- Синергетика
- Кибернетика
- Теория информации
- Теория управления

ТРИЗ (Викибук)

Источники идей: изобретательство

Наиболее распространены следующие **простейшие** приемы изобретательства

Аналогия

При решении задач идею решения можно получить путем применения **известного аналогичного решения**, «подсказанного» технической или художественной литературой, увиденного в кино или «подсмотренного» в природе. Выявлением и использованием «механизмов природы» занимается наука **бионика**. Она исследует объекты живого и растительного мира и выявляет принципы их действия и конструктивные особенности, с целью применения этих знаний в науке и технике.

Инверсия

Прием «инверсия» или «обратная аналогия» означает — **выполнить что-нибудь наоборот**. Для него характерны выражения: перевернуть вверх «ногами», вывернуть наизнанку, поменять местами и т. д.

Этот прием может означать, что если объект рассматривается снаружи, то, возможно, мы достигаем желательного результата, если будем его исследовать изнутри. Если какой-то объект расположен вертикально, то применение инверсии означает, что его ставят горизонтально — и наоборот.

Приемы изобретательства (Викибук)

Источники идей: изобретательство

Эмпатия

отождествление себя с **личностью другого**. Иногда об этом действии говорят «войти в шкуру другого», то есть поставить себя на место другого. Таким приемом часто пользуются артисты, писатели, художники и т. п. Подобным же образом можно использовать этот прием при разработке объекта.

Применение приема заключается в том, чтобы человек, посмотрел с позиции детали (с «ее точки зрения»), что можно сделать для устранения недостатков или для выполнения новых функций.

Фантазия

Прием фантазия связан с желанием получить то, чего желаешь. Использование фантазии для стимулирования новых идей заключается в размышлении над некоторыми **фантастическими** решениями, в которых при необходимости используются **нереальные вещи** или **сверхъестественные процессы**. Часто бывает полезно рассматривать **идеальные решения**, даже если это сопряжено с некоторой долей фантазии. Разумеется, есть надежда, что размышления о желательном может натолкнуть нас на новую идею или точку зрения, которая, в конечном счете, приведет к новому, осуществимому решению.

Приемы изобретательства (Викибук)

Модели качества программных средств

ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

Для выполнения разнообразных функций как в бизнесе, так и для персонального назначения в современных условиях все большее распространение получают программные продукты и преимущественно программные вычислительные системы.

Оценка может быть выполнена на основе определения необходимых и требуемых **характеристик качества**, связанных с **задачами заинтересованных сторон** и **целями системы**, включая характеристики качества, относящиеся к системе **программного обеспечения** и **данным**, а кроме того, и воздействие системы на ее заинтересованные стороны.

Важно, чтобы, по возможности, характеристики качества были **определены**, **измерены** и **оценены** с использованием проверенных или широко распространенных **показателей** и методов измерения.

ГОСТ Р ИСО/МЭК 25010-2015

Модели качества программных средств

ГОСТ Р ИСО/МЭК 25010-2015 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов

Модель **качества при использовании** определяет **пять** характеристик, связанных с результатами взаимодействия с системой: результативность, производительность, удовлетворенность, свободу от риска и покрытие контекста

Модель **качества продукта** сводит свойства качества системы/программного продукта к **восьми** характеристикам, которыми являются: функциональная пригодность, уровень производительности, совместимость, удобство пользования, надежность, защищенность, сопровождаемость и переносимость (мобильность)

ГОСТ Р ИСО/МЭК 25021-2014 Информационные технологии (ИТ). Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Элементы показателя качеств

качество данных (data quality): Степень, с которой характеристики данных удовлетворяют заявленным и подразумеваемым требованиям при использовании в заданных условиях (ИСО/МЭК 25012).

ГОСТ Р ИСО/МЭК 25021-2014

Терминология: авторский надзор

Федеральный закон "Технический регламент о безопасности зданий и сооружений" от 30.12.2009 N 384-ФЗ

авторский надзор

контроль лица, осуществившего подготовку проектной документации, **за соблюдением** в процессе строительства **требований проектной документации**

Второй закон Вейнберга: если бы строители строили здания так же, как программисты пишут программы, первый залетевший дятел разрушил бы цивилизацию (Законы Мерфи)

1С:КОНСАЛТИНГ. Авторский надзор корпоративных проектов

Проекты, выполняемые на **крупных предприятиях**, в группах компаний и холдингах с территориально распределённой структурой, предъявляют особые **требования к проектированию** информационных систем и **технологиям управления изменениями**.

Для повышения качества корпоративных проектов фирма «1С» предлагает услугу авторского надзора (курирования) проекта. Это позволит снизить риски проекта и повысить качество создаваемой информационной системы.

384-ФЗ

Терминология: авторский надзор

Ролевой состав группы авторского надзора проекта:

Куратор проекта

Осуществляет ежедневный контроль выполнения работ, соответствие сроков и состава работ на проекте запланированным, налаживает эффективные коммуникации и взаимодействие участников проекта, принимает участие в заседаниях органов управления проектом.

Системный архитектор

Обеспечивает достижение оптимальной функциональной и технической архитектуры системы, на периодической основе осуществляет аудит разрабатываемого программного продукта, вырабатывает рекомендации по технологическим вопросам разработки, достижения требуемых показателей производительности и надежности.

Методолог по участку учета (функциональному направлению)

Обеспечивают полную и качественную реализацию методологических и регламентных документов функциональной области, консультирует по сложным вопросам учета, осуществляет взаимодействие с разработчиками тиражного программного продукта.

Администратор проекта

Обеспечивает документооборот в группе надзора проекта.

Терминология: сопровождение

ГОСТ Р ИСО/МЭК 14764-2002. Информационная технология. Сопровождение программных средств

Сопровождение программных средств является одним **из основных процессов** их жизненного цикла, что описано в ГОСТ Р ИСО/МЭК 12207.

Процесс сопровождения состоит из работ и задач, реализуемых персоналом сопровождения (сопроводителем).

Из-за **ограничений** в **стоимости** и **сроках** разработки, а также отсутствия опыта в применении ГОСТ Р ИСО/МЭК 12207 программные средства нередко поставляют в "сыром" виде. Поэтому возникает необходимость в последующей **корректировке ошибок**, обнаруженных при их эксплуатации.

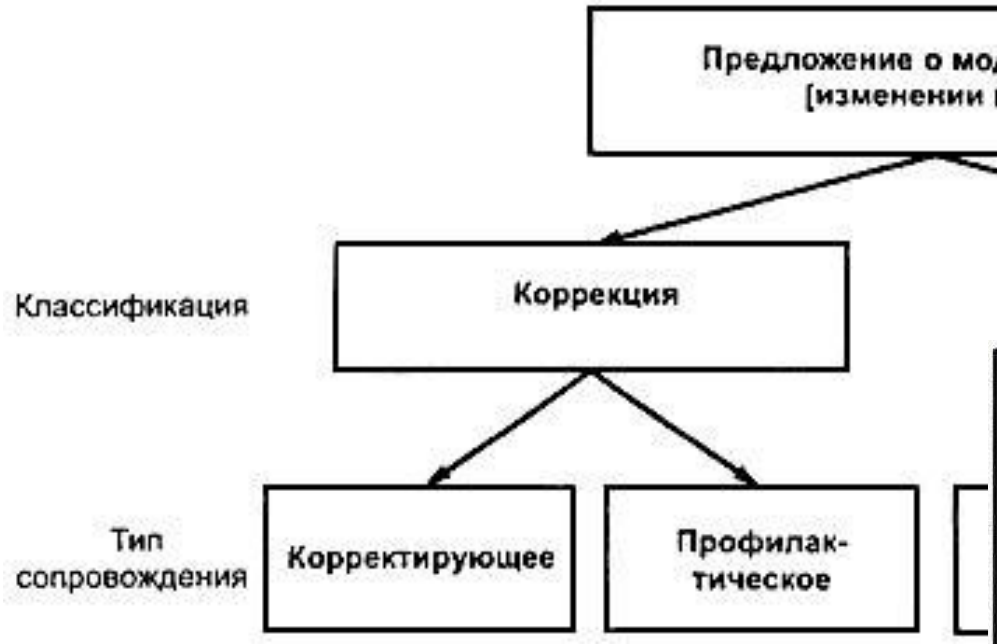
Часто необходимо **модернизировать** программное средство, чтобы удовлетворить **изменившимся требованиям** пользователя.

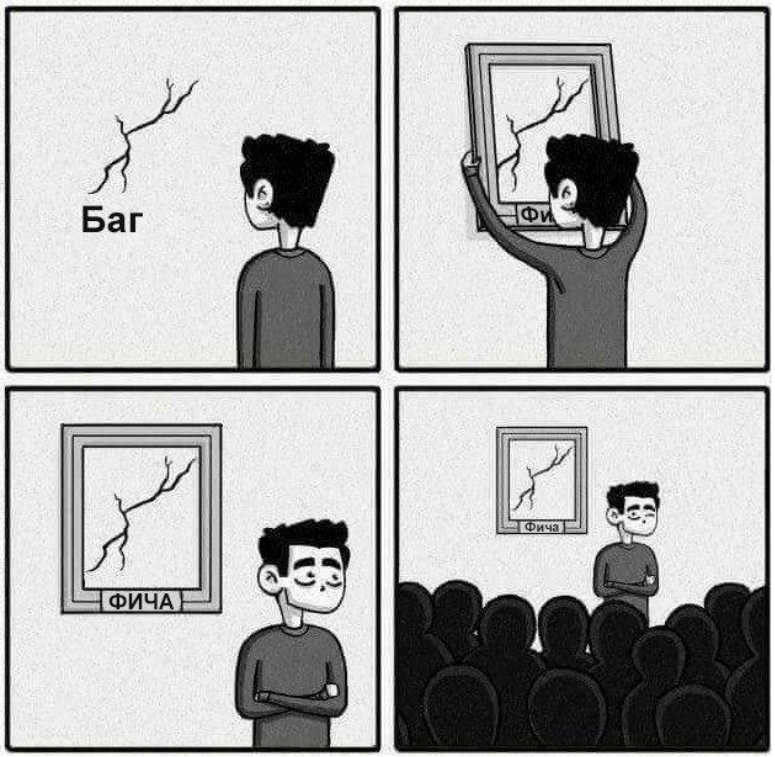
Сопровождение программного средства может в стоимостном выражении составлять наибольшую часть жизненного цикла.

ГОСТ Р ИСО/МЭК 14764-2002

Терминология: сопровождение

ГОСТ Р ИСО/МЭК 14764-2002. Информационная технология. Сопровождение программных средств

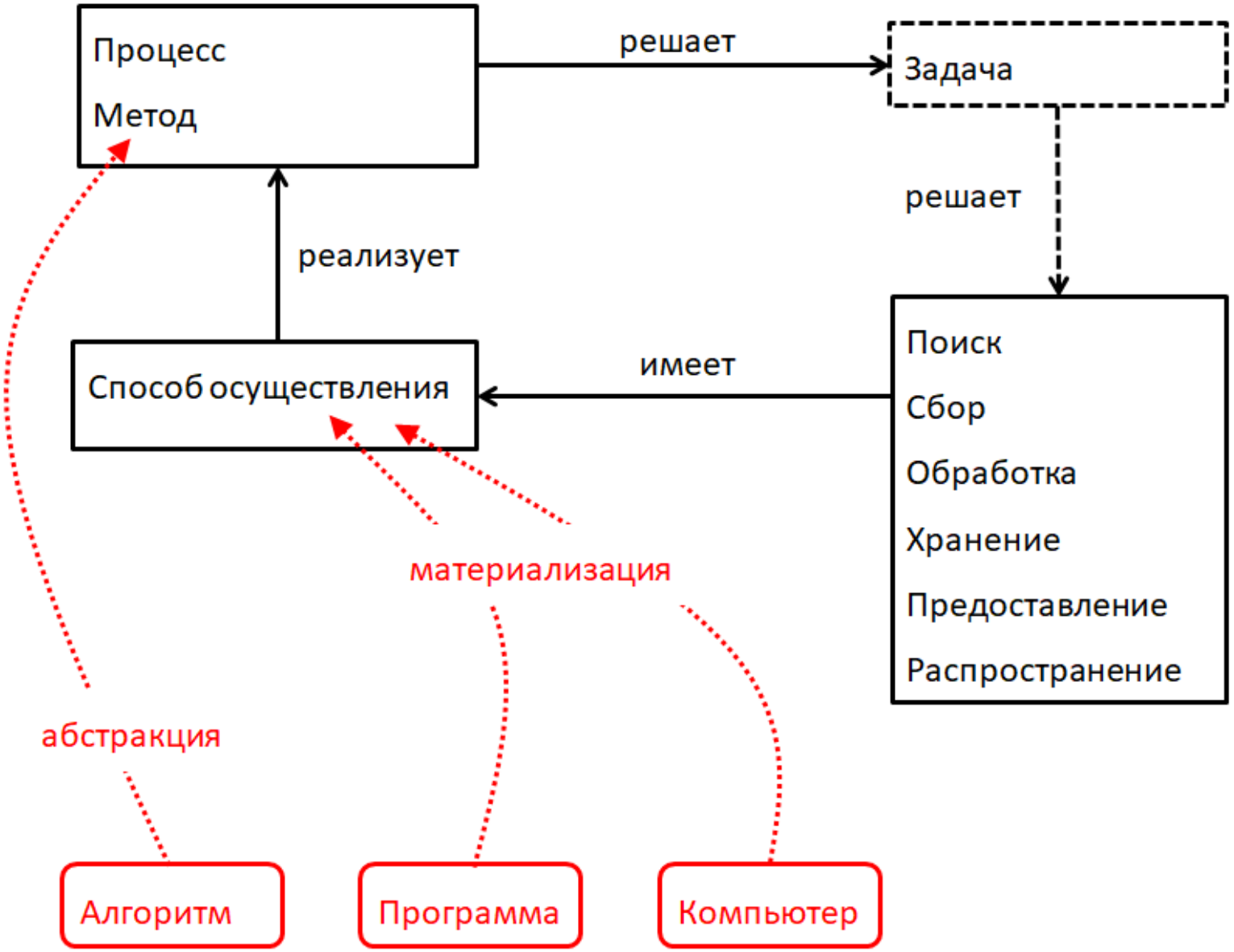




Баг *— это ошибка в компьютерной программе, приводящая к нежелательному результату,* ***фича*** *— русифицированное произношение слова feature [фИчэ] со значениями «свойство, особенность, характерная черта; функция, возможность»*

ГОСТ Р ИСО/МЭК 14764-2002

Объектно-ориентированный подход (инженерия знаний)



Объектно-ориентированный анализ (OOA) и проектирование (OOD)

Объектно-ориентированный анализ — это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области

Объектно-ориентированное проектирование — это методология проектирования, соединяющая в себе процесс **объектной декомпозиции** и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы

Объектно-ориентированное проектирование:

1) основывается на объектно-ориентированной декомпозиции
2) использует многообразие приемов представления моделей, отражающих:
   логическую (классы и объекты) и
   физическую (модули и процессы) структуру системы,

а также ее

статические и
динамические аспекты

Объектно-ориентированный подход (инженерия знаний)

Базовыми видами отношений для представления фреймов в виде семантической сети являются:

• отношение **AKO** (англ. a kind of), являющееся отношением **классификации** и позволяющее строить иерархические связи между объектами, реализующие основные принципы **наследования** свойств объектов

общее <- (AKO) – частное (только абстракции)

• отношение **HasPart** (англ. has part), являющееся отношением **вхождения** и позволяющее **декомпозировать** сложные объекты на их составляющие

целое <- (HasPart) – часть (только абстракции)

• отношение **ISA** (англ. is a), являющееся отношением между фреймом-**образцом** и фреймом-**экземпляром** , появляющимся **в результате классификации** (отнесения конкретного объекта к абстрактному классу)

абстрактное <- (ISA) – конкретное

Семантическая сеть Минский М. Фреймы для представления знаний. М.: Мир, 1979.

Объектно-ориентированное программирование

О системах программирования, основанных на фреймах, говорят, что они являются *объектно-ориентированными*

Каждый **фрейм** соответствует некоторому **объекту** предметной области, а **слоты содержат** описывающие этот объект **данные**, то есть в слотах находятся значения признаков объектов. Фрейм может быть представлен в виде **списка свойств**, а если использовать средства базы данных, то в виде **записи**

Более узкое определение: язык программирования является **объектно-ориентированным** тогда и только тогда, когда выполняются следующие условия:

- **поддерживаются объекты**, то есть абстракции данных, имеющие интерфейс в виде именованных операций и собственные данные, с ограничением доступа к ним
- **объекты** относятся к соответствующим **типам (классам)**
- типы (классы) могут **наследовать** атрибуты супертипов (суперклассов)

Фрейм

ООП: языки разметки

HTML (HyperText Markup Language)

```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html;
      <title>HTML Document</title>
   </head>
   <body>
      <p>
         <b>
            Этот текст будет полужирным,
            <i>а этот — ещё и курсивным</i>
         </b>
      </p>
   </body>
</html>
```

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eve</td>
    ckson</td>
    </td>
</table>
```

```
<iframe src="demo_iframe.htm" name="iframe_a"></iframe>

<p><a href="http://www.w3schools.com" target="iframe_a">W3Schools.com</a></p>
```

ООП: языки разметки

HTML5 (HyperText Markup Language 5)

```
<!DOCTYPE html>
<html>
   <head>
      <meta charset="utf-8">
      <title>
         (Это title) Пример страницы на HTML5
      </title>
   </head>
   <body>
      <header>
         <hgroup>
            <h1>
               Заголовок "h1" из hgroup
            </h1>
            <h2>
               Заголовок "h2" из hgroup
            </h2>
         </hgroup>
      </header>
      <nav>
         <menu>
            <li>
```

```
<video poster="movie.jpg" controls>
        <source src='movie.webm' type='video/webm; codecs="vp8.0, vorbis"'/>
        <source src='movie.ogv' type='video/ogg; codecs="theora, vorbis"'/>
        <source src='movie.mp4' type='video/mp4; codecs="avc1.4D401E, mp4a.40.2"'/>
        <p>This is fallback content</p>
</video>
```

ООП: управление отображением

CSS (Cascading Style Sheets, каскадные таблицы стилей)

```css
p.Big {
    font-family: arial, helvetica, sans-serif;
    color: maroon;
}
div#First {
    background-color: silver;
}
```

```css
p {
    font-family: arial, helvetica, sans-serif;
}
h2 {
    font-size: 20pt;
    color: red;
    background: white;
}
.note {
    color: red;
    background-color: yellow;
    font-weight: bold;
}
p#paragraph1 {
    padding-left: 10px;
}
a:hover {
    text-decoration: none;
}
#news p {
    color: blue;
}
[type="button"] {
    background-color: green;
}
```

ООП: функции клиентской части

JavaScript (JS)

```
window.onload = function() {
    var linkWithAlert = document.getElementById("alertLink");
    linkWithAlert.onclick = function() {
        return confirm('Вы уверены?');
    };
};
```

```
var fruits, text, fLen, i;

fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;
text = "<ul>";
for (i = 0; i < fLen; i++) {
    text += "<li>" + fruits[i] + "</li>";
}
```

```
var obj;
obj = document.getElementById("demo");
obj.innerHTML = "Hello";
```

```
xhttp.open("GET", "ajax_info.txt", false);
xhttp.send();
document.getElementById("demo").innerHTML = xhttp.responseText;
```

```
<script>
window.onload = downScripts;

function downScripts() {
    var element = document.createElement("script");
    element.src = "myScript.js";
    document.body.appendChild(element);
}
</script>
```

ООП: Управление поведением

JQuery

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
<script src="my_jquery_functions.js"></script>
</head>
```

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("[href]").hide();
});
</script>
</head>
<body>

<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<a href="http://www.w3schools.com/html/">HTML Tutorial</a>
<a href="http://www.w3schools.com/css/">CSS Tutorial</a>

</body>
```

```
$("button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");
});
```

```
$("button").click(function(){
    $.get("demo_test.asp", function(data, status){
        alert("Data: " + data + "\nStatus: " + status);
    });
});
```

ООП: определение данных

XML Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="country">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="country_name" type="xs:string"/>
        <xs:element name="population" type="xs:decimal"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
<grammar xmlns="http://relaxng.org/ns/structure/1.0">
   <start>
      <element name="book">
         <oneOrMore>
            <ref name="page"/>
         </oneOrMore>
      </element>
   </start>
   <define name="page">
      <element name="page">
         <text/>
      </element>
   </define>
</grammar>
```

XSD

ООП: обмен данными

XML

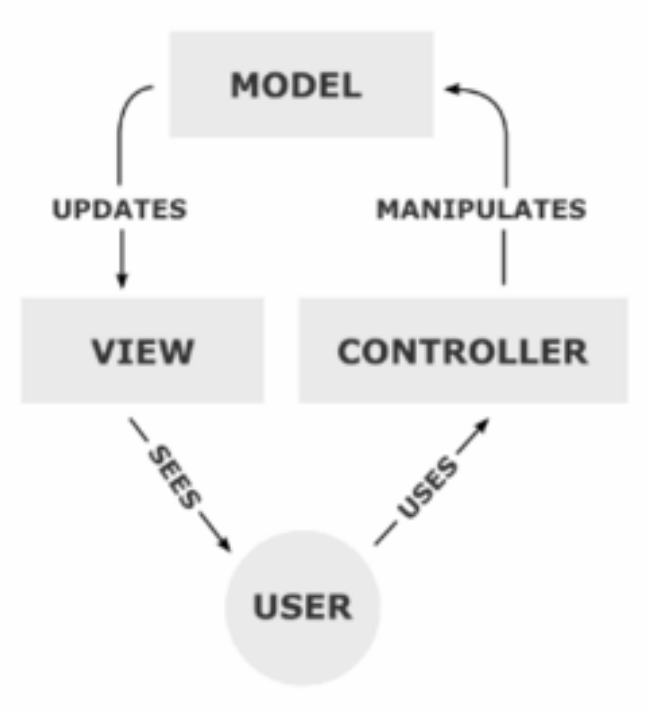
```xml
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE recipe>
<recipe name="хлеб" preptime="5min" cooktime="180min">
   <title>
      Простой хлеб
   </title>
   <composition>
      <ingredient amount="3" unit="стакан">Мука</ingredient>
      <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
      <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
   </composition>
   <instructions>
     <step>
        Смешать все ингредиенты и тщательно замесить.
     </step>
     <step>
        Закрыть тканью и оставить на один час в тёплом помещении.
     </step>
```

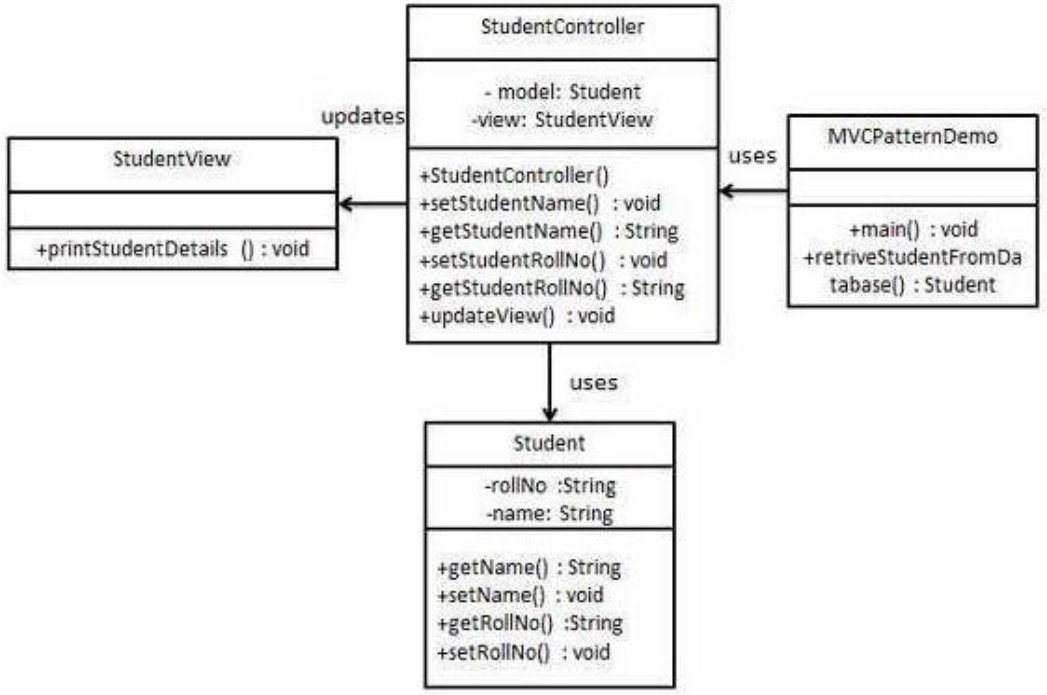
```xml
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Example text.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

ООП: фреймворк

Model-view-controller (MVC, «модель-представление-контроллер», «модель-вид-контроллер») — схема использования нескольких шаблонов проектирования, с помощью которых модель приложения, *пользовательский интерфейс* и взаимодействие с пользователем разделены на три отдельных компонента таким образом, чтобы модификация одного из компонентов оказывала минимальное воздействие на остальные.

Данная схема проектирования часто используется для построения *архитектурного каркаса*, когда переходят от теории к реализации в конкретной предметной области





Model-View-Controller (Википедия)

Понятие аспекта и сложности: оценка

Оценка (estimation) - это процесс поиска оценки или **приближения**, которое представляет собой значение, которое можно использовать для некоторых целей, даже если входные данные могут быть **неполными**, **неопределенными** или **нестабильными**.

Оценка определяет, сколько денег, усилий, ресурсов и времени потребуется для создания конкретной системы или продукта.

Исходными данными для оценки могут служить:

- **прошлые** данные / прошлый опыт
- **доступные** документы / знания
- **предположения**
- выявленные **риски**

Четыре основных шага в оценке программного проекта:

- оцените **размер продукта** разработки.
- оцените **усилия** в человеко-месяцах или человеко-часах.
- оценивайте **расписание** в календарных месяцах.
- оценить стоимость проекта в согласованной валюте.

Оценка: функциональные точки

Функциональная точка (functional point, FP)

> является **единицей измерения**, выражающей количество **бизнес-функциональности**, которую информационная система (как продукт) обеспечивает пользователю. FP измеряют размер программного обеспечения.

COSMIC - ISO/IEC 19761:2011 Разработка программного обеспечения. Метод измерения функционального размера.

FiSMA - ISO/IEC 29881:2008 Информационные технологии - Программное обеспечение и системная инженерия - Метод измерения функционального размера FiSMA 1.1.

IFPUG - ISO/IEC 20926:2009 Разработка программного обеспечения и систем - Измерение программного обеспечения - Метод измерения функционального размера IFPUG.

Mark-II - ISO/IEC 20968:2002 Разработка программного обеспечения - Анализ функциональных точек Ml II - Руководство по методам подсчета.

NESMA - ISO/IEC 24570:2005 Разработка программного обеспечения - Метод измерения размера функции NESMA, версия 2.1 - Определения и рекомендации по подсчету для применения анализа функциональных точек.

Оценка: функциональные точки IFPUG

Концепция функциональных точек была введена Аланом Альбрехтом из IBM в 1979 году. В 1984 году Альбрехт усовершенствовал этот метод. Первые рекомендации по функциональным точкам были опубликованы в 1984 году.

Международная группа пользователей функциональных точек (IFPUG) - это всемирная организация пользователей программного обеспечения для анализа функциональных точек в США.

International Function Point Users Group Users Group (IFPUG) является некоммерческой, саморегулируемой организацией, основанной в 1986 году. IFPUG владеет Function Point Analysis (FPA), как определено в стандарте ИСО 20296:2009, который определяет определения, правила и меры для применения метода измерения функционального размера (FSM) IFPUG.

IFPUG поддерживает Руководство по методам подсчета функциональных точек (CPM). CPM 2.0 был выпущен в 1987 году, и с тех пор было сделано несколько итераций. Версия 4.3 CPM была выпущена в 2010 году.

Оценка: функциональные точки IFPUG

Одна ф.т. приблизительно равна 162 операторам языка С или **30 операторам** языка программирования Visual Basic. На реализацию проектов размером 1 ф.т. требуется **1 день**, и они практически всегда заканчиваются успешно. Это, как правило, создание небольших утилит для временных нужд.

Объем в **10 ф.т.** - типичный объем **небольших приложений** и **дополнений**, вносимых в готовые системы. Такие проекты требуют до **1 месяца** работ и тоже всегда успешны.

Объем в **100 ф.т.** близок к **пределам возможностей** программиста-одиночки. Проект доводится до конца за **6 месяцев** в 85% случаев.

Объем проекта в **1000 ф.т.** характерен для большинства сегодняшних коммерческих настольных и небольших клиент-серверных приложений. Приличную долю в общем объеме начинает занимать **документация**. Для разработки необходима **группа** примерно из **10 программистов, проектировщиков, специалистов по управлению качеством** и около **года работы**. Проваливается 15% подобных групповых проектов и 65% попыток программистов-одиночек. В 20% случаев в срок уложиться не удается. Перерасход средств отмечается в 25% проектов.

Для проекта объемом в **10 000 ф.т**. требуется около **100 разработчиков.**

Понятие аспекта и сложности: IFPUG

Function points are now a standard **sizing** metric

Function points are now a standard **productivity** metric

Function points are now a powerful **quality** metric

Function points are now a powerful **schedule** metric

Function points are now a powerful **staffing** metric

Function points are now used in software **litigation**

Function points are now used for **outsource contracts**

Function points can be used for **cost analysis** (with care)

Function points can be used for **value analysis** (with care)

Понятие аспекта и сложности: IFPUG

Function points used for **portfolio analysis**

Function points used for **backlog analysis**

Function points used for **risk analysis**

Function points used for **real-time requirements changes**

Function points used for **software usage studies**

Function points used for **delivery analysis**

Function points used for **COTS analysis**

Function points used for **occupation group analysis**

Function points used for **maintenance analysis**

Понятие сложности: эволюция индустрии

Moving from software development to software delivery

Development rates < 12.5 function points per staff month

Delivery rates > 100 function points per staff month

Delivery issues: quality, security, band width

Delivery methods:

- Service Oriented Architecture (SOA)
- Software as a Service (SaaS)
- DevOps and other new methodologies
- Mashups from reusable components

Понятие сложности: эволюция индустрии

Measure **delivered features** as well as development

Measure and **catalog** reusable features

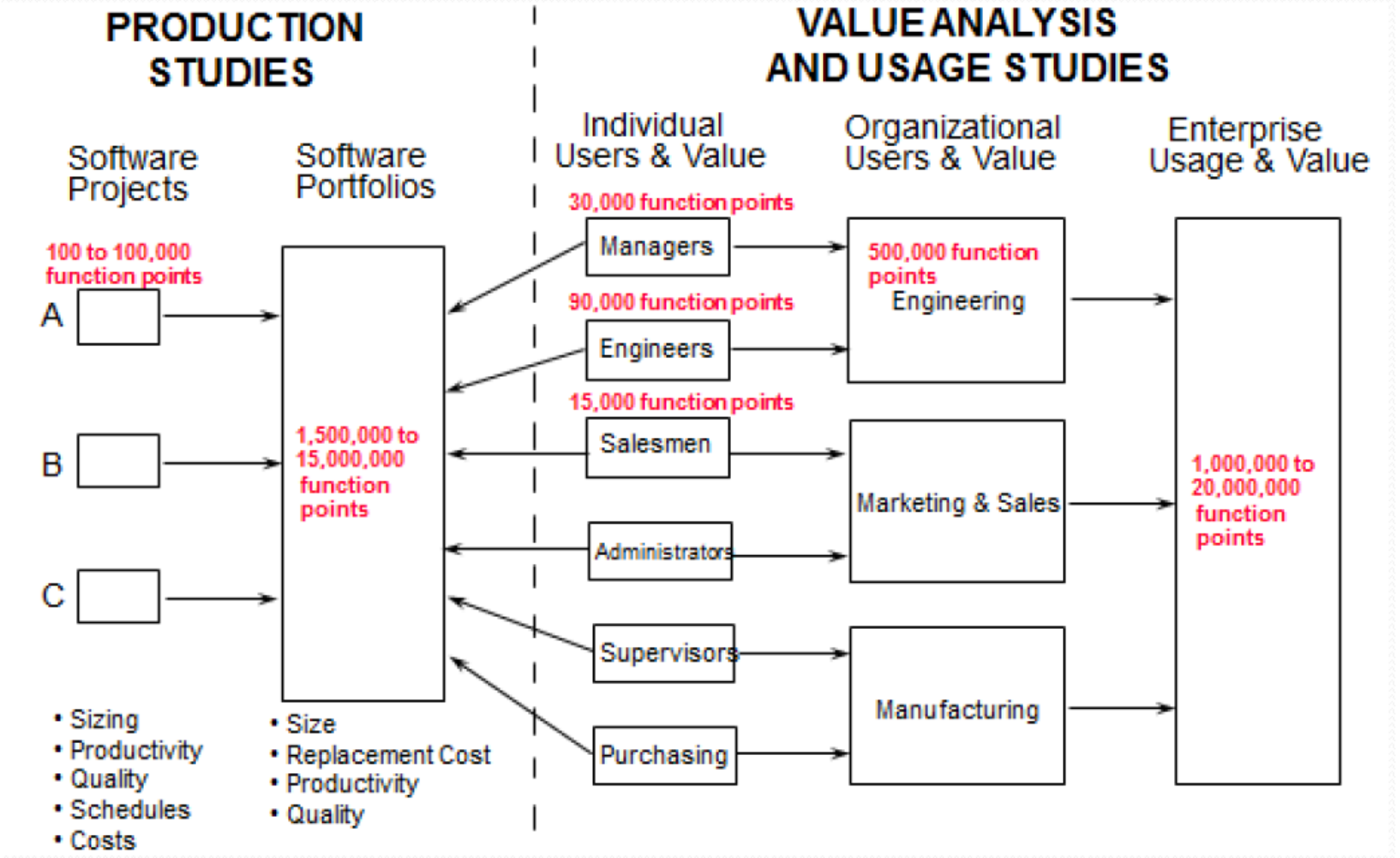
Measure **deployment**, **installation**, and **usage**

Measure **quality** and **security**

New measurements needed:

- Sizes of reusable components
- Quality, security of reusable components
- Sizes of delivered applications
- Deployment and usage of delivered applications

Понятие сложности: эволюция индустрии



Понятие сложности: измерение приложений

Applications	Approximate Size in Function Points
Star Wars Missile Defense	300,000
ERP (SAP, Oracle, etc.)	250,000
Microsoft Windows 10	198,000
Microsoft Office	98,000
Airline Reservations	50,000
NASA space shuttle	25,000

Понятие сложности: структура затрат

Application = 1,000 Function points

Activity	Staff	Effort (months)	Schedule (months)
Requirements	2	5	2.5
Design	2	6	3.0
Coding	5	40	8.0
Testing	7	25	7.0
Documentation	1	7	7.0
Management	1	25	24.0
OVERALL	4.5	108	24.0

Понятие сложности: измерение проекта

Function points ^ 0.40 power = calendar months in schedule

Function points ^ 1.15 power = pages of paper documents

Function points ^ 1.20 power = number of test cases

Function points ^ 1.25 power = software defect potential

Function points / 150 = development technical staff

Function points / 1,500 = maintenance technical staff

Понятие сложности: измерение задач и рисков

ONE YEAR FOR 10,000 FUNCTION POINTS

Starting size = 10,000 function points

Task	Size (FP)	Staff	Effort	Cost
Large updates	500	3.33	33.33	$250,000
Small "				
V. Small "	1			
Bug fixes	17			
TOTAL	25			

RESULTS FOR 10,000 FUNCTION POINTS

No formal risk analysis = 35% chance of failure

Risk analysis **after** requirements = 15% chance of failure

Risk analysis **before** requirements = 5% chance of failure

- Early sizing and formal risk analysis raise the chances of success for large software projects

Понятие сложности: измерение цифровизации

FUNCTION POINTS AND USAGE ANALYSIS

Occupation	Function Points Available	Software Packages	Daily usage (hours)
1 Military planners	5,000,000	30	7.5
2 Physicians	3,000,000	20	3.0
3 FBI Agents	1,500,000	15	3.5
4 Attorneys	325,000	10	4.0
5 Air-traffic controllers	315,000	3	8.5
6 Accountants	175,000	10	5.0
7 Pharmacists	150,000	6	4.0
8 Electrical engineers	100,000	20	5.5
9 Software engineers	50,000	20	7.0
10 Project managers	35,000	10	1.5

Функциональные точки