

Федеральное государственное автономное образовательное
учреждение высшего образования
«Санкт-Петербургский государственный университет
аэрокосмического приборостроения»

Методология программной инженерии

Методические указания к выполнению курсового проекта

Составители:

к.т.н., стар. преп. П.А. Охтилев, асс. А.Э. Зянчурин

Рецензент:

д.т.н., проф. М.Ю. Охтилев

Санкт-Петербург – 2022

Содержание

Курсовой проект. Техническое проектирование.....	3
Цель работы.....	3
Задание на курсовой проект	3
Общие рекомендации по выполнению курсового проекта	3
Основные положения технического проектирования	3
Раздел "Введение"	8
Раздел "Назначение и область применения"	13
Раздел "Техническая характеристика"	17
Раздел "Описание и обоснование выбранной конструкции"	29
Раздел "Описание организации работ с применением разрабатываемого изделия"	47
Оформление приложений к пояснительной записке технического проекта	65
Рекомендации по оформлению пояснительной записке к техническому проекту	68
Подготовка к защите курсового проекта (курсовой работы)	69
Список литературы.....	70
Список сокращений	72
Список иллюстраций	73
Варианты заданий для выполнения лабораторных работ.....	74

Курсовой проект. Техническое проектирование

Цель работы

Целью работы является ознакомление с требованиями к выполнению технического проекта на создание изделий – автоматизированные системы и программное обеспечение, а также получение навыков технического проектирования и оформления отчетных материалов с его результатами в соответствии с отечественными стандартами и международными практиками с учетом современных требований промышленного производства программного обеспечения.

Задание на курсовой проект

Описание назначения и области применения автоматизированной системы (программного обеспечения). Обоснование выбранных технических решений. Формализация архитектурных решений. Формализация и анализ технического задания. Описание перспективы применения технических решений. Формирование показателей качества автоматизированной системы (программного обеспечения) и их проверка (верификация) в соответствии с условиями применения.

Общие рекомендации по выполнению курсового проекта

Основные положения технического проектирования

Место и роль технического проекта в соответствии с ГОСТ 2.103 и ГОСТ 19.404-79

Промышленность является одной из самых важных отраслей экономики и народного хозяйства, которая оказывает влияние на материальную сферу общества. Промышленность – совокупность предприятий, ориентированных на создание изделий (продукции), обработку различного вида сырья, производство энергии, орудий труда и на многое другое.

Процесс создания нового изделия включает в себя, прежде всего, анализ потребности общества в данном изделии. Если по результатам анализа установлено, что потребность в изделии имеется, то в установленном порядке заключается контракт (договор) между заказчиком и исполнителем, а также разрабатывается ТЗ или ТТЗ (тактико-техническое задание) на его создание. При разработке новых сложных изделий, в которых заказчиком является крупная корпорация или государство ТЗ может быть разработано на опытный образец изделия. В таких случаях открываются опытно-конструкторские работы, имеющие свою специфику выполнения.

Опытно-конструкторская работа (ОКР) – комплекс работ по разработке конструкторской и технологической документации на опытный образец изделия, изготовлению и испытаниям опытного образца (опытной партии), выполняемых для создания (модернизации) продукции. ОКР относится к разработке как серийной, так и единичной продукции [15]. ОКР, как правило, начинается с ТТЗ. При проведении ОКР в ее состав могут быть включены работы по созданию технологической документации и средств технологического оснащения для изготовления опытных образцов или установочных серий, а также единичной продукции. Опытными образцами продукции (изделий) могут является также автоматизированные системы (АС) или программное обеспечение (ПО) – в виде самостоятельных комплексов или самостоятельных составных частей АС.

Технический проект (ТП) является проектной стадией ОКР по ГОСТ 2.103, и его следует разрабатывать в соответствии с ТЗ, определяя технические решения, которые дадут полное представление о разрабатываемом изделии (АС или ПО) или его опытном образце и исходных данных для разработки рабочей конструкторской документации (РКД). Требования к ТП регламентируются ГОСТ 2.120 «Единая система конструкторской документации. Технический проект» и ГОСТ 19.404-79 «Единая система программной документации. Пояснительная записка».

В рамках проектной стадии ТП, при необходимости, проектируются и разрабатываются макеты изделий и/или их составные части. Макет – изделие, отражающее или подтверждающее совокупность свойств, важных для решения задач определенной стадии разработки или этапа жизненного цикла (ЖЦ). Документация на макет может выпускаться в виде эскизной документации в соответствии с ГОСТ 2.125. Необходимость проектирования и разработки макета определяется в ТЗ на работу.

Приемка ТП проводится приемочной комиссией заказчика ОКР. Приемочной комиссии предъявляется следующий перечень документов:

- пояснительная записка ТП;
- отчет о патентных исследованиях (если его разработка предусмотрена контрактом);
- результаты испытаний макета (если определено ТЗ).

По результатам выполнения ТП формируется окончательный вариант ТЗ на разработку, в котором определяются специальные требования к изделию (АС и/или ПО). После согласования и утверждения ТП начинается этап РКД для изготовления опытного образца изделия и проведения его испытаний.

Программная инженерия в техническом проектировании

В начале 70-х годов XX века наметился кризис в области промышленной разработки ПО, который выражался в том, что проекты на разработку ПО стали выполняться с отставанием от графика и/или с превышением финансовых расходов, а разработанный программный продукт не соответствовал предъявленным требованиям, что по характеристикам отдаляло его от первоначального замысла. В конечном итоге качество, прошедших приемку программных продуктов не устраивало заказчиков.

Аналитические исследования, выполненные в последующие годы ведущими зарубежными аналитиками, показывали не слишком обнадеживающие результаты. Известным примером являются исследования, проведенные в 1995 году компанией Standish Group, проанализировавшей работу 364 американских корпораций и итоги выполнения более 23 тысяч проектов, связанных с разработкой ПО, которые показали, что только 16,2% проектов завершились в срок, не превысили запланированный бюджет и реализовали все требуемые функции. С опозданием были завершены 52,7% проектов, расходы на их разработку превысили запланированный бюджет, а требуемые функции не были реализованы в полном объеме. Аннулированы до завершения 31,1% проектов. Для двух последних категорий проектов бюджет среднего проекта превышен на 89%, а срок выполнения на 122%. В 1998 году процентное соотношение трех перечисленных категорий лишь немного изменилось в лучшую сторону (26, 46 и 28 % соответственно) и в последующие годы улучшилось незначительно. Причинами столь низких показателей, по мнению разработчиков, являются следующие [10,14]:

- нечеткая и неполная формулировка требований к программным продуктам;
- недостаточное вовлечение пользователей в работу над проектом;
- отсутствие необходимых ресурсов;
- неудовлетворительное планирование и неэффективное управление проектом;
- частые изменения требований и спецификаций;
- новизна и несовершенство используемой технологии;
- недостаточная поддержка со стороны высшего руководства;

- недостаточно высокая квалификация разработчиков, отсутствие необходимого опыта.

Также следует отметить проблему неадекватной оценки значимости ПО заказчиком, которая характерна при изготовлении изделий, которые имеют аппаратную часть – АС. Практика показывает, что существует непонимание того факта, что основной функционал изделий таких как АС лежит именно на ПО. В результате этого непонимания заказчик с трудом идет трату ресурсов необходимых для производства ПО и, как следствие, не обеспечиваются те характеристики, которые могли бы быть в конечном изделии.

Таким образом, возникла потребность в контроле и управлении процессом разработки ПО. Как результат – произошел переход от хаотичного подхода к разработке ПО к промышленным (индустриальным) способам его создания. Было положено начало развитию инженерных методов и средств создания ПО, которые были объединены в общую дисциплину «Программная инженерия».

Программная инженерия – это дисциплина компьютерной науки и технологии, назначением которой является исследование методов, подходов и средств создания сложных программных систем с участием слаженных коллективов разработчиков различных специальностей и квалификаций. Суть методологии программной инженерии состоит в применении систематизированного, научного и предсказуемого процесса проектирования, разработки и сопровождения программных систем [10,14].

В основе программной инженерии лежит фундаментальная идея: «проектирование ПО является формальным процессом, который можно изучать и совершенствовать». Однако, как показывает опыт, использовать математические методы для формализации процесса проектирования программных систем не является тривиальной задачей.

Методы программной инженерии предназначены для организации и оптимизации технологического процесса промышленного производства ПО. Основные концепции дисциплины формализованы целым комплексом международных стандартов, охватывающих лучшие практики организации процессов ЖЦ ПО. Понятие ЖЦ отражает совокупность этапов и процессов ПО от формирования замысла до реализации и прекращения применения. Программы и данные, которые они обрабатывают, являются наиболее изменчивыми составляющими программной инженерии и подвержены изменениям в течении всего их ЖЦ.

Обобщенная модель ЖЦ ПО состоит из следующих основных этапов:

- определение потребностей заказчика;
- исследование и описание основных концепций ПО;
- проектирование и разработка ПО;

- испытания ПО;
- распространение и продажа ПО;
- эксплуатация ПО;
- сопровождение и мониторинг ПО;
- утилизация ПО.

В ТП основной упор делается на этапы «определение потребностей заказчика», «исследование и описание основных концепций ПО» и «проектирование ПО». В процессе разработки ТП выполняются работы, необходимые для обеспечения предъявляемых к изделию требований и позволяющие получить полное представление об изделии (ПО), оценить его соответствие требованиям ТЗ, технологичность конструкции, степень сложности изготовления, специфику эксплуатации и перспективы использования. Роль программной инженерии в ТП заключается в обеспечении надежности и эффективности принятых и разработанных технических решений, которые являются основой разрабатываемого изделия.

Порядок оформления пояснительной записки к техническому проекту

Курсовой проект предполагает оформление итогового отчета, в который должны быть включены материалы ТП, имеющие вид пояснительной записки в соответствии со структурой ГОСТ 2.103 и ГОСТ 19.404-79. Следующие разделы (от раздела «Введение» до раздела «Оформление приложений к пояснительной записке технического проекта») должны быть включены в пояснительную записку к ТП. Необходимо отметить, что в каждом разделе имеется набор подразделов, содержание которых должно быть раскрыто в пояснительной записке к ТП. Однако по согласованию с преподавателем подразделы могут быть изменены по причине содержательных особенностей темы курсового проекта.

Приведем порядок оформления пояснительной записки к ТП на основе общих требований к оформлению ГОСТ 2.105 «Единая система конструкторской документации. Общие требования к текстовым документам», с которым рекомендуется ознакомиться перед выполнением работы.

1. При выполнении документов автоматизированным способом рекомендуется применять шрифты, используемые средствами вычислительной техники размером 11-14 пт для основного текста, а для приложений, примечаний, таблиц, сносок и примеров – на 1-2 пт меньше.
2. При оформлении документа допускается использовать перенос в словах, кроме заголовков. Текст документа рекомендуется оформлять с использованием полуторного межстрочного интервала.

3. Расстояние от рамки формы до границ текста в начале и в конце строк – не менее 3 мм.
4. Расстояние от верхней или нижней строки текста до верхней или нижней рамки должно быть не менее 10 мм.
5. Абзацы в тексте начинают отступом, равным пяти знакам используемой гарнитуры шрифта (12,5-17 мм). Абзацный отступ должен быть одинаковым по всему тексту документа.
6. В документах следует применять научно-технические термины, обозначения и определения, установленные соответствующими стандартами, а при их отсутствии – общепринятые в научно-технической литературе.
7. Наполнение и структура ТП должна соответствовать приведенным разделам ниже.
8. Наименование в итоговом отчете курсового проекта раздел «Оформление приложений к пояснительной записке технического проекта» должен быть назван – «Приложения».
9. В каждом разделе должно быть выделено положение как отдельный пункт.
10. Таблицы, рисунки, графики и диаграммы должны быть подписаны и иметь свой уникальный номер в тексте по ГОСТ 2.105.
11. Названия разделов и пунктов не должны содержать аббревиатуры.
12. В тексте работы не должно содержать положений противоречащих ГОСТ 2.103 и ГОСТ 19.404-79.
13. По ходу текста работы должны быть приведены ссылки на литературные и иные источники информации.
14. В тексте должны быть учтены положения раздела «Рекомендации по оформлению пояснительной записки к техническому проекту».

Раздел "Введение"

Актуальность темы выполняемой работы

Актуальность темы разработки АС или ПО заключается в наличии потребности решения некоторой задачи в предметной области за счет автоматизации. В явном виде необходимо подчеркнуть факторы, которые связаны с решаемой задачей и их значимость.

Характеристика актуальности темы. Актуальность темы выполняемой работы характеризуется:

- аспектами темы, которые требуют решения;
- описанием факторов, которые связаны с проблематикой предметной области;

- обоснованностью практического эффекта от внедрения изделия (АС или ПО);
- прогнозом желательных эффектов от автоматизации тех или иных процессов предметной области.

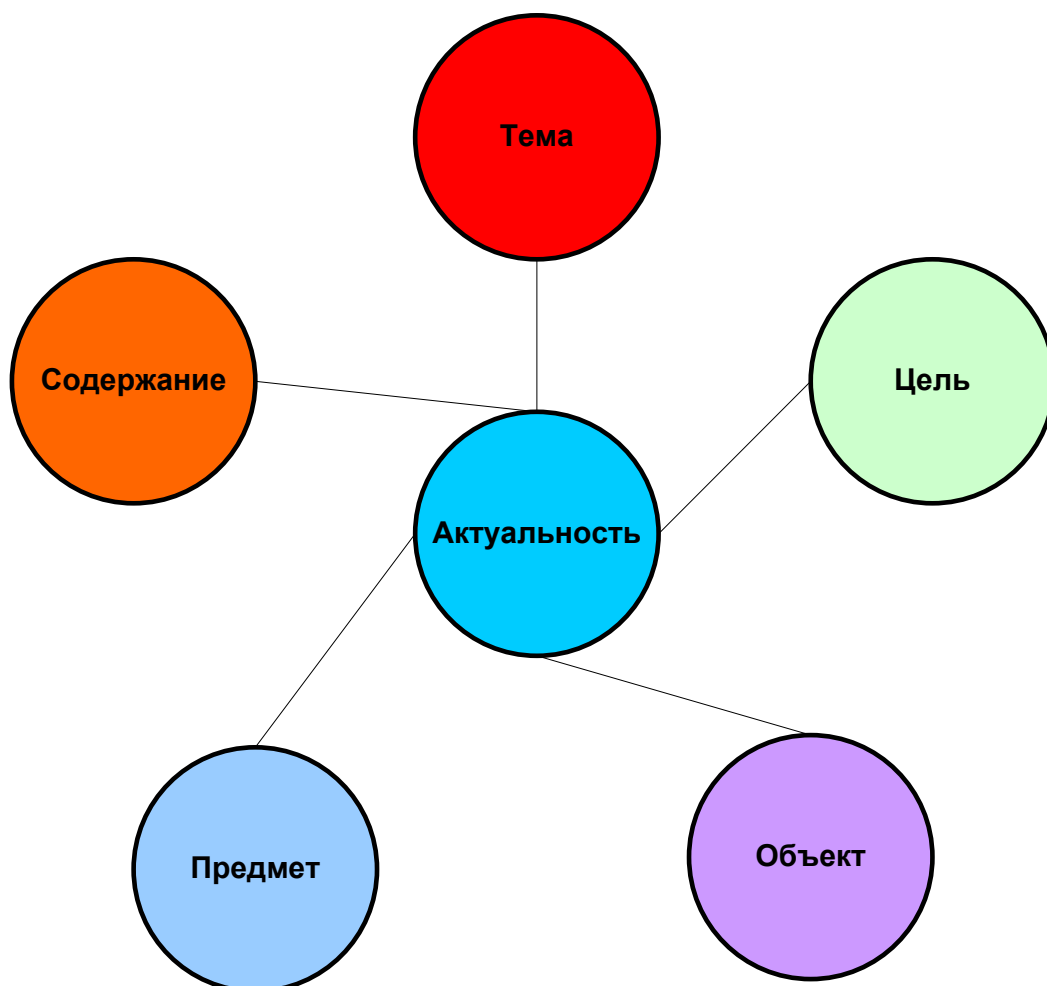


Рис. 1 Место актуальности в структуре работы

План написания раздела «актуальности темы». Не существует строгого алгоритма описания раздела «актуальности», но формализовать некоторые положения необходимо. Описать актуальность темы рекомендуется в соответствии со следующими пунктами.

1. Анализ состояния предметной области: приведение фактов и тенденций, законодательных и правовых актов, регламентов, стандартов, статистики, последствий, исследований и т.д.
2. Поиск и описание противоречивых ситуаций, выявление отсутствия систематизации процессов, обозначение нежелательных эффектов.
3. Изложение проблемы автоматизации предметной области.

4. Вывод об актуальности и описание причины выбора данной темы работы.

По ходу написания необходимо приводить аргументы для обоснования того или иного положения. Примерами аргументов могут выступить:

- специфика условий и факторов предметной области;
- расхождение теории и практики в предметной области;
- привлечение внимания к проблеме в предметной области;
- правительственные постановления, относящиеся к предметной области;
- возникновение спроса на рекомендации по внедрению изделия в рамках предметной области;
- и т.д.

Проверка корректности описания актуальности темы. Корректность описания актуальности темы можно проверить по следующим положениям, которые приведены в таблице ниже.

№	Критерий проверки	Возможное упущение
1.	Размещение в разделе «Введение»	Неверное размещение в тексте
2.	Объем актуальности от 15 до 25 предложений (максимум 1,5 страницы А4)	Отклонение от общепринятых норм объема
3.	Достоверность и современность сведений	Аргументация и заинтересованность автора в теме работы
4.	Согласованность с основной частью пояснительной записки ТП	Отсутствие согласованности с основной частью пояснительной записки ТП
5.	Уникальность	Низкая уникальность авторской проработки

Цель технического проектирования

Цель определяет предполагаемый результат, который проявляется в виде желаемого эффекта (улучшение показателей) от внедрения автоматизации в предметной области.

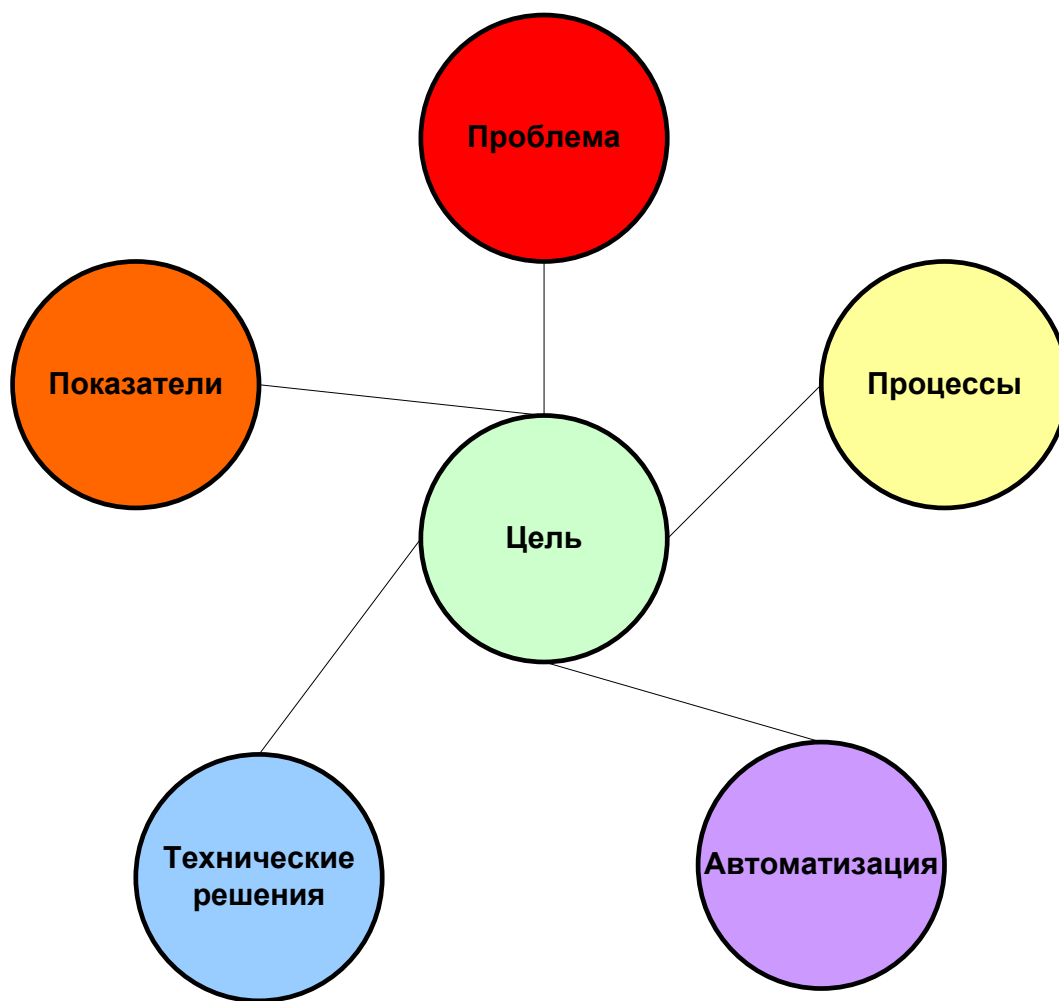


Рис. 2 Место цели в структуре работы

Цель в ТП формулируется в виде одного предложения, в котором должен быть обозначен желаемый результат. Цель автоматизации в общем виде можно охарактеризовать как повышение эффективности процесса или объекта в рамках тех или иных условий применения. Приведем несколько типовых обобщенных примеров возможных целей ТП:

- улучшение и повышение продуктивности бизнес-процессов;
- уменьшение затрат;
- улучшение операционной бизнес-деятельности;
- повышение эффективности управления;
- уменьшение рисков;
- повышение эффективности предприятия;
- повышение продуктивности работы пользователей (операторов);
- повышение возможности и прозрачности взаимодействия;
- уменьшение стоимости поддержки ЖЦ изделия;
- улучшение характеристик безопасности;
- повышение управляемости.

Объект и предмет технического проектирования

Объектом ТП являются технические решения, которые обеспечивают определенный уровень автоматизации процесса(-ов) в предметной области.

Предметом ТП является процесс разработки технических решений, который взаимосвязан с проблематикой предметной области на всех стадиях ЖЦ изделия (АС или ПО).

Объект и предмет должны быть согласованы с целью технического проектирования.

Новизна технических решений

Одной из характеристик новизны технических решений является их уникальность относительно от аналогов. Уникальность должна быть формализована с позиции преимуществ разработанных технических решений, которые могут быть посчитаны математически.

Обоснованность и достоверность технических решений

Обоснованность и достоверность технических решений определяется предыдущим опытом внедрения изделия в конкретной области применения и его эффективностью функционирования в данных условиях.

Структура и объем работы

В заключении данного раздела указывается объем работы, описывается структура пояснительной записки с количеством разделов, приложений, списка сокращений, иллюстраций, таблиц и графиков, а также литературных источников.

Раздел "Назначение и область применения"

Системный анализ предметной области и методов решения задачи автоматизации

Системный анализ предметной области предполагает ее содержательное описание и установление взаимосвязи объектов и факторов, взаимодействие которых является порождением нежелательных эффектов, которые предполагается устранить за счет внедрения изделия (АС или ПО).

Для проведения системного анализа привлекается специализированный набор источников информации, которые дадут понимание контекста и методов решения задач, использованных ранее. Результатом системного анализа должен быть детальный набор выводов о проблематике и несовершенстве методов, с помощью которых осуществлялось решение.

Любое предприятие и любую предметную область можно представить в виде системы, обладающей некоторой структурой и поведением. Если говорить о предприятии, то оно имеет организационную структуру, включающую в себя набор подразделений, связанных определенным образом друг с другом. Что касается предметной области, то она состоит из множества объектов и взаимосвязей между ними. Как объекты предметной области, так и подразделения предприятия являются участниками или непосредственно исполнителями некоторых процессов. Объекты могут сами обладать некоторым поведением либо могут изменяться под воздействием других объектов в ходе того или иного процесса.

Применительно к процессу автоматизации деятельности предприятия его подразделения могут осуществлять какие-либо бизнес-процессы или действия в рамках процессов, либо могут играть роль вспомогательных единиц, предоставляющих необходимые для осуществления рассматриваемого процесса ресурсы, либо они могут являться потребителями результатов выполнения процесса. Такие же роли могут играть и внешние по отношению к данному предприятию субъекты (другие организации или физические лица) и объекты (например, информационные системы). Часть осуществляемых процессов может быть автоматизирована. В этих случаях можно говорить о том, что данный процесс выполняется данным подразделением с использованием некоторой информационной системы или программно-инструментального средства [14].

Выделим основные этапы системного анализа:

- содержательная постановка задачи;
- построение модели изучаемой системы;
- поиск решения задачи с помощью модели;
- проверка (верификация) решения с помощью модели;

- адаптация решения под среду;
- реализация решения.

Содержательная постановка задачи является основанием для системного анализа предметной области и методов решения задачи автоматизации. На этом этапе необходимо формализовать конечный результат, который, в свою очередь, станет основанием для разработки модели изучаемой системы.

Краткая характеристика условий и области применения изделия

Формирование понимания условий и области применения является важной темой с точки зрения идентификации всех возможных источников требований, значимых в рамках выполнения работ над изделием (АС или ПО). При грамотном подходе к выявлению значимых и существенных условий и факторов предметной области появляется возможность конструктивно критиковать требования ТЗ и, тем самым, устранить ошибки заложенные в них. Данная тема фокусируется на следующем перечне вопросов.

- Кто является заинтересованными лицами (интересантами)?
- Что из себя представляет среда?
- Каких целей необходимо достичь?

Интересанты. Понимание того, кто является интересантом и какие у интересантов потребности, а также в каких условиях эти потребности сформированы и чем характеризуются является важным фактором успешного решения. В этом процессе поиска полезны следующие вопросы [9]:

- кто является пользователем (оператором) изделия (АС или ПО)?
- на кого и на что окажет влияние результаты функционирования АС или ПО?
- какие процессы (бизнес-процессы) являются существенными для рассмотрения?
- кто будет осуществлять сопровождение АС или ПО;
- кто и как будет оценивать и принимать изделие (АС или ПО), когда оно будет поставлено и развернуто (установлено)?
- какой набор признаков присущ каждому объекту в предметной области?
- обеспечена ли полнота информации в рассуждениях? Все ли учтено?

Интересант – некое лицо, имеющее возможность влиять на работу над изделием (АС или ПО), поэтому их выявление является крайне важной задачей. Интересанты в проекте выступают в тех или иных ролях. Приведем некоторые типичные примеры ролей для более простой идентификации интересантов [1, 15].

1. **Пользователи (Users):** группа, охватывающая тех людей, кто будет непосредственно использовать программное обеспечение; пользователи могут описать задачи, которые они решают (планируют решать) с использованием программной системы, а также ожидания по отношению к атрибутам качества, отображаемые в пользовательских требованиях.
2. **Заказчики (Customers):** те, кто отвечают за заказ программного обеспечения или, в общем случае, являются целевой аудиторией на рынке программного обеспечения (образуют целевой рынок ПО).
3. **Аналитики (Market analysts):** продукты массового рынка программного обеспечения (как и других массовых рынков, например, бытовой техники) не обладают «заказчиками» в понимании персонификации тех, кто “заказывает разработку. В то же самое время, лица, отвечающие за маркетинг, нуждаются в идентификации потребностей и обращению к тем, кто может играть роль квалифицированных «представителей» потребителей.
4. **Регуляторы (Regulators):** многие области применения (“домены”) являются регулируемыми, например, телекоммуникации или банковский сектор. Программное обеспечение для ряда целевых рынков (в первую очередь, корпоративного сектора) требует соответствия руководящим документам и прохождения процедур, определяемых уполномоченными органами.
5. **Инженеры** по программному обеспечению, инженеры-программисты (Software Enginner): лица, обладающие обоснованным интересом к разработке программного обеспечения, например, повторному использованию тех или иных компонент, библиотек, средств и инструментов. Именно инженеры ответственны за техническую оценку путей решения поставленных задач и последующую реализацию требований заказчиков.

Среда. Среда является внешним окружением по отношению к изделию. Для характеристики взаимодействия изделия и среды существенны следующие два его аспекта:

- направленность этого взаимодействия («изделие – > среда» или «среда – > изделие»);
- тип взаимодействия (информационное, вещественное, энергетическое).

В рамках данных методических указаний ограничимся только информационным взаимодействием, так как вещественное и энергетическое играют посредственную роль в рассматриваемым типам изделий, а именно АС или ПО.

В среде информационных технологий и систем характеризующейся сложностью информационных процессов (бизнес-процессов), прежде чем пытаться определить конкретные проблемы, достойные решения, необходимо понять некоторые моменты самих информационных процессов (бизнес-процессов). Данная среда состоит не просто из одного-двух пользователей и пользовательского интерфейса ПО, а из организаций, предприятий, департаментов, глобальных и локальных корпоративных сетей, заказчиков, пользователей, готовой продукции, систем организации и управления и т.д.

Кроме того, даже при создании отдельного приложения необходимо помнить о многогранности среды, в которой оно будет функционировать. Таким образом, полезно ответить на следующие вопросы [9]:

- для каких нужд создается изделие?
- где будет осуществляться размещение (развертывание) изделия?
- в какие моменты времени информационные процессы (бизнес-процессы) осуществляются в ручном режиме?
- следует ли рассматривать возможность реструктуризации предприятия?

Назначение изделия, постановка задачи автоматизации и цифровизации предметной области

В назначении приводят вид автоматизируемой деятельности с ее кратким описанием, а также указывают показатели назначения изделия (АС или ПО).

Данный раздел также предназначен для описания функций изделия (АС или ПО) в обобщенном виде и что они обеспечивают. В данном разделе может быть уточнено назначение описанное в ТЗ.

Постановка задачи автоматизации и цифровизации предметной области формулируется относительно цели технического проектирования и целей определенных (уточненных) в теме «Краткая характеристика условий и области применения изделия». Цель предполагает некое улучшение показателей, которыми она характеризуется. В качестве обязательного пункта постановка задачи предполагает описание условий, в которых она решается.

Пример постановки задачи

По результатам проведенного анализа установлено, что для повышения эффективности предприятия целесообразно внедрение АС учета и формирования заказов.

Внедрение систем типа АС учета и формирования заказов показывает на практике повышение оперативности решения логистических задач в среднем на 12%. Таким образом, представляется актуальным решение следующих задач.

1. Анализ требований ТЗ на АС учета и формирования заказов...
2. Разработка моделей...
3. Разработка архитектуры...
4. Сравнительное исследование эффективности применения АС учета и формирования заказов с аналогами...
5. ...

Раздел "Техническая характеристика"

Анализ требований технического задания

Следует понимать, что процесс работы с требованиями не является дискретным, а постоянно воспроизводимым на всех этапах ЖЦ изделия (АС или ПО). Процесс идентификации и систематизации требований как открывает начало работ над будущим изделием, так и является их завершающей фазой. Анализ требований ТЗ в рамках стадии ТП предполагает взаимосвязанное выполнение процедур, которые включают решение следующих задач:

- обнаружение и разрешение конфликтов между требованиями;
- определение границ задачи, решаемой создаваемым изделием;
- детализация требований ТЗ для формирования программных требований;
- анализ осуществимости требований;
- классификация детализированных требований;
- приоритизация требований.

Результатом выполнения данных процедур является однозначно интерпретируемые требования, выполнение которых проверяемо, а стоимость и ресурсозатраты – предсказуемы.

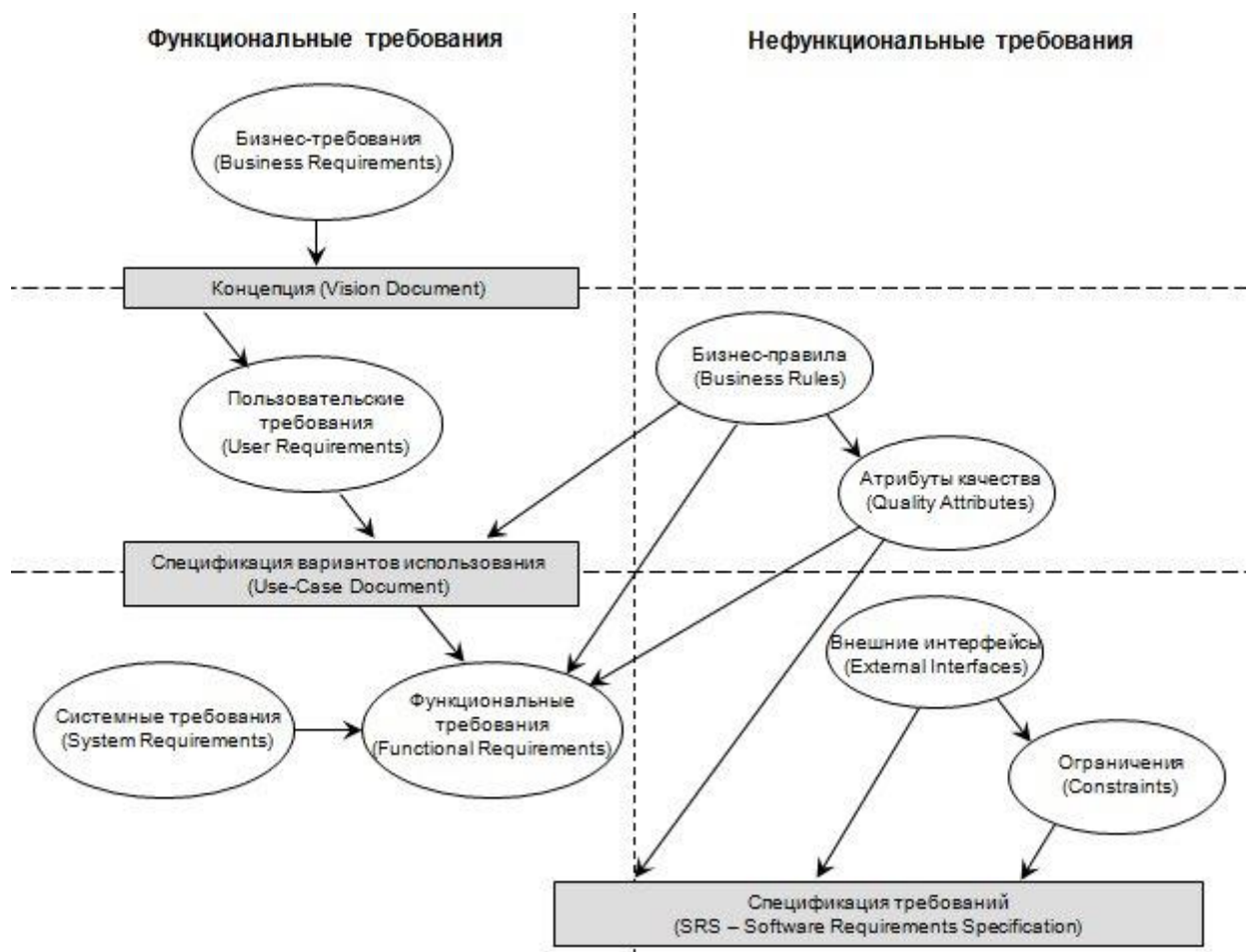


Рис. 3 Уровни требований по Вигерсу

Также, анализ требований еще является важным этапом для того, чтобы перейти к вопросам концептуального моделирования.

Описание вида автоматизируемой деятельности. Критерии и характеристики качества изделия. Предложения по совершенствованию

В данной теме на основе описания вида автоматизируемой деятельности необходимо установить взаимосвязь между целями и критериями качества и выдать рекомендации по совершенствованию требований к изделию, относительно представленных в ТЗ.

Сферы автоматизируемой деятельности позволяют дать еще один фокус рассмотрения будущего изделия и уточнить класс решаемых задач. Сферы автоматизируемой деятельности можно разделить на следующие:

- 1) автоматизированные системы управления (ОАСУ, АСУП, АСУ ТП, АСУ ГПС и др.);

- 2) системы автоматизированного проектирования (САПР);
- 3) автоматизированные системы научных исследований (АСНИ);
- 4) АС обработки и передачи информации (АСОИ);
- 5) автоматизированные системы технологической подготовки производства (АСТПП);
- 6) автоматизированные системы контроля и испытаний (АСК);
- 7) системы, автоматизирующие сочетания различных видов деятельности.

Таким образом, первый шаг это определить вид деятельности и дать ее описание в произвольном формате (таблицы, графики, иллюстрации и т.д.). В результате должен получиться детальный ответ на вопрос что происходит (делается) в рассматриваемой предметной области без наличия изделия.

Затем, полученное описание необходимо сравнить с целями, формализованными в предыдущем разделе. Цели (напомним, что это цепочка «Цель» – «Решаемая проблема» – «Что необходимо реализовать») при сравнении с описанием автоматизируемой деятельности дадут список функций будущего изделия, через которые возможно определить его характеристики и критерии качества.

Для получения характеристик и критериев качества будущего изделия предлагается рассмотреть пример на основе таблицы №1.

Таблица №1. Функции, критерии и характеристики изделия

Функция	Характеристика	Критерий
Генерация документа	1. Скорость выполнения	Сохранение корректности данных
	2. Простота использования	

В заключении темы необходимо сформулировать предложения по совершенствованию замысла изделия относительно описанного в ТЗ. Оформить предложения возможно в произвольном формате.

Концепция и принципы построения изделия. Требования технического задания и их выполнение

Как известно, концепция это система взглядов на что-либо, которые связаны между собой по смысловой нагрузке (семантически). В тоже время концепция это еще и некая трактовка какого-либо процесса, объекта или явления. Ведущий замысел также относится к понятию «концепция». Таким

образом, можно сделать вывод, что описание концепции, например, изделия (АС или ПО) не предполагает следование строгим правилам. В рамках данных методических указаний будет предложена одна из возможных точек зрения и ряд обязательных тем, что никак не ограничивает введение дополнительных элементов рассмотрения этой широкой темы.

В рамках ТП при описании концепции предлагается уделить внимание следующим темам:

- детализация и структуризация целей разрабатываемого изделия;
- определение класса изделия как информационной системы;
- разработка контекстной диаграммы;
- анализ соответствия концептуального описания и требований ТЗ;
- принципы построения изделия.

Детализация и структуризация целей разрабатываемого изделия. В некоторых компаниях в рамках этой темы разрабатывают отдельные документы, где прописывается львиная доля функционала. Данный факт говорит о том, что банальные («стать компанией номер один») и нечетко сформулированные («обеспечить высокий уровень сервиса») цели нельзя считать ни полезными, ни поддающимися верификации (проверке). Необходимо помнить, что цели дают ответ на вопрос ЧТО, а не на вопрос КАК.

Приведем пример описания целей в таблице №2.

Таблица №2. Пример описания целей разрабатываемого изделия

Цель	Решаемая проблема	Что необходимо реализовать
Повысить оперативность выполнения процесса X на Y % процентов	Сокращение затрат на N % за выделенный временной интервал	Автоматизировать в процессе X выполнение операций S1, S3 и т.д.
...

Определение класса изделия как информационной системы. АС или ПО возможно рассмотреть как информационную систему (ИС). Из определения ИС – программно-аппаратная система, предназначенная для автоматизации целенаправленной деятельности конечных пользователей (операторов) обеспечивающую, в соответствии с заложенной в неё логикой обработки, возможность получения, модификации и хранения информации [11].

Если точно определить класс ИС, то появится возможность сделать выводы о будущих ее задачах и функционале по крайней мере в обобщенном виде и поставить в соответствие определенные ранее цели. Рассмотрим наиболее важные признаки классификации.

1. Классификация по масштабу.

Однопользовательские ИС, как это ясно из названия, предназначены для использования на одном рабочем месте. В настоящее время на мировом и отечественном рынке представлено множество решений, предназначенных для автоматизации деятельности отдельно взятого пользователя. Как правило, это – решения, ориентированные на специалиста в той или иной области, будь то составление спецификаций для сборки изделий из комплектующих, планирование ремонтов оборудования, учёт расходов и доходов частного предпринимателя оптовой торговли, либо составление расписаний занятий в деканате.

В настоящее время альтернативу таким узкоспециализированным системам составили табличные процессоры, не имеющие проблемной специализации, в первую очередь – MS Excel. Системы этого класса трудно отнести к классу ИС, но зачастую они позволяют непрограммирующему специалисту создать и, что очень важно, самостоятельно развивать собственные решения, заменяющие, а местами и перекрывающие функционал однопользовательских систем образца 90-х годов.

В основе большинства однопользовательских систем лежит стандарт X-Base (Clipper, FoxPro, dBase). Широко используются также решения на базе систем Paradox, Clarion, MS Access. Каждая из перечисленных конкурирующих систем обладает собственной высокоуровневой инструментальной средой, позволяющей спроектировать базу данных, логику обработки, пользовательский интерфейс, отчёты с помощью «помощников»-построителей. На рубеже тысячелетий появились также и однопользовательские решения на базе промышленных реляционных СУБД. В этом случае ПО сервера устанавливается непосредственно на рабочую станцию пользователя. Примером может служить Personal Oracle. Данные решения предъявляют значительные требования к ресурсам рабочей станции, однако несут в себе многие преимущества промышленных СУБД.

Групповые системы предназначены для автоматизации деятельности в рабочей группе (отделе, кластере, группе проекта и т.д.). В отличие от однопользовательских ИС, групповые системы, как правило, представляют специализированные клиентские решения (их часто называют автоматизированными рабочими местами, АРМ) для различных участников группы. Например, для оптовой фирмы, ИС может представлять набор таких АРМ, как «Менеджер по продажам», «Кладовщик», «Снабженец», «Директор». Для учебного планирования – «Преподаватель», «Работник

бюро планирования», «Работник учебного отдела», «Специалист по планированию на кафедре», «Работник деканата».

При создании групповых ИС в целом используются те же средства и инструментальные среды, что и при создании однопользовательских ИС. Следует, однако, отметить, что для использования в группе при выборе между системами с файловым и реляционным сервером следует отдавать предпочтение реляционному серверу, причём целесообразно использование выделенного сервера. Это может быть, например, сервер Oracle, DB2, MS SQL, Sybase, Informix.

Корпоративные ИС (КИС) предназначены для автоматизации деятельности предприятия. В англоязычной литературе понятие «КИС» неразрывно связано с понятием «ERP» (Enterprise Resource Planning). В основе ERP-систем лежит международный стандарт управления предприятием MRP-II (Manufacture Resource Planning), обеспечивающий возможность учета, анализа и планирования основных ресурсов – финансовых, человеческих, материальных. Соответственно, корпоративные ERP-системы – набор интегрированных приложений, которые комплексно, в едином информационном пространстве поддерживают все основные аспекты управленческой деятельности предприятий: планирование ресурсов (финансовых, человеческих, материальных) для производства товаров (услуг), оперативное управление выполнением планов (включая снабжение, сбыт, ведение договоров), все виды учета и анализ результатов хозяйственной деятельности.

Среди требований, предъявляемым к современным КИС:

- централизация данных в единой базе (в основе – всегда промышленная СУБД),
- близкий к реальному времени режим работы,
- сохранение общей модели управления для предприятий разных отраслей,
- поддержка территориально-распределенных структур,
- работа на широком круге аппаратно-программных платформ и СУБД.

Примеры ERP-систем – SAP R3, «Галактика», MS Navision Axapta.

2. Классификация по архитектуре.

Архитектура «Файл-сервер». Исторически первая архитектура информационных систем. Как исполняемые модули, так и данные размещаются в отдельных файлах операционной системы. Доступ к данным осуществляется путём указания пути (path) и использования файловых операций (открыть, считать, записать). Для хранения данных используется выделенный сервер (отдельный компьютер), который и является файловым

сервером. Исполняемые модули хранятся либо на рабочих станциях, либо на файловом сервере. В последнем случае упрощается процедура их администрирования, но при этом возрастают требования к надёжности сети.

Архитектура «Клиент-сервер». Клиент-сервер – это не только архитектура, это – новая парадигма, пришедшая на смену устаревшим концепциям. Суть её заключается в том, что клиент (исполняемый модуль) запрашивает те или иные сервисы в соответствии с определённым протоколом обмена данными. При этом, в отличие от ситуации с файловым сервером, нет необходимости в использовании прямых путей операционной системы: клиент их «не знает», ему «известны» лишь имя источника данных и другие специальные сведения, используемые для авторизации клиента на сервере. Сервер, который физически может находиться на том же компьютере, а может – на другом конце земного шара, обрабатывает запрос клиента и, производя соответствующие манипуляции с данными, передаёт клиенту запрашиваемую порцию данных.

В рамках направления «клиент-сервер» существуют два основных «диалекта»: «тонкий» и «толстый» клиент. В системах на основе тонкого клиента используется мощный сервер баз данных, это – высокопроизводительный компьютер и библиотека так называемых хранимых процедур, позволяющих производить вычисления, реализующие основную логику обработки данных, непосредственно на сервере. Клиентское приложение, соответственно, предъявляет невысокие требования к аппаратному обеспечению рабочей станции. Основное достоинство таких систем – относительная дешевизна клиентских станций.

Системы с толстым клиентом, напротив, реализуют основную логику обработки на клиенте, а сервер представляет собой в чистом виде сервер баз данных, обеспечивающий исполнение только стандартизованных запросов на манипуляцию с данными (как правило – чтение, запись, модификацию данных в таблицах реляционной базы данных). В системах такого класса требования к рабочей станции выше, а к серверу – ниже. Достоинство архитектуры – переносимость серверной компоненты на серверы различных производителей: все промышленные серверы баз данных реляционного типа поддерживают работу со стандартизованным языком манипулирования данными SQL, но внутренний встроенный язык обработки данных, необходимый для реализации логики обработки на сервере у каждого из серверов свой.

Трёхслойная архитектура. Базируется на дальнейшей специализации компонент архитектуры: клиент занимается только организацией интерфейса с пользователем, сервер баз данных – только стандартизованной обработкой данных. Для реализации логики обработки данных архитектура предусматривает отдельный слой – слой бизнес-логики. Этот слой может представлять собой либо выделенный сервер (сервер приложений), либо размещаться на клиенте в качестве динамической библиотеки. Данная архитектура позволила соединить достоинства тонкого и толстого клиентов:

хорошая переносимость соединяется в ней с невысокими требованиями к клиенту.

С развитием интранет-интернет технологий появилась разновидность трёхслойной архитектуры на основании использования web-технологий. В этой разновидности роль сервера приложений играет web-сервер, а в качестве клиента используется стандартный web-браузер. Достоинства – в пониженных требованиях к клиенту и в легкой встраиваемости данной архитектуры в мировые информационные сети. Основной недостаток - известные ограничения, накладываемые на интерфейс пользователя web- браузерами.

3. Классификация по характеру использования информации.

С некоторой степенью приближения все ИС можно разделить на 2 класса: информационно-поисковые и управляющие.

Конечные пользователи информационно-поисковых систем (ИПС), как правило, имеют доступ к хранимым данным только «по чтению» и используют данные системы для поиска ответов на те или иные вопросы. Доступ по модификации данных имеет администратор системы, в функции которого входит обеспечение актуальности информации, устранение ошибок.

Классические примеры ИПС – системы поиска в библиотеках, на транспорте (справки о наличии билетов). На современном этапе развития информационных технологий классические ИПС постепенно вытесняются поисковыми серверами Интернет – общего назначения и специализированными.

Альтернатива ИПС – управляющие системы автоматизируют (полностью или частично) деятельность, связанную с принятием решений. Действия конечных пользователей таких систем приводят к модификации информации, что, конечно, не исключает возможности просто получать информацию, как в ИПС.

Примеры управляющих систем – системы бухгалтерского учета, системы планирования производственных ресурсов и т.п.

4. Классификация по системе представления данных.

Среди наиболее распространённых средств и моделей представления данных следует выделить:

- «самодельные» форматы хранения данных, хранящихся в файлах (текстовых, бинарных);
- специализированные форматы хранения данных, использовавшиеся в «дореляционный» период (например, x-Base, paradox);
- языки структурированной разметки на основе формата xml;
- реляционная модель; SQL сервер;
- объектная, объектно-реляционная модель;

- документоориентированное хранилище (IBM Lotus/Domino).

5. Классификация по поддерживаемым стандартам управления и технологиям коммуникации.

Эпоха стихийной разработки АИС закончена. Современные автоматизированные информационные системы разрабатываются, исходя из сложившихся реалий автоматизированного управления бизнесом. Существует значительное количество концепций, технологий, подходов, нашедших своё эффективное применение в различных отраслях промышленности по всему миру. Некоторые из них приобрели статус международных стандартов. В спецификации АИС, разрабатываемой для массовой продажи, как правило, указывается – какие стандарты и технологии управления она поддерживает. Менее строги требования к АИС, создаваемым под заказ для конкретного предприятия. Однако и в этом случае не учитывать сложившийся в мире позитивный опыт просто неразумно. Ниже перечислены некоторые, наиболее важные, технологии и стандарты.

- MRP (Material Requirements Planning) – планирование поставок материалов, исходя из данных о комплектации производимой продукции и плана продаж.
- CRP (Capacity Requirements Planning) – планирование производственных мощностей, исходя из данных о технологии производимой продукции и прогноза спроса.
- MRPII (Manufacture Resource Planning) – планирование материальных, мощностных и финансовых ресурсов, необходимых для производства. Стандартизовано ISO.
- ERP (Enterprise Resource Planning) – финансово-ориентированное планирование ресурсов предприятия, необходимых для получения, изготовления, отгрузки и учёта заказов потребителей на основе интеграции всех отделов и подразделений компании.
- SCM (Supply Chain Management) – управление цепочками поставок. Реализация бизнес-процессов на базе внешних предприятий и торговых площадок Основано на референтной модели SCOR, стандартизованной Supply Chain Council.
- CRM (Customer Relationship Management) - управление взаимоотношениями с заказчиками. Комплекс методов и средств, нацеленный на завоевание, удовлетворение требований и сохранение платежеспособных клиентов.
- ERP II (Enterprise Resource & Relationship Processing) - управление ресурсами и взаимоотношениями предприятия. Объединяет в себе 3 вышеперечисленные технологии.
- Workflow – технология, управляющая потоком работ при помощи программного обеспечения, способного интерпретировать описание

процесса, взаимодействовать с его участниками и при необходимости вызывать соответствующие программные приложения.

- OLAP (Online Analytical Processing) – оперативный анализ данных. Технология поддержки принятия управленческих решений на основе концепции многомерных кубов информации.
- Project Management – управление проектами. Поддерживается рядом международных стандартов.
- CALS (Continuous Acquisition and Lifecycle Support) — непрерывная информационная поддержка поставок и жизненного цикла. Описывает совокупность принципов и технологий информационной поддержки жизненного цикла продукции на всех его стадиях. Объединяет в себе практически все вышеперечисленные подходы и технологии.

Столь лаконичные определения, конечно же, позволяют лишь ознакомиться с современной терминологией.

6. Классификация по степени автоматизации.

Ручные ИС характеризуются отсутствием современных технических средств переработки информации и выполнением всех операций человеком. Например, о деятельности менеджера в фирме, где отсутствуют компьютеры, можно говорить, что он работает с ручной ИС.

Автоматические ИС выполняют все операции по переработке информации без участия человека.

Автоматизированные ИС предполагают участие в процессе обработки информации и человека, и технических средств, причем главная роль отводится компьютеру. В современном толковании в термин "информационная система", как правило, вкладывается понятие автоматизируемой системы.

Разработка контекстной диаграммы. Разработка контекстной диаграммы уточняет ограничения и формализует связи разрабатываемого изделия с внешней средой. Контекстная диаграмма графически иллюстрирует границы АС или ПО и представляет собой высокий уровень абстракции, определяя конечные элементы изделия и их взаимодействие как между собой, так и с элементами внешней среды. Приведем пример части контекстной диаграммы на примере ИС для заказа химикатов.

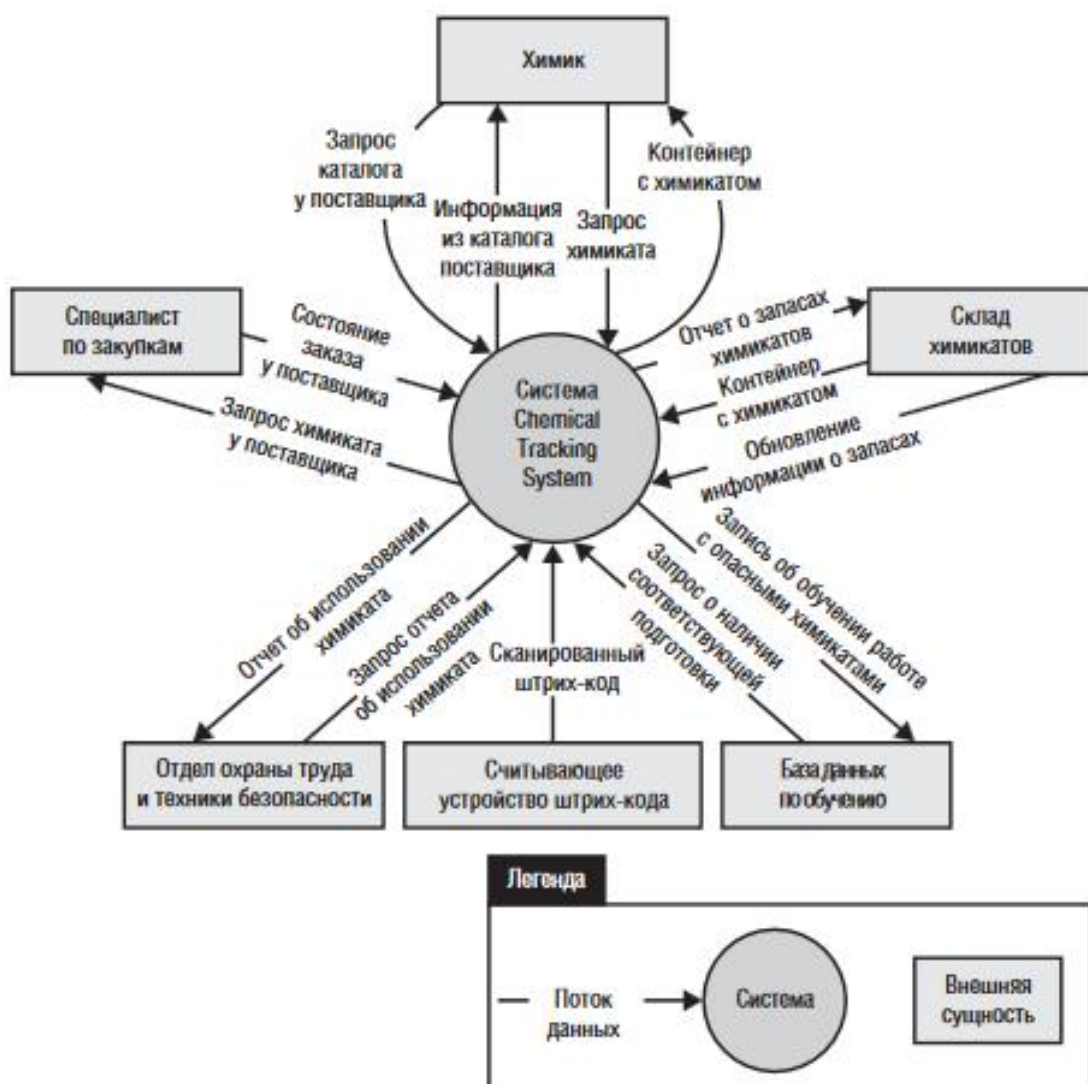


Рис. 4 Пример контекстной диаграммы ИС для заказа химикатов

Анализ соответствия концептуального описания и требований ТЗ. В данной теме необходимо раскрыть все ли система взглядов на изделие является корректной относительно требований ТЗ. Для проведения такого анализа необходимо ответить на следующие вопросы:

- под каждый ли класс требований ТЗ есть функциональный элемент изделия?
- соответствует ли класс ИС цели разработки?
- по всем ли элементам изделия даны пояснения о их назначении?
- соблюдена ли полнота требований ТЗ относительно целей и функционала изделия?
- осуществимы ли все требования ТЗ?
- возможно ли проверить требования ТЗ и согласованы ли они с целями изделия?
- какова полезность каждого требования?

Форма представления анализа соответствия концептуального описания и требований ТЗ может быть произвольной (таблицы, графики, иллюстрации и т.д.).

Принципы построения изделия. Принципы построения изделия (АС или ПО) формулируются от цели разработки и назначения изделия. Приведем примеры описания принципов построения изделия.

Пример №1.

- в основе построения АС управления предприятием N должен лежать поэтапный переход к безбумажным технологиям ведения сфер деятельности;
- основным структурным элементов АС управления предприятием N должны быть базы данных (БД), хранящих информацию об организационно-штатной структуре предприятия и его бизнес-процессов;
- ...

Пример №2.

- формирование единого информационного пространства предприятий и организаций холдинга N;
- процесс функционирования АС предполагает взаимодействие со смежными системами других предприятий;
- ...

В заключении оформления принципов построения изделия (АС или ПО) необходимо дать краткую справку по каждому формализованному принципу в произвольном формате (таблицы, графики, иллюстрации и т.д.).

***Сведения о соответствии или отклонениях от требований,
установленных техническим заданием с обоснованием отклонения***

При разработке ТЗ не всегда удастся полно и точно описать будущее изделие. Особенно это касается сложных изделий таких как АС или ПО, поэтому при более детальной проработке концептуальных и технических решений могут быть выявлены невыполнимые, противоречивые и не однозначно интерпретируемые требования, а также может быть не соблюдена полнота требований ТЗ.

При обнаружении такой ситуации необходимо зафиксировать проблемные требования и принять решение об их устранении, корректировке или полноценной замене с обоснованием данного решения и соответствующими рекомендациями. Оформить данную тему возможно в произвольном формате.

Раздел "Описание и обоснование выбранной конструкции"

Моделирование предметной области

Моделирование предполагает строгую формализацию объектов, процессов и явлений предметной области на предметно-ориентированном языке. Центральными понятиями в раскрытии данной тематики являются – модель и моделирование [13].

Модель – это упрощенное целевое отображение оригинала», а «моделирование – это замещение одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала с помощью объекта-модели». В дальнейшем изложении эти понятия детально обсуждаются и конкретизируются применительно к сфере их эксплуатации.

В конце 1997 года в качестве отраслевого стандарта был принят графический язык визуализации, спецификации, конструирования и документирования артефактов предметных областей, в которых интенсивно используется ПО. Таким языком является UML (Unified Modeling Language). UML покрывает многие аспекты моделирования предметной области однако существуют и более специализированные

В стандарте ISO 9000-2001 процесс определен как «совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы и выходы». Описание процессов необходимо с целью их дальнейшего исследования и улучшения. Моделирование процессов позволяет проанализировать не только как работает система в целом, как осуществляется взаимодействие с внешней средой, но и какие действия происходят в отдельно взятых её элементах.

Модель процесса должна давать ответы на следующие вопросы, которые позволят провести всесторонний анализ, взглянуть со всех точек зрения на процесс, детализировать его:

- какие действия необходимо выполнить для получения искомого результата;
- какая последовательность действий должна быть;
- кто/что является инициатором процесса;
- какие входные данные использует каждая процедура процесса;
- какие выходные данные формирует процедура процесса;
- какие ресурсы необходимы для выполнения для каждой процедуры;

- какая документация регламентирует выполнения процедуры;
- какие параметры характеризуют выполнение процедур и процесса в целом.

Под моделированием процесса предметной области будем понимать деятельность по выявлению особенностей реальных объектов.

Стандарт Information and Data Modeling Method (IDEF)

Одним из самых известных и широко используемых методов моделирования бизнес-процессов являются семейство стандартов IDEF. Дадим их краткое описание.

- 1) *IDEF0* – методология функционального моделирования. Используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.
- 2) *IDEF1* – методология моделирования информационных потоков внутри систем, позволяющая отображать структуру и взаимосвязи. Методология применяется для разработки информационной модели, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы.
- 3) *IDEF1X (IDEF1X Extended)* – методология разработки реляционных информационных структур. IDEF1X относится к типу методологий «сущность-связь», и как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе.
- 4) *IDEF2* – методология динамического моделирования развития систем, которая позволяет разработать динамическую модель меняющихся во времени поведения функций, информации и ресурсов системы. В настоящее время известны алгоритмы и их компьютерные реализации, позволяющие превращать набор статических диаграмм IDEF0 в динамические модели, построенные на базе «раскрашенных сетей Петри» (CPN – Color Petri Nets).

- 5) *IDEF3* – методология документирования процессов, происходящих в системе. С помощью *IDEF3* описываются сценарии и последовательность операций для каждого процесса. Функция в диаграмме *IDEF3* может быть представлена в виде отдельного процесса средствами *IDEF3*.
- 6) *IDEF4* – методология построения объектно-ориентированных систем. Средства *IDEF4* позволяют наглядно отображать структуру объектов и заложенные принципы взаимодействия, позволяя тем самым анализировать и оптимизировать сложные объектно-ориентированные системы.
- 7) *IDEF5* – методология онтологического исследования сложных систем. С помощью этой методологии системы описываются при помощи определенного словаря терминов и правил, на основе которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени. На основе этих утверждений формируются выводы о дальнейшем развитии системы и производится её оптимизация.
- 8) *IDEF6 (Design Rational Capture)* – метод рационального представления процесса проектирования информационных систем, позволяющей обосновать необходимость проектируемых моделей, выявить причинно-следственные связи и отразить это в итоговой документации системы.
- 9) *IDEF8 (User Interface Modeling) – Human – System Interaction Design Method* – метод проектирования взаимодействия пользователей с системами различной природы.
- 10) *IDEF9 (Business Constraint Discovery Method)* – метод изучения и анализа бизнес-систем в терминах «ограничений». Ограничения инициирует результат, руководят и ограничивают поведения объектов и агентов (автономных программных модулей) для выполнения целей или намерений системы.

11) *IDEF14 (Network Design Method)* – метод проектирования вычислительных сетей, позволяющих установить требования, определять сетевые компоненты, анализировать существующие сетевые конфигурации и формулировать желаемые характеристики сети.

Таким образом, стандарт IDEF для решения задач связанных с обеспечением автоматизации процессов и цифровизации объектов предметной области не позволяет учесть всех аспектов при проектировании ПО.

Язык моделирования процессов Yet Another Workflow Language

Язык Yet Another Workflow Language (YAWL) для описания процессов, позволяющий описывать операционную семантику. Данный язык поддерживает шаблоны потоков операций, системы распределения ресурсов, динамическую адаптацию моделей потоков операций, интеллектуальные функции проверки моделей потоков операций.

В YAWL определено порядка сорока специализированных шаблонов для реализации параллельных потоков управления, потоков данных, ресурсов и их распределения. В языке имеются компоненты синхронизации и параллелизма при случаях, когда необходимо задать логику «объединительное или» и «разъединительное или».

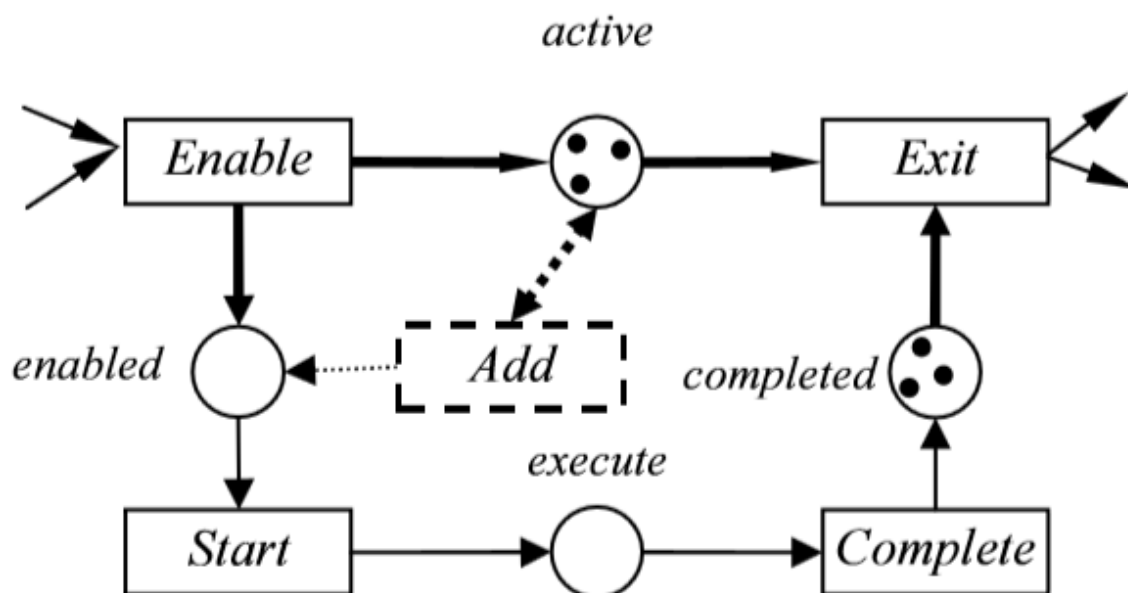


Рис. 5 Диаграмма перехода состояний для экземпляра задачи

Не смотря на все возможности языка YAWL, он обладает существенным недостатком, который заключается в том, что в нём есть возможность описывать только реактивное поведение, чего явно недостаточно для спецификации сценарного поведения пользовательского интерфейса.

Поскольку при формировании модели автоматизируемого процесса предполагается выбор сценария поведения изделия (АС или ПО), то необходимо учитывать временные задержки, пользовательские действия и неизвестные для изделия (АС или ПО) состояния. Другими словами необходимо учитывать концепцию «исключений», которая в последнее время в программировании является обыденной. Кроме того, аналитиками было отмечено, что данный язык является слишком сложным для восприятия на интуитивном уровне, что недопустимо в рамках концепции «программирование без программирования».

Язык Scenario Knowledge Base Language

Сравнительный анализ показал недостатки существующих языков спецификации взаимосвязанных процессов, нетрудно заметить, что ни один из них не обладает выразительными возможностями, необходимыми для спецификации проблем представления и управления процессами в приложениях человеко-ориентированного типа. Перечислим еще раз основные выявленные недостатки этих языков, которые ограничивают их использование в рассматриваемом классе задач:

- отсутствие явного представления потоков данных (WN);
- слабые возможности по описанию временных аспектов, в частности, временной синхронизации процессов (WN);
- отсутствие возможности выбора альтернативного сценария поведения, если такая необходимость возникла по некоторым причинам в реальном времени; неспособность описывать «интеллектуальное» поведение (YAWL, OMG);
- отсутствие возможности спецификации циклического поведения (Graph-based Workflow Language);
- ограничения на возможности обмена сообщениями (OMG);
- игнорирование того факта, что любое действие (переход в новое состояние) имеет продолжительность во времени (большинство языков).

Хотя перечисленные недостатки касаются разных языков, но важен тот факт, что среди них нет языка, свободного от всех перечисленных недостатков. Отметим ключевые особенности еще одного языка моделирования, который далее кратко называется SKBL (Scenario Knowledge Base Language):

- использование понятий атомарных и вложенных сценариев;

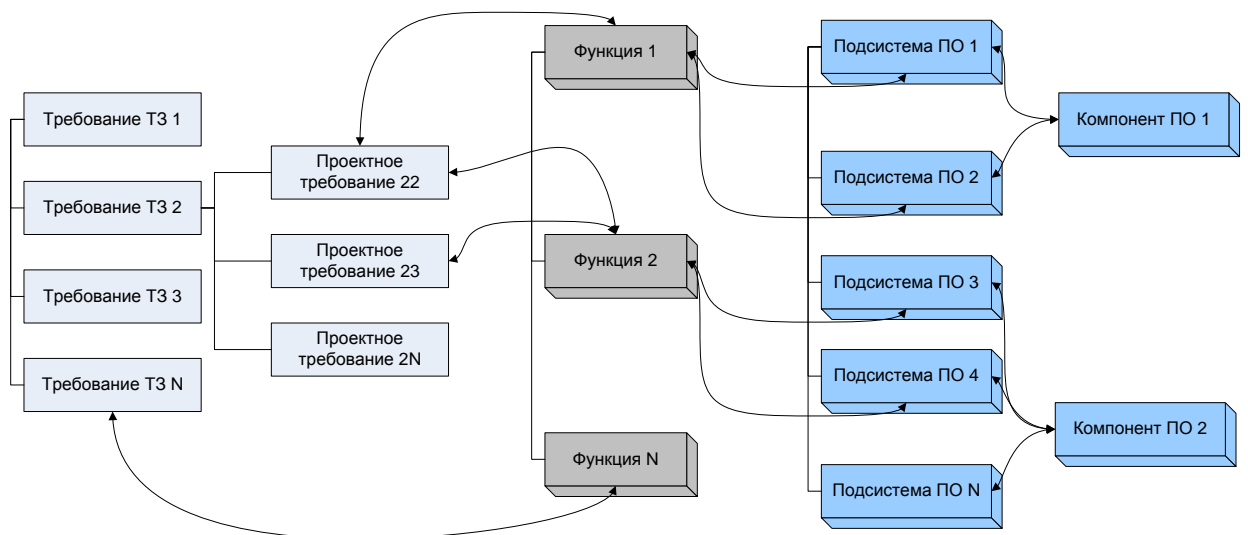
- возможность явного описания предусловий, задающих допустимость исполнения действий (временные условия, условия по доступности данных и др.);
- учет продолжительности исполнения действий;
- возможность диагностики нештатных ситуаций и их обработки.

Рассмотрим далее основные компоненты языка. Ключевыми понятиями разработанного языка являются понятия «состояние» и «действие». Формально «состояние» описывается в виде набора данных об объектах. Оно может соответствовать состоянию простого или сложного объекта, состоянию физической или виртуальной сущности. Состояния могут интерпретироваться как конечные или промежуточные цели, которые должны быть достигнуты.

Выбор подходящего языка моделирования предметной области зависит от специфики автоматизируемого процесса и конкретных условий применения, которые определяются и выявляются в разделе «Назначение и область применения». Оформить модель предметной области возможно на любом языке моделирования как из обсужденных в данном разделе, так и подобранных в самостоятельном порядке. Также возможно их комбинирование при необходимости.

Функциональная структура изделия. Алгоритмизация решения задачи автоматизации процессов предметной области

Чтобы установить функциональные блоки (подсистемы) ПО необходимо на основе полученного варианта структуры требований сформировать цепочку «требование – функция – подсистема ПО – компонент ПО» как показано на рисунке.



*Рис. 6 Обобщенный пример цепочки
«требования – функции – подсистема ПО –Компонент ПО»*

Проектные требования, формализованные на этапе анализа требований ТЗ, необходимо сопоставить с функциями, которые получены по результату описания вида деятельности автоматизации. Следующий шаг – формализация подсистем ПО и сопоставление с ранее полученными функциями ПО. Полученные подсистемы в основе реализации имеют компоненты ПО, которые представлены сервисами, СУБД, подпрограммами и т.д.

Таким образом, за каждой полученной подсистемой ПО закреплена своя задача или класс задач. Для алгоритмизации задач каждого модуля ПО принято соглашение на базе ГОСТ 19.701-90 «Схемы алгоритмов программ, данных и систем».

Оформление функциональной структуры и алгоритмизации задачи автоматизации процессов предметной области необходимо привести структурную схему, которая иллюстрирует цепочку «требование – функция – подсистема ПО – Компонент ПО» и схемы алгоритмов каждого модуля ПО.

Моделирование программной структуры и архитектурное описание изделия

В строгом значении архитектура ПО (software architecture) – описание подсистем и компонент программной системы, а также связей между ними. Архитектура пытается определить внутреннюю структуру получаемой системы, задавая способ, которым система организована или конструируется.

В середине 90-х, на волне распространения клиент-серверного подхода и начала его трансформации в «многозвенный клиент-сервер», призванный

обеспечить централизованное развертывание и управление общей (для клиентских приложений) бизнес-логикой, вопросы организации архитектуры программного обеспечения стали складываться в самостоятельную и достаточно обширную дисциплину. В результате, сформировалась точка зрения на архитектуру не только в приложении к конкретной программной системе, но и развился взгляд на архитектуру, как на приложение общих (generic) принципов организации программных компонент. В итоге, уже на сегодняшний день, на фоне такого развития понимания архитектуры, накоплен целый комплекс подходов и созданы (и продолжают создаваться и развиваться) различные архитектурные «фреймворки» (шаблоны), то есть систематизированные комплексы методов, практик и инструментов, призванные в той или иной степени формализовать имеющийся в индустрии опыт (как положительный – например, design patterns, так и отрицательный – например, anti-patterns) [1].

Принципы проектирования – это обычно независимые от предметной области стратегии, положенные в основу архитектурного проектирования. Однако есть принципы, относящиеся специально к проектированию программных (и других) систем для компьютеров [7].

Методы проектирования архитектуры – это обычно зависимые от области приложения (предметной области) тактические средства, определяющие те или иные этапы процесса проектирования. Сам процесс проектирования подразумевает принятие решения о форме и содержании предмета проектирования. Метод должен помогать разработчику в упорядочении принимаемых решений. Методы проектирования опираются, в конечном счете, на свойства мыслительного процесса разработчика и обеспечивают определенные алгоритмы принятия решений.

По представлению архитектуры программного обеспечения можно выделить следующие подходы [14]:

- модульно-интерфейсный подход;
- объектно-ориентированный;
- компонентный подход.

Дадим краткие пояснения для данных подходов.

Модульно-интерфейсный подход. В общем случае модульная программная система представляет собой древовидную структуру, в узлах которой размещаются программные модули, а направленные дуги показывают статическую подчиненность модулей. Если в тексте модуля имеется ссылка на другой модуль, то их на структурной схеме соединяет дуга, которая исходит из первого модуля и входит во второй модуль. При этом модульная структура программной системы, кроме структурной схемы, должна включать в себя совокупность спецификаций модулей, образующих эту систему.

Функции верхнего уровня обеспечиваются главным модулем. Он управляет выполнением нижестоящих функций, которым соответствуют подчиненные модули. При определении набора модулей, реализующих функции конкретного алгоритма, необходимо учитывать следующее:

- модуль вызывается на выполнение вышестоящим по иерархии модулем и, закончив работу, возвращает ему управление;
- принятие основных решений в алгоритме выносится на максимально высокий по иерархии уровень;
- если в разных местах алгоритма используется одна и та же функция, то она оформляется в отдельный модуль, который будет вызываться по мере необходимости.

Модульно-интерфейсный подход – один из методов структурного проектирования. Спецификации модулей и их интерфейсов дают структурную основу для проектирования каждого модуля и системы в целом. Спецификация программного модуля состоит из функциональной спецификации, описывающей семантику функций, выполняемых этим модулем по каждому из его входов, и синтаксической спецификации его входов, позволяющей построить на используемом языке программирования синтаксически правильное обращение к модулю.

Объектно-ориентированный подход. Выявление классов, правильно описывающих предметную область проекта, является самой сложной задачей, которую предстоит решить разработчику. Выявление нужных классов намного сложнее процесса создания диаграмм. Если команда разработчиков не смогла определить классы, точно описывающие задачу, то неважно, сколько средств будет потрачено на средства разработки, в конечном счете есть большая вероятность того, что созданные модели окажутся неработоспособными.

Не каждый класс является классом предметной области. Массивы, наборы и классы графического интерфейса не являются классами предметной области. Классы предметной области – это сущности, описывающие задачу предметной области, например, такие классы, как студент, преподаватель, регистрация, экзамены в приложении, хранящем информацию о поступающих в вуз, или такие классы, как заказы, люди, уголовные дела, счета, денежные вклады и др. Каждому классу предметной области соответствует объект предметной области.

Объект – это сущность, которая используется при выполнении некоторой функции или операции (преобразования, обработки, формирования и т.д.). Объекты могут иметь динамическую или статическую природу: динамические объекты используются в одном цикле воспроизводства, например заказы на продукцию, счета на оплату, платежи; статические объекты используются во многих циклах воспроизводства, например, оборудование, персонал, запасы материалов. На концептуальном уровне

построения модели предметной области уточняется состав классов объектов, определяются их атрибуты и взаимосвязи. Таким образом строится обобщенное представление структуры предметной области.

Компонентный подход. Компоненты представляют собой автономный код (подсистему), который может быть повторно использован за счет его независимого развертывания. Компоненты не обязательно должны быть большими по объему кода, но в основном они больше по объему отдельного класса или группы слабо связанных классов. Компоненты имеют множество предоставляемых и требуемых интерфейсов и используются в больших сложных приложениях с десятками и сотнями классов предметной области. В сложных случаях разработки больших систем, например при разработке корпоративной информационной системы потребуются диаграммы компонентов.

При разработке компонентной системы методом снизу вверх в первую очередь определяются классы предметной области, т.е. те, которые служат непосредственно для решения задач предметной области, а не вопросов архитектуры приложения. В результате основные усилия концентрируются, прежде всего, на решении поставленной задачи вместо описания сложной архитектуры. Применяя метод разработки снизу вверх и выявляя классы предметной области, проектировщики имеют большой потенциал для решения поставленной задачи. Кроме того, всегда можно объединить классы предметной области в компоненты, если сложность проекта растет, некоторые группы классов можно развернуть и повторно использовать, включив их в состав компонентов.

Оба метода проектирования компонентных систем имеют право на использование. Для небольших приложений и приложений среднего размера, вероятно, не потребуются компоненты и в этих случаях можно воспользоваться методом снизу вверх. Для приложений корпоративного масштаба потребуется более мощная методология, и в этом случае метод проектировании сверху вниз может оказаться более подходящим.

Архитектурное решение. Любая архитектура, с одной стороны, оказывает влияние на поведение, производительность, отказоустойчивость и удобство сопровождения, но с другой – ее характер формируется с учетом требований, выставляемых ко всем этим атрибутам [2].

Приведем пример архитектурного решения системы управления воздушным движением. Рассмотрение данного примера будет дано со следующих позиций:

- описание физической среды размещения ПО системы управления воздушным движением;
- перечисление ряда программно-архитектурных представлений;
- обозначение тактик реализации программно-архитектурных решений.

Перед системой управления воздушным движением были поставлены следующие задачи:

- получение отчетов о радиолокационных целях, которые хранятся в существующей систем управления полетами;
- преобразование радиолокационных отчетов в форму, пригодную для отображения на экране, и их пересылка на все консоли;
- обработка предупреждений о конфликтах, передаваемых с хост-компьютера;
- предоставление хост-компьютеру интерфейса для подачи входных данных и получения планов;
- предоставление подробных данных текущего контроля и управляющей информации, при помощи которых администраторы узлов проводят реконфигурацию систем реального времени;
- обеспечение средств резервирования на случай сбоев на хост-компьютере, в основной сети передачи данных и в основных радиолокационных датчиках.

Физическая среда размещения ПО системы управления воздушным движением. Система управления воздушным движением является распределенной системой, многочисленные элементы которой соединяются посредством локальных сетей.

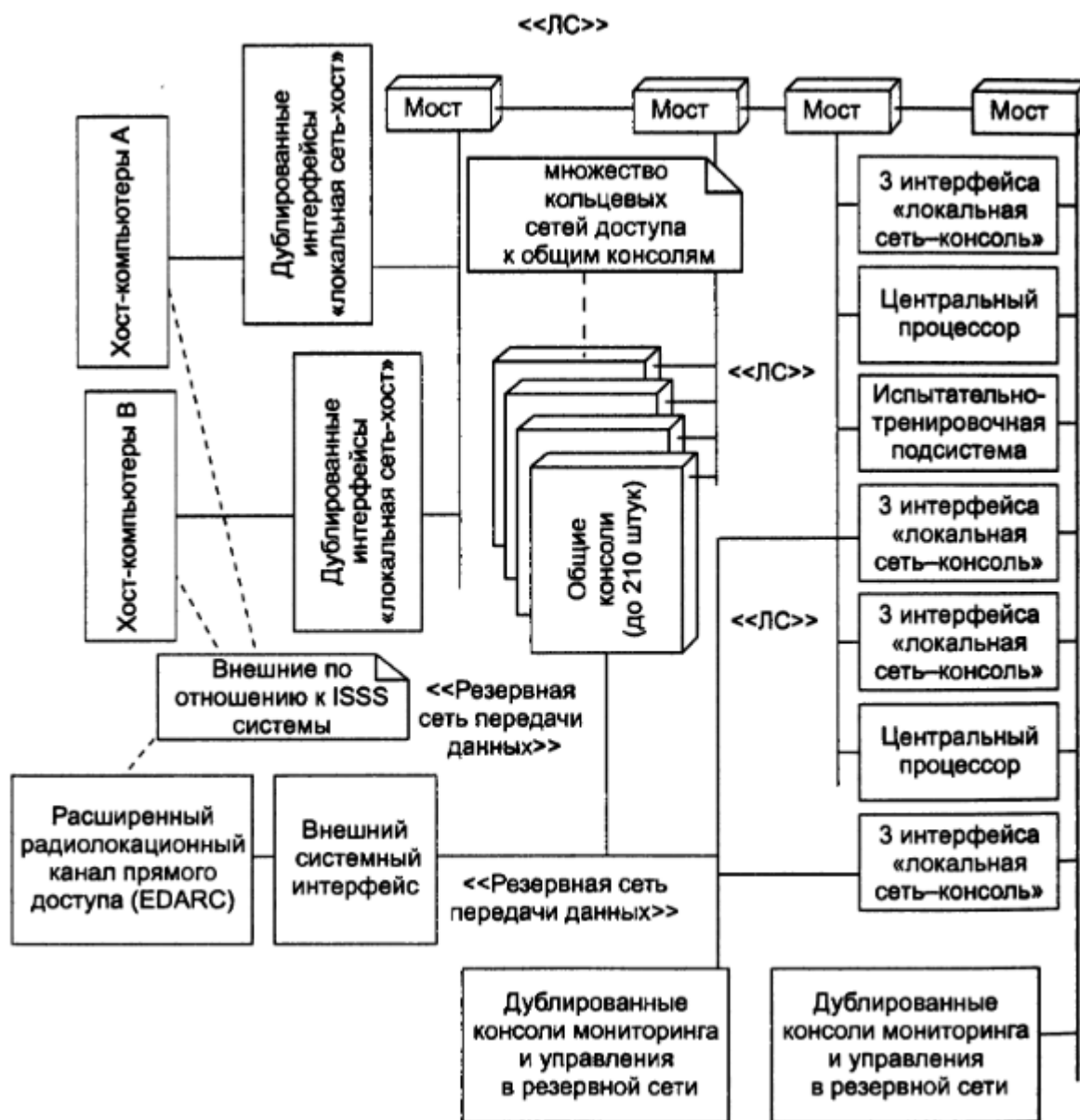


Рис. 7 Физическое представление системы управления воздушным движением

Основными элементами системы воздушного движения являются:

- хост-компьютер, который является ядром системы;
- общие консоли, которые выступают в роли рабочих станций;
- локальная сеть, которая связывает хост-компьютеры с общими консолями;
- параллельные кольцевые сети с маркерным доступом;
- расширенный радиолокационный канал прямого доступа (EDARC);
- резервированная сеть передачи данных;
- испытательно-тренировочная система;
- центральный процессор (ЭВМ).

Представление декомпозиции модулей. В системе управления воздушным движением пять ключевых элементов:

1. Элемент управления выводом.
2. Общие системные службы.
3. Модуль «Запись, анализ и считывания».
4. Модуль внесения изменений в ПО хост-компьютера.
5. Операционная система.

Перечисленные элементы являются поставляемыми модулями документации и программ.

Клиент-серверное представление. Поскольку взаимодействие между приложениями проходит по модели клиент-сервер, имеет смысл рассмотреть соответствующее представление (составлено на языке UML).

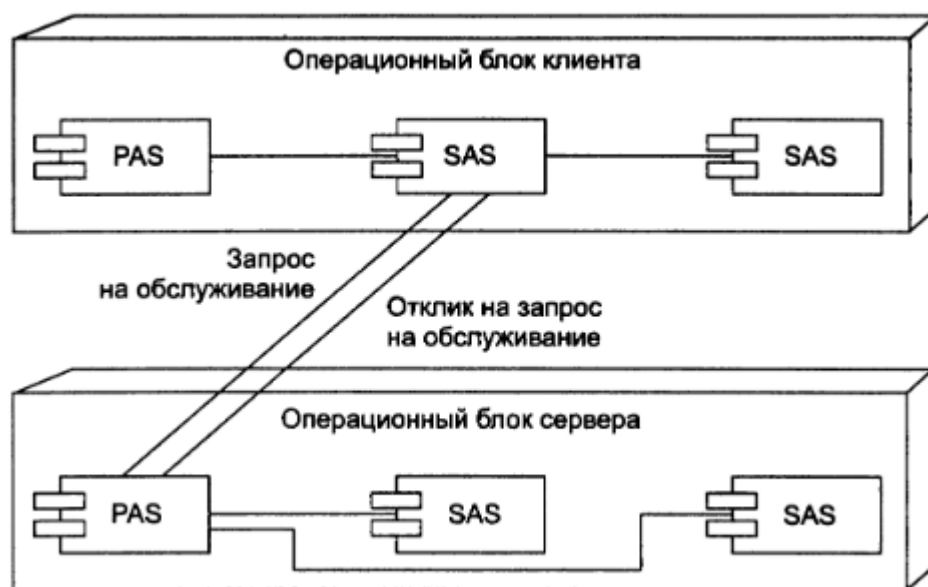


Рис. 8 Клиент-серверное представление архитектуры ПО системы управления воздушным движением

Конструкция клиентов и серверов предполагает наличие в них совместимых интерфейсов. Этому содействует практика применения простых протоколов передачи данных. Результат соответствует положениям ряда тактик реализации модифицируемости: «поддержка стабильности интерфейсов», «заменяемость компонентов», «применение предписанных протоколов».

Многоуровневое представление. ПО системы управления воздушным движением исполняются в операционной среде AIX –одной из коммерческих разновидностей UNIX.

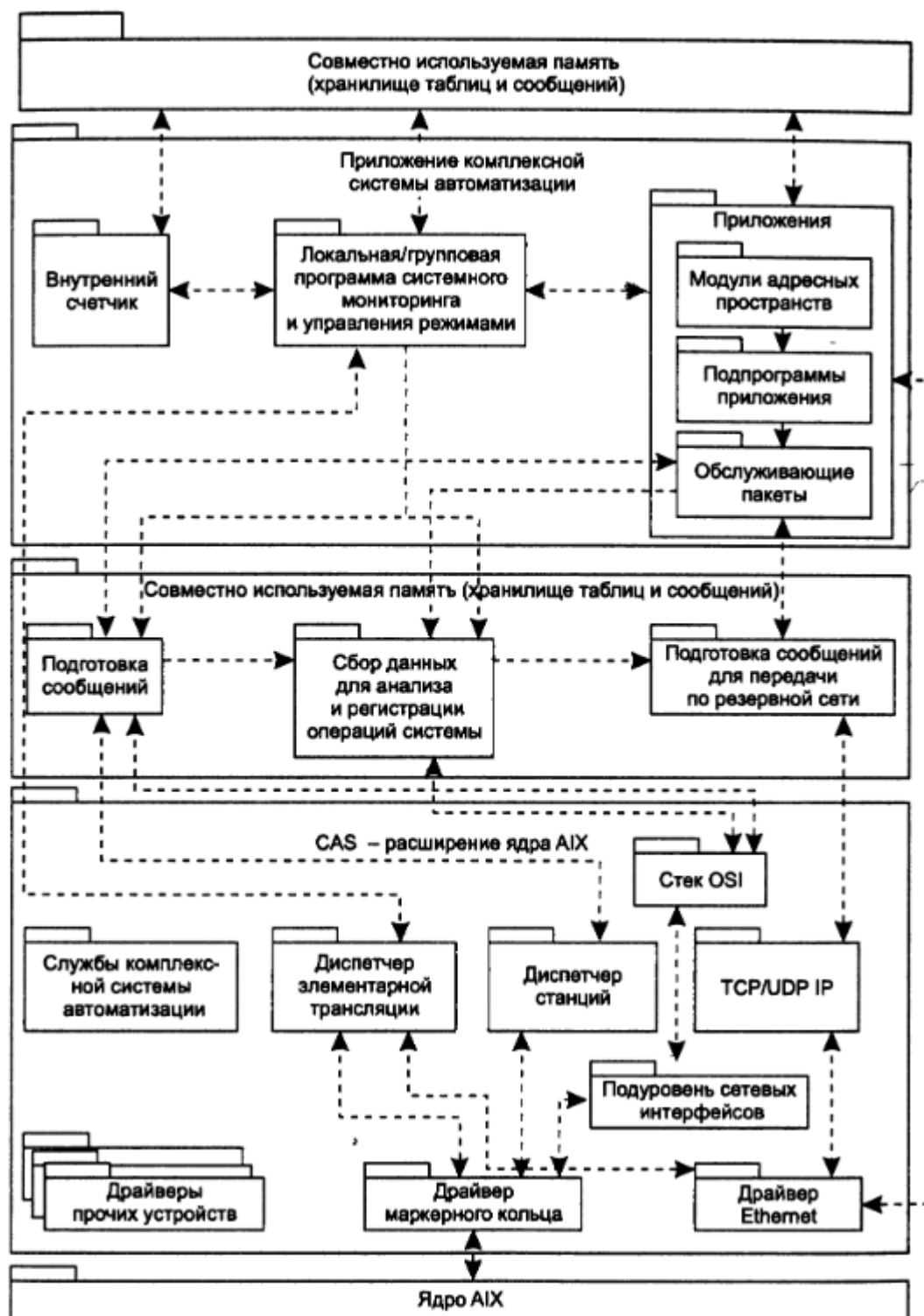


Рис. 9 Многоуровневое представление ПО системы управления воздушным движением

Уровни программной архитектуры системы управления воздушным движением на схем демонстрируют потоки данных и управления – таким образом, получается «накладка» многоуровневого представления на представление компонент-соединитель.

Представление «компонент-соединитель». Из-за повышенных требований к готовности системы управления воздушным движением

отказоустойчивость стала неотъемлемой характеристикой ее проектного решения.

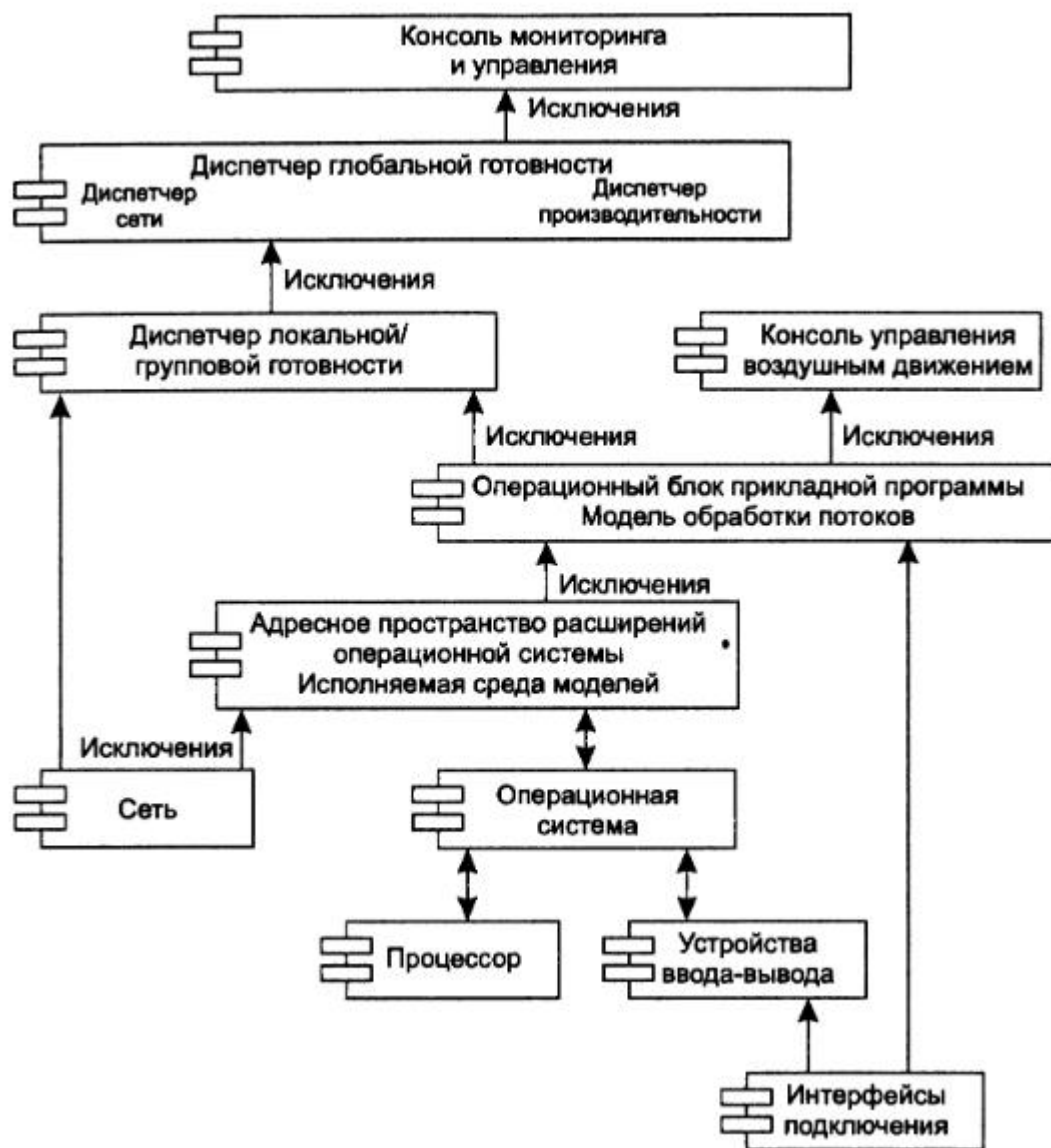


Рис. 10 Представление «компонент-соединитель» отказоустойчивого функционирования системы управления воздушным движением

Иерархия отказоустойчивости предусматривает несколько уровней обнаружения неисправностей и восстановления. Каждый из них в асинхронном режиме выполняет следующие действия:

- проводит обнаружение ошибок на своем уровне и на нижележащем;
- обрабатывает исключения, порождаемые на нижележащих уровнях;
- проводит диагностические мероприятия, восстановление, составляет отчетность и порождает исключения.

Тактики реализации программно-архитектурных решений. В рамках данных методических указаний тема тактик реализации программно-

архитектурных решений подробно раскрыта не будет, поэтому предлагается перечислить некоторые из них:

- взаимоотношения представлений;
- адаптационные данные (проработка конфигурационных данных);
- поддержание стабильности интерфейсов;
- семантическая связность;
- ограничение внешних воздействий;
- и т.д.

Приведенный материал по теме моделирования программной структуры и архитектурного описания изделия (АС или ПО) не является исчерпывающим. Всего перечня, необходимого для отображения всех аспектов архитектуры даже не очень крупной программной системы не может быть в прицепе. На сегодняшний день наметилась положительная тенденция осознания того, что от архитектурных решений зависит реализация атрибутов качества вследствие чего, архитектурному проектированию стали уделять больше внимания при разработке ПО.

Сравнительный анализ аналогов изделия

Проведение сравнительного анализа проводится по ряду введенных критериев в соответствии со спецификой задачи автоматизации. Приведем пример такого сравнительного анализа CASE-систем в задаче автоматизации информационно-аналитической поддержки ЖЦ сложных организационно-технических объектов.

Состав критериев был определен следующий:

- в части информационного моделирования:
 1. описания концептуальных схем разнотипных источников данных в терминах предметной области;
 2. описания взаимосвязей терминов предметной области различных источников данных;
- в части поведенческого моделирования:
 3. описания бизнес-процессов информационно-аналитической поддержки в терминах действий, событий и условий их применения;
 4. описания взаимодействия бизнес-процессов (EAI);
- в части функционального моделирования:

5. описания структурно-функциональных моделей решения аналитических вычислительных задач, ориентированных на распознавание образов (классов состояния исследуемых объектов);
6. описания моделей запросов к источникам данных в терминах предметной области;
7. соответствия ETL-принципу;

— в части проектирования:

8. верификации спецификаций, схем программ, текста запросов на предмет их корректности;
9. формальной проверки согласованности спецификаций;
10. наличия графической нотации;
11. автоматического синтеза совокупности схем программ информационно-аналитической поддержки с запросами к источникам данных;

— в части функционирования:

12. параллелизма, асинхронности и потоковости вычислительного процесса с реализацией принципа управления по данным;
13. функционирования в режиме РВ;
14. синхронизации программ информационно-аналитической поддержки по событиям.
15. направленности процесса вычислений в соответствие с целью функционирования.

Результат сравнительного анализа приведен на рисунке ниже.

Модели		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Программные продукты
Инф. модели	IDEFIX, ERD, Диаграмма классов UML	+/-	-	-	-	-	-	-	+/-	-	+	-	-	-	-	-	Erwin, Design/IDEF, S-Designer, PowerDesigner, Oowin, Designer/2000 Rational Rose
	Онтологии для OBDA-систем, Express-модели	+	+	-	-	-	+/-	-	+/-	-	+/-	-	-	-	-	+	Apache Jena, Protege, Unigraphics NX, T-Flex CAD
Концепт. модели процессов	Процессы Милнера, STRIPS, PSL	-	-	-	+	-	-	-	+/-	-	-	-	+	-	-	+	UPPAAL, KRONOS, STRIPS
	EDPC, Workflow Net, WPD, YAWL, GBWL, BPMN, UML, SKBL, SDL, OOSE, IDEF0,2,3, SADT	-	-	+	+	-	-	-	+/-	-	+	-	+/-	-	+	+	SmartDraw, Workflow engine, Div. WF and Modeling Tools, YAWL system, ELMA BPM, Microsoft Visio, IBM Rational SDL, BPwin, PowerDesigner, Dia, Design/IDEF
Функц. модели, потоки данных	DFD, Д-ммы переходов состояний, Сети Петри	-	-	-	-	-	-	-	+/-	-	+	-	+	+/-	-	+	Silverrun, PowerDesigner, CASE – Аналитик, Workflow Petri Net Designer
	Дракон-схемы, Н-модели, Выч. модели, G-модели	-	-	-	-	+	-	-	+/-	-	+	+/-	+	+	+	+	Дракон, ПРИЗ, АПИД/Анализ, ТАО
Онтолог. система СИАП ЖЦ СОТО		+	+/-	+	+	+	+	+	+/-	+	+	+	+	+	+	+	СИАП ЖЦ СОТО

Рис. 11 Сравнительный анализ CASE-систем в задаче автоматизации информационно-аналитической поддержки ЖЦ сложных организационно-технических объектов (СИАП ЖЦ СОТО как пример разрабатываемого изделия)

Раскрыть тему сравнительного анализа возможно в произвольном формате (таблицы, графики, иллюстрации и т.д.).

Перспективы применения технологических и технических решений

В теме перспектив применения технологических и технических решений раскрывается возможное применение изделия, которое в явном виде не подразумевалось на этапе замысла. В данном подразделе должны быть описаны задачи, которые изделие принципиально может решать, но требований по их реализации не закладывались на этапе разработки ТЗ. Кроме того, здесь указываются заложенные в конструкцию (архитектуру) решения, которые могут оказаться востребованы при интеграции со смежными системами.

Раздел "Описание организации работ с применением разрабатываемого изделия"

Организация коллективной разработки

Проект и управление проектом. В различных источниках можно найти различные определения понятия «управление»:

- элемент, функция организованных систем различной природы (биологических, социальных, технических), обеспечивающая сохранение их определенной
- структуры, поддержание режима деятельности, реализацию их программ и целей;
- руководство, направление чей-либо деятельности;
- изменение состояния объекта, системы или процесса, ведущее к достижению поставленной цели (словарь по кибернетике).

С точки зрения последнего определения, существенным является:

- наличие измеряемой цели управления (функции цели);
- наличие (возможность) управления (управляющего воздействия, влияющего на изменение функции цели);
- наличие измерений состояния объекта или процесса;
- ограниченность управления.

Слово «проект» происходит от латинского *projectus*, что означает «брошенный вперед». Приведем несколько определений понятия проект.

- Проект – это произвольный ряд действий или задач, имеющий определенную цель, которая будет достигнута в рамках выполнения некоторых заданий, характеризующимися определенными датами начала и окончания, пределами финансирования и ресурсами (Г. Керцнер).
- Проект – одноразовая работа, которая имеет определенные даты начала и окончания, ясно определенные цели, возможности и, как правило, бюджет (Д. Льюис).
- Проект – временное усилие, применяемое для того, чтобы создать уникальный продукт или услугу с определенной датой начала и окончания действия, отличающегося от продолжающихся, повторных действий и требующего прогрессивного совершенствования характеристик (PMI).

В этих определениях в той или иной степени отражаются следующие определяющие характеристики проекта:

1. Цель проекта. Наличие четко выраженного конечного результата, выхода, продукции, определяемых в терминах затрат, качества и времени реализации.
2. Уникальность. Проект – это разовое начинание, которое не будет повторяться. Даже “повторяющиеся” проекты, например, по строительству еще одного предприятия по той же проектной документации, значительно отличаются друг от друга используемыми ресурсами и средой реализации.
3. Ограниченность во времени. Проект имеет начало и конец. Для его реализации необходима временная концентрация ресурсов. По минованию надобности, ресурсы используются на другие цели.
4. Ограниченность ресурсов, выделяемых на выполнение проекта (финансовых, людских, материальных).
5. Сложность. Для достижения целей проекта необходимо решить множество задач. Отношения между задачами могут быть довольно сложными, особенно, если в проекте много задач.
6. Неопределенность. Возможность достижения цели в указанные сроки с выделенными ресурсами заранее не гарантирована.
7. Предсказуемость. По мере реализации проекта, изменяется потребность в тех или иных ресурсах. Это изменение идет в определенной предсказуемой последовательности, определяемой жизненным циклом проекта.

Иными словами, проект – это достаточно сложный вид деятельности, которым сложно управлять в силу его уникальности и ограниченности ресурсов и времени. Это обстоятельство вносит в проект элемент неопределенности, а правильно организованное управление делает результаты предсказуемыми. Кстати, предсказуемый – не значит успешный. Это значит – вовремя заверченный (успешно) или вовремя прекращенный (неуспешно) [].

Свод знаний по управлению проектами, который разработан и развивается Институтом Проектного Менеджмента содержит описание состава знаний

1. Управление интеграцией проекта (Integration)
 - a. Создание плана проекта (Project Plan Development)
 - b. Исполнение плана проекта (Project Plan Execution)
 - c. Контроль изменений в проекте (Integrated Change Control)
2. Управление объемом работ (Scope)
 - a. Инициирование (Initiation) - формальное принятие решения о начале проекта (следующей фазы проекта)
 - b. Планирование объема работ (Scope Planning) - разработка документа, описывающего объем работ
 - c. Формализация объема работ (Scope Definition) - декомпозиция всего объема работ (основных необходимых результатов) на мелкие, измеримые задачи

- d. d. Верификация (Scope Verification) - подтверждение объема работ – формальная проверка приемлемости результатов работы.
- e. e. Управление изменениями объема работ (Scope Change Control) - контроль и утверждение изменений
- 3. Управление временем выполнения (Time)
 - f. Определение состава работ (Activity Definition)
 - g. Определение взаимосвязей работ (Activity Sequencing)
 - h. Оценка длительностей работ (Activity Duration Estimating)
 - i. Составление расписания проекта (Schedule Development)
- 4. Управление стоимостью (Cost)
 - a. Планирование ресурсов (Resource Planning) - какие ресурсы в каком количестве нужны для работ
 - b. Оценка стоимостей (Cost Estimating) - ресурсов, необходимых для работ
 - c. Разработка бюджета (Cost Budgeting) - бюджетирование – распределение затрат по компонентам проекта
 - d. Контроль стоимости (Cost Control) - управление изменениями бюджета
- 5. Управление качеством (Quality)
 - a. Планирование качества (Quality Planning) - определение стандартов качества и средств для их достижения
 - b. Обеспечение качества процесса (Quality Assurance) - плановая, регулярная оценка исполнения – проверка производственных процессов
 - c. Контроль качества результатов (Quality Control) – мониторинг результатов проекта, определение их соответствия стандартам, выявление и устранение причин несоответствия качества
- 6. Управление персоналом (Human Resource)
 - a. Организационное планирование (Organizational Planning) – идентификация, документирование, и назначение проектных ролей, обязанностей и структуры отчетности
 - b. Подбор кадров (Staff Acquisition) - получение необходимых для проекта человеческих ресурсов, назначение персонала в команду проекта
 - c. Развитие команды проекта (Team Development) – повышение производительности труда: индивидуальной и команды в целом (улучшение взаимодействия)
- 7. Управление коммуникациями (Communications)
 - a. Планирование взаимодействия (Communications Planning) – определение потребностей участников проекта в информации и планирование информационных потоков
 - b. Распределение информации (Information Distribution) - регулярное и своевременное обеспечение участников проекта необходимой информацией

- c. Оценка исполнения (Performance Reporting) - сбор и распространение отчетности о текущем состоянии проекта, достигнутом прогрессе и ожидаемых результатах
 - d. Административное завершение (Administrative Closure) - создание, распространение (уничтожение) информации, необходимые для формального завершения проекта/фазы
8. Управление рисками (Risk)
- a. Планирование управления рисками (Risk Management Planning)
 - b. Идентификация рисков (Risk Identification)
 - c. Качественный анализ рисков (Qualitative Risk Analysis)
 - d. Количественный анализ рисков (Quantitative Risk Analysis)
 - e. Планирование реагирования на риски (Risk Response Planning)
 - f. Мониторинг и контроль рисков (Risk Monitoring and Control)
9. Управление закупками и поставками (Procurement)
- a. Планирование закупок (Procurement Planning) - определение какие продуктов и услуг нужны извне
 - b. Планирование предложений (Solicitation Planning) – документирование требований к продуктам и услугам от внешних поставщиков
 - c. Получение предложений (Solicitation)
 - d. Выбор поставщиков (Source Selection)
 - e. Управление контрактами (Contract Administration) - регулирование отношений с поставщиками
 - f. Завершение контрактов (Contract Closeout) - подтверждение выполнения, разрешение споров.

Управление программным проектом включает решение трех основных задач: подбор и управление командой, выбор процесса и выбор инструментальных средств. Хотя все три задачи одинаково важны для успеха проекта, ведущую роль играет правильный подбор и управление командой. Успех проекта во многом зависит от того, насколько состав участников проекта сможет быть преобразован в команду единомышленников, насколько эта команда будет активной и инициативной с одной стороны и управляемой с другой. Из множества вопросов управления командой проекта мы рассмотрим три:

- ролевая модель команды.
- модели организации команд.
- общение в команде.

Ролевая модель команды. Состав команды определяется опытом и уровнем коллектива, особенностями проекта, применяемыми технологиями и

уровнем этих технологий. «Классический» вариант состава команды включает следующие позиции:

1. Менеджер проекта – главное действующее лицо, обладающее знаниями и навыками, необходимыми для успешного управления проектом. Его основные функции:
 - Подбор и управление кадрами
 - Подготовка и исполнение плана проекта
 - Руководство командой
 - Обеспечение связи между подразделениями
 - Обеспечение готовности продукта
2. Проектировщик – это функция проектирования архитектуры высокого уровня и контроля ее выполнения. В небольших командах функция распределяется между менеджером и разработчиками. В больших проектах это может быть целый отдел. Основными функциями проектирования являются:
 - Анализ требований
 - Разработка архитектуры и основных интерфейсов
 - Участие в планировании проекта
 - Контроль выполнения проекта
 - Участие в подборе кадров
3. Разработчик – роль, ответственная за непосредственное создание конечного продукта. Помимо собственно программирования (кодирования) в его функции входит:
 - Контроль архитектурных и технических спецификаций продукта
 - Подбор технологических инструментов и стандартов
 - Диагностика и разрешение всех технических проблем
 - Контроль за работой разработчиков документации, тестирования, технологий
 - Мониторинг состояния продукта (ведение списка обнаруженных ошибок)
 - Подбор инструментов разработки, метрик и стандартов. Контроль их использования.
4. Тестировщик – роль, ответственная за удовлетворение требований к продукту (функциональных и нефункциональных). В функции тестировщика входит:
 - Составление плана тестирования. План тестирования составляет один из элементов проекта и составляется до начала реализации (разработки) проекта. Время, отводимое в плане на тестирование может быть сопоставимо с временем разработки.
 - Контроль выполнения плана. Важнейшая функция контроля – поддержка целостности базы данных зарегистрированных ошибок. В этой базе регистрируется: кто, когда и где обнаружил,

описание ошибки, описание состояния среды; статус ошибки: приоритет, кто разрешает; состояние ошибки: висит, в разработке, разрешена, проблемы

- Разработка тестов. Самая трудоемкая часть в работе тестировщика. Тестирование должно обеспечить полную проверку функциональности при всех режимах работы продукта.
- Автоматизация тестирования включает автоматизацию составления тестов, автоматизацию пропуска тестов и автоматизацию обработки результатов тестирования. В виду важности автоматизации тестирования, иногда вводят нового участника – инженера по автоматизации.
- Выбор инструментов, метрик, стандартов для организации процесса тестирования.
- Организация Бета тестирования - тестирования почти готового продукта внешними тестерами (пользователями). Эту важную процедуру надо продумать и организовать в случае разработки коробочного продукта.

5. Инженер по качеству. В современном представлении рассматривается три аспекта (уровня) качества:

- качество конечного продукта – обеспечивается тестированием,
- качество процесса разработки (тезис: для повышения качества продукта надо повысить качество процесса разработки),
- качество (уровень) организации (тезис: для повышения качества процесса надо повысить качество организации работ). В некоторых случаях функции инженера по качеству возлагаются на тестировщика. На самом деле они шире – два следующих уровня качества. Здесь приведены функции, отличные от функции тестировщика:
- Составление плана качества. План качества включает все мероприятия по повышению качества (на всех уровнях). Имеет долговременный характер. План тестирования – его оперативная составляющая.
- Описание процессов. Описание процессов является их формализацией. При описании вводятся метрики процесса, влияющие на качество продукта.
- Оценка процессов включает регистрацию хода выполнения процессов и оценку значений установленных метрик процессов. Выявление «слабых» мест и выработка рекомендаций по улучшению процессов.
- Улучшение процессов - переопределение процесса, автоматизация части работ, обучение персонала. Повышение качества процессов требует участия всех действующих лиц. Принятое решение должно быть обосновано, всем понятно и всеми принято. При повышении качества организации работа по улучшению

процессов проводится по определенной схеме. На каждом шаге повышения уровня организации работ выделяются ключевые процессы и выполняются работы по улучшению этих процессов.

6. Технический писатель или разработчик пользовательской (и иной) документации как части программного продукта. Функциями технического писателя являются:

- Разработка плана документирования, который включает состав, сроки подготовки и порядок тестирования документов.
- Выбор и разработка стандартов и шаблонов подготовки документов
- Выбор средств автоматизации документирования
- Разработка документации
- Организация тестирования документации
- Участие в тестировании продукта. Технический писатель все время работает с продуктом (его готовыми версиями) и выступая от имени пользователя видит все недочеты и несоответствия.

7. Технолог разработки ПО обеспечивает выполнение следующих задач:

- Поддержка модели ЖЦ - создание служб и структур по поддержке работоспособности принятой модели ЖЦ ПО. В поддержке модели ЖЦ принимают участие все. Но контроль возложен на технолога.
- Создание и сопровождение среды сборки продукта. Функция особенно важна на завершающих этапах разработки или при использовании модели прототипирования. В такой ситуации сборка будет проводиться достаточно часто (в некоторых случаях - ежедневно). Среда сборки должна быть подготовлена заранее, сборка должна проводиться быстро и без сбоев. С учетом сборки версий это не простая задача.
- Создание и сопровождение процедуры установки с тем, чтобы каждая сборка устанавливалась автоматически с учетом версии и конфигураций сред.
- Управление исходными текстами – сопровождение и администрирование системы управления версиями исходных текстов.

Выделенные позиции не обязательно представлены конкретными людьми. Это список основных функциональных ролей в команде (ролевая модель команды). В малых командах роли могут совмещаться. В больших – выделяться группы или отделы (отдел проектирования, отдел тестирования, отдел контроля качества, отдел подготовки документации).

Административная модель организации команды разработки. Это традиционный стиль управления, связанный с иерархической административно-командной моделью, которую используют военные организации. В основе лежит теория Х, которая утверждает, что такой подход

необходим, поскольку большинство людей по своей природе не любит работу, и будет стремиться избежать ее, если у них есть такая возможность. Однако менеджеры должны принуждать, контролировать, направлять сотрудников и угрожать им, чтобы получить от них максимальную отдачу. Девиз теории и модели: «Люди делают только то, что вы контролируете». Или в более мягком варианте: «Люди делают то, что они не хотят делать, только если вы их контролируете». Теория утверждает, что большинство людей предпочитают, чтобы им говорили, что следует делать и им не придется ничего решать самим.

Характерные черты модели:

- Властная пирамида – решения принимаются сверху-вниз.
- Четкое распределение ролей и обязанностей.
- Четкое распределение ответственности.
- Следование инструкциям, процедурам, технологиям.
- Роль менеджера: планирование, контроль, принятие основных решений.

Преимущества модели: ясность, простота, прогнозируемость. Модель хорошо сочетается с каскадной моделью ЖЦ и применима в тех же случаях, что и каскадная модель. Модель эффективна в случае установившегося рабочего процесса. Недостатки модели связаны с тем, что административная система стремится самосохранению (стабильности) и плохо восприимчива к изменению ситуации – новые типы проектов, применение новых технологий, оперативная реакция на изменение рынка. Кроме того, в административной модели плохо уживаются индивидуалисты и генераторы идей.

Модель хаоса. В основе модели хаоса лежит Теория Y, которая является полной противоположностью Теории X. Основной тезис Теории Y: работа — естественная и приятная деятельность и большинство людей, на самом деле, очень ответственны и не увиливают от работы.

- Характерными чертами модели хаоса являются:
- Отсутствие явно выраженных признаков власти.
- Роль менеджера – поставить задачу, обеспечить ресурсами, не мешать и следить, чтобы не мешали другие.
- Отсутствие инструкций и регламентированных процедур.
- Индивидуальная инициатива - решения по проблеме принимаются там, где проблема обнаружена.
- Процесс напоминает творческую игру участников на основе дружеской соревновательности.

Преимущества такой модели в том, что творческая инициатива участников ничем не связана и потенциал участников раскрывается в полной мере. Это бывает особенно эффективно в случае, когда для решения проблемы требуется поиск новых подходов, методов, идей и средств. Команда становится командой «прорыва», а работа проходит в форме игры, цель которой – поиск наилучшего результата. Процесс напоминает случайный

поиск, когда идеи и решения рождаются при живом и как бы случайном обсуждении проблем в коридоре, столовой на пикнике. Собрать такую команду в рабочей комнате и устроить обсуждение по регламенту часто просто не удастся – это команда творческих индивидуалистов.

Недостатки модели связаны с тем, что при определенных условиях команда прорыва может стать командой провала. Причинами провала могут быть:

- Творческая соревновательность переходит в конкуренцию сначала идей, а потом – личностей.
- Процесс начинает преобладать над целью проекта – высказанные идеи не доводятся до конца и сменяются новыми идеями, преобладание получают «красивые» идеи, лежащие в стороне от основных целей проекта.
- Люди, способные к генерации идей, редко обладают терпением доведения идей до полной реализации.

Модель открытой архитектуры. Административная и хаотическая модели являются двумя «крайностями», между которыми находятся множество моделей, сочетающих преимущества «крайних» моделей. Одной из таких моделей является модель открытой архитектуры, основанная на Теории Z. Эта теория была сформулирована Уильямом Оучи на основе изучения опыта японского стиля управления (Theory Z: How American Business Can Meet the Japanese Challenge,» Perseus Publishing, 1981). Теория Z предполагает (но не декларирует) наличие внутреннего механизма управления, основанного на влиянии со стороны коллег и группы в целом. Дополнительное воздействие оказывают культурные нормы конкретной корпорации. Основной принцип модели можно сформулировать так: «Работаем спокойно. Работаем вместе». Особенности этой модели являются [7]:

- Адаптация к условиям работы – если делаем независимые модули, то расходимся и делаем, если нужна архитектура базы данных, то собираемся вместе и обсуждаем идеи.
- Коллективное обсуждение проблем, выработка консенсуса и принятие решения – не все могут согласиться, но принятое решение является коллективным и в силу этого – обязательным для всех.
- Распределенная ответственность – отвечают все, кто обсуждал, вырабатывал, принимал.
- Динамика состава рабочих групп в зависимости от текущих задач.
- Отсутствие специализации – участники меняются ролями и функциями и могут при необходимости заменить друг друга.
- Задача менеджера – активное (но рядовое, не руководящее) участие в процессе, контроль конструктивности обсуждений, обеспечение возможности активного участия всех.

Принципы управления качеством проекта. Качество – это свойство товара (услуги) наиболее полно удовлетворять требованиям и пожеланиям потребителя. Введенное понятие качества есть «качественный» показатель продукта. Наиболее общим является подход, при котором вводятся понятия:

- Ценность изделия – способность удовлетворять потребности.
- Качество изделия – соответствие между свойствами изделия и его ценностью.
- Мера качества – соотношение ценности и стоимости.

Следующим важным шагом в обеспечении качества стал переход к управлению качеством, цель которого состояла не в том, чтобы обнаружить и изъять негодные изделия до их отгрузки покупателю, а в том, чтобы увеличить выход годных изделий в процессе производства. Основными этапами такого перехода стали:

- Управление процессами – переход от контроля к управлению отдельными процессами.
- Управление производством – переход от управления отдельными процессами к управлению производством в целом.

Главная идея перехода от контроля к управлению качеством состояла в том, что качество продукта должно обеспечиваться качеством процесса его изготовления. В 1957 г. Фейгенбаум опубликовал статью, в которой изложил 8 принципов TQM (Total Quality Management) тотального управления качеством и параллельного (одновременного) инжиниринга. Эти принципы лежат в основе современных систем управления качеством:

1. Ориентация организации на потребителя (Customer-Focused Organization). Организации зависят от своих потребителей и, таким образом, должны понимать текущие и будущие потребности потребителей, удовлетворять их требования и стремиться превзойти их ожидания.
1. Лидерство (Leadership) Лидеры организаций обеспечивают единство назначения и направления организации. Они должны создать и поддерживать внутреннюю окружающую среду, в которой люди могут в полной мере участвовать в достижении стратегических целей организации.
2. Вовлечение персонала (Involvement of People) Люди составляют сущность организации на всех уровнях, и их полная вовлеченность способствует применению их способностей на благо организации.
3. Процессный подход (Process Approach) Желаемый результат достигается более эффективно, когда связанные ресурсы и деятельность управляются как процесс.
4. Системный подход к административному управлению (System Approach to

5. Management) Выявление, понимание и административное управление системой взаимосвязанных процессов для заданной стратегической цели повышает эффективность и результативность организации.
6. Непрерывное усовершенствование (Continual Improvement) Непрерывное усовершенствование должно быть постоянной стратегической целью организации.
7. Основанный на фактах подход к принятию решений (Factual Approach to Decision Making) Эффективные решения базируются на анализе данных и информации.
8. Взаимовыгодные отношения с поставщиками (Mutually beneficial supplier relationship)

Организация и ее поставщики взаимозависимы, и взаимовыгодные отношения повышают способность обоих производить ценности.

Модель технологической зрелости. Исследователи SEI пошли достаточно простым путем. Следуя TQM, оценку зрелости организаций они решили проводить на основе анализа выполняемых этой организацией процессов по разработке ПО. При этом считалось (а это – в духе ISO9000), что организация является тем более зрелой (более предсказуемой), чем более установленными являются применяемые процессы.

Первые исследования организаций с этих позиций показали, что организации находятся на разных уровнях зрелости – была проведена классификация этих уровней, разработаны методы оценки уровня организации и предложены способы повышения уровня зрелости организации. Все это вместе и составляет модель технологической зрелости организации, или модель зрелости ее технологических процессов.

Модель технологической зрелости

В соответствии с СММ модель технологической зрелости - это описание стадий эволюции, которые проходят организации-разработчики по мере того, как они (организации) определяют, реализуют, измеряют, контролируют и совершенствуют процессы создания ПО. Модель помогает выбрать адекватную стратегию усовершенствования процессов, предоставляет методическую основу для определения текущего уровня их совершенства и выявления проблем, критичных для качества разрабатываемого ПО [7,12].

Основу модели СММ составляют следующие фундаментальные понятия:

- Process (технология, технологический процесс, процесс) – последовательность шагов (действий), предпринимаемых с заданной целью. Более точно, процесс определяется так: Производственный

процесс - набор операций, методов, практик и преобразований, используемых разработчиками для создания и сопровождения ПО и связанных с ним продуктов (например, планов проекта, проектных документов, кодов, сценариев тестирования и руководств пользователя).

- Process Capability (продуктивность, совершенство технологии/процесса) – диапазон результатов, которые можно ожидать от организации, соблюдающей данный технологический процесс. Это понятие имеет отношение к будущим проектам, но базируется на фактических характеристиках технологии, достигнутых на предыдущих проектах.
- Process Performance (производительность технологии/процесса) – фактические результаты, достигнутые организацией, соблюдающей данную технологию/процесс. Это понятие ассоциируется с уже выполненными проектами. Таким образом, производительность фокусируется на достигаемых результатах, в то время как его продуктивность опирается на ожидаемые результаты.
- Process Maturity (зрелость технологии) - степень определенности, управляемости, наблюдаемости, контролируемости и эффективности процесса, технологии. Фактически, это индикатор полноты технологии и степени последовательности (настойчивости) организации в ее применении на всех проектах. Зрелость определяет потенциал дальнейшего роста совершенства технологии/процесса.

CMM. Пять уровней зрелости

Разработчики модели CMM (SEI) определили пять уровней технологической зрелости, по которым заказчики могут оценивать потенциальных поставщиков (претендентов на получение контракта), а поставщики могут совершенствовать процессы создания ПО. Каждому из уровней технологической зрелости внутри модели CMM дано следующее краткое определение:

1. Начальный (Initial). Технология разработки ПО характеризуется как произвольная (импровизированная), в некоторых случаях — даже хаотическая. Лишь некоторые процессы определены, успех всецело зависит от усилий отдельных сотрудников.
2. Повторяемый (Repeatable). Базовые процессы управления проектом ПО установлены для отслеживания стоимости, графика и функциональности выходного продукта. Необходимая дисциплина соблюдения установленных процессов имеет место и обеспечивает возможность повторения успеха предыдущих проектов в той же прикладной области.
3. Определенный (Defined). Управленческие и инженерные процессы задокументированы, стандартизованы и интегрированы в

унифицированную для всей организации технологию создания ПО. Каждый проект использует утвержденную, адаптированную к особенностям данного проекта, версию этой технологии.

4. Управляемый (Managed). Детальные метрики (объективные данные) о качестве исполнения процессов и выходной продукции собираются и накапливаются. Управление процессами и выходной продукцией осуществляется по количественным оценкам.
5. Оптимизируемый (Optimized). Совершенствование технологии создания ПО осуществляется непрерывно на основе количественной обратной связи от процессов и плотного внедрения инновационных идей.

СММ. Определение модели зрелости

Модель зрелости должна дать ответ на два вопроса:

- На каком уровне зрелости находится организация?
- Что надо делать, чтобы перейти на следующий уровень?

Модель зрелости СММ является гибкой – она не содержит четких и конкретных (формализованных) указаний на этот счет, а дает некоторую схему поиска ответов на поставленные вопросы и рекомендации по ее использованию. По мнению разработчиков, это расширяет применимость модели. Структура (схема) модели СММ содержит следующие основные элементы:

- *Группы ключевых процессов.* Каждый уровень зрелости содержит описание группы ключевых процессов, которые должны выполняться на этом уровне.
- *Цели.* Для каждого ключевого процесса определены цели, которые нужно достигнуть для перехода на следующий уровень. Цели (целевые установки):
 - служат критерием эффективной реализации группы ключевых процессов в организации
 - выражают объем, границы и смысл каждой группы ключевых процессов
 - после того, как цели будут реализованы на постоянной основе для всех проектов, можно будет сказать, что организация установила уровень продуктивности своего производственного процесса, характеризующийся данной группой ключевых процессов

Кроме того, определяется блок связанных работ, после выполнения которых достигаются цели, и круг проблем, которые необходимо решить для достижения следующего уровня зрелости.

- *Разделы.* Описания ключевых процессов организованы по разделам, которые представляют собой атрибуты, указывающие, являются ли эффективными, повторяемыми и устойчивыми реализация и установление групп ключевых процессов. Ниже перечислены пять основных разделов:
 - Обязательства по выполнению. Описывают действия, которые должна выполнить организация, чтобы обеспечить установление и стабильность процесса. Обязательства по выполнению обычно касаются установления организационных политик и поддержки со стороны высшего руководства.
 - Необходимые предпосылки. Описывают предварительные условия, которые должны выполняться в проекте или организации для компетентного внедрения производственного процесса, обычно касаются ресурсов, организационных структур и требуемого обучения.
 - Выполняемые операции. В разделе «Выполняемые операции» описаны роли и процедуры, необходимые для внедрения группы ключевых процессов. Выполняемые операции обычно включают в себя создание планов и реализацию процедур, выполнение и отслеживание работ, а также, по мере необходимости, выполнение корректирующих действий. Практики раздела «Выполняемые операции» описывают, что должно быть реализовано для получения продуктивного процесса. Все остальные практики вместе формируют базис, с помощью которого организация может внедрить практики, описанные в разделе «Выполняемые операции».
 - Измерения и анализ. Раздел «Измерения и анализ» описывает, что необходимо для измерения процесса и анализа результатов измерений. В этом разделе обычно приводятся примеры измерений, с помощью которых можно определить статус и эффективность выполняемых операций.
 - Проверка внедрения. В разделе «Проверка внедрения» описываются шаги, позволяющие убедиться в том, что операции выполняются в соответствии с установленным процессом. В этот раздел обычно входят проверки и аудиты со стороны руководства и работы по обеспечению качества ПО.
- Ключевые практики. Описание каждого раздела ключевых процессов выражается ключевыми практиками и подпрактиками, выполнение которых способствует достижению целей группы. Ключевые практики описывают инфраструктуру и операции, которые дают наибольший вклад в эффективное внедрение и установление группы ключевых процессов.

СММ. Группы ключевых процессов

Уровни зрелости включают следующие группы ключевых процессов:

- Начальный
 - Компетентность специалистов, самопожертвование и героизм
- Повторяемый
 - Управление требованиями. Планирование проекта ПО. Отслеживание и контроль проекта ПО. Управление субподрядом. Обеспечение качества ПО. Конфигурационное управление ПО
- Определенный
 - Фокус организации на процессах. Определение процессов в организации. Программа обучения. Интегральное управление ПО. Разработка программной продукции. Координация между группами. Коллегиальное рассмотрение (Peer Review).
- Управляемый
 - Количественное управление процессами. Менеджмент качества ПО.
- Оптимизируемый
 - Предупреждение дефектов. Управление изменениями в технологиях. Управление изменениями в процессах

СММ. Критерии оценки уровня зрелости

В СММ предлагаются следующие критерии оценки соответствия организации тому или иному уровню зрелости:

- Целевые установки группы ключевых процессов считаются удовлетворенными, если действующая в организации практика соответствует всем ключевым элементам практики СММ для данной области или их адекватному эквиваленту.
- Группа ключевых процессов считается удовлетворяющей соответствующему уровню, если все целевые установки СММ в данной области удовлетворены и не удовлетворяющей, если полностью не удовлетворена хотя бы одна ее целевая установка.
- Организация считается соответствующей уровню зрелости, если все ключевые области процессов этого и всех нижестоящих уровней удовлетворены и не считается соответствующей, если хотя бы одна ключевая область процессов этого или любого нижестоящего уровня не удовлетворяет СММ.

СММ. Резюме: СММ в тезисах

Схематично основные принципы СММ можно представить в виде следующих тезисов:

- Зрелость организации есть возможность выполнять сложные проекты
 - Зрелость организации определяется через зрелость ее технологических процессов
 - Можно выделить уровни зрелости организаций (процессов). В СММ их пять.
 - Модель зрелости – описание способа оценки уровня зрелости и путей перехода на следующий уровень
 - Модель зрелости описывается:
 - Ключевыми процессами, которые должны выполняться на каждом уровне зрелости
 - Каждый ключевой процесс описывается целями и набором разделов – атрибутов, определяющих различные аспекты выполнения процесса
 - Каждый атрибут описывается в виде ключевых практик – отдельных действий и условий, которые должны выполняться
 - Достижение уровня зрелости определяется по критерию:
 - уровень достигнут, если удовлетворены все ключевые процессы этого уровня
 - ключевой процесс удовлетворен, если достигнуты все его цели
 - цели процесса достигнуты, если выполняются все ключевые практики всех разделов или их аналоги

Представленная схема дает способ оценки уровня зрелости организации и определения путей перехода на следующий уровень.

Для оформления данной темы на основе представленного материала необходимо разработать модель работ в соответствии с вариантом задания. Оформить данный подраздел возможно в произвольном формате (таблицы, графики, иллюстрации и т.д.).

Развертывание и поставка программного обеспечения

Цикл работ по развертыванию и поставке ПО начинается в момент фиксации финальной версии ПО и подготовки всех спецификаций для конфигурирования системного программного обеспечения. Следующим шагом создаются сценарии автоматизации развертывания ПО и конфигураций

операционной среды именуемые ролью, которые содержат описание последовательности выполнения команд на целевом узле (виртуальной машине). По смысловой нагрузке каждая роль объединяет в себе набор задач, которые должны быть решены, чтобы отдельно взятый узел был сконфигурирован и готов к эксплуатации в соответствии со своим целевым назначением. Решение таких задач вручную является ресурсоемким процессом, который требует привлечения широкого круга специалистов и предполагает высокие временные затраты. Автоматизация данного процесса за счет специальных программных средств сокращает временные издержки на подготовку оборудования и ПО к работе.

Автоматизация развертывания и поставки ПО имеет ряд важных преимуществ при выполнении работ:

- удаленное размещение конфигураций операционной среды и удаленной установки ПО, которая не требует предварительной установки программ-агентов (клиентов), которые выполняют сценарии развертывания;
- позволяет оператору эксплуатировать систему развертывания, не владеющему навыками программирования;
- предоставление возможности удаленного подключения по стандартизированным протоколам, что позволяет взаимодействовать с различными операционными системами;
- предполагает повторное использование сценариев для типовых задач развертывания и конфигурирования, что существенно ускоряет процесс ее настройки.

В задачи автоматизации развертывания и поставки системного и специального ПО входят:

- базовая настройку целевых узлов, что реализуется за счет автоматической установки операционных систем на аппаратные средства АС;
- удаленное автоматизированное конфигурирование операционных сред под целевое назначение ПО;
- удаленное автоматизированное формирование виртуальной инфраструктуры под целевое назначение ПО;
- удаленное автоматизированное конфигурирование сетевой инфраструктуры для АС как для физических, так и виртуальных узлов;

- удаленное автоматизированное конфигурирование единого пространства пользователей в выделенном сегменте сети с учетом приоритета настройки контроллера домена;
- удаленное автоматизированное конфигурирование программных средств защиты информации;
- удаленное автоматизированное конфигурирование рабочих репозиториев для физических (хостовых) и виртуальных узлов под целевое назначение ПО;
- удаленное автоматизированное конфигурирование различных типов подключений к физическим и виртуальным узлам по широкому спектру протоколов;
- централизованное, распределенное, массовое развертывание типовых конфигураций для операционных сред одновременно на нескольких целевых узлах АС.

Предполагаемый состав системы автоматизированного развертывания и конфигурирования системного и специального ПО входят следующие компоненты:

- ядро, являющееся управляющим компонентом системы;
- служебная часть автоматизации передачи учетных данных удаленному узлу для установления соединения;
- пользовательский консольный интерфейс для возможности удаленного подключения к системе;
- комплект сценариев создания, управления, миграции и резервирования рабочих репозиториев;
- комплект сценариев загрузки образов различных операционных систем и различных файловых ресурсов на удаленные целевые узлы;
- комплект сценариев управления виртуальными машинами на удаленных физических целевых узлах;
- комплект сценариев автоматической установки операционной системы на целевые узлы АС.

Ключевым элементом автоматизации задачи автоматизации развертывания и поставки ПО являются программные средства управления конфигурациями. Основная идея программных средств управления конфигурации одного или нескольких удаленных хостов (виртуальных машин) заключается в отправке команд или наборов последовательных инструкций (сценариев), подключаясь к ним по унифицированным протоколам в зависимости от типа операционной системы целевого узла.

Данные сценарии состоят из одной или более команд, которые описываются с помощью функциональности модуля программных средств

автоматизации развертывания или сторонних модулей, которые могут потребоваться в специфических ситуациях. Сценарии – последовательные наборы команд, в которых могут быть описаны:

- переменные для хранения IP-адресов одного или множества целевых узлов;
- проверки условий: если условие не выполняется, определенные команды могут пропускаться;
- создание папок в файловой системе целевого узла;
- удаленная загрузка файлов конфигурации и их подмена или обновление;
- наборы команд для их удаленного выполнения на целевом узле;
- установка и конфигурирование системного программного обеспечения индивидуально для каждого из узлов;
- удаленная остановка или перезагрузка целевого узла.

В рамках темы автоматизации развертывания и поставки ПО необходимо описать порядок развертывания, привести список программных средств развертывания и провести сравнительный анализ программных средств необходимых для реализации данного процесса.

Оформление приложений к пояснительной записке технического проекта

Материалы художественно-конструкторской проработки

Материалы художественно-конструкторской проработки, не являющиеся конструкторской документацией включаются в данный раздел. Примером материалов художественно-конструкторской проработки может являться эскизный рисунок изделия или медиа-материалы связанные с ним.

Перечень работ по проекту

Перечень работ по проекту оформляется в соответствии с этапами ЖЦ изделия (АС или ПО) с указанием предполагаемого результата и сроков выполнения в табличном виде. Пример приведен в таблице ниже.

Таблица №3. Пример перечня работ по проекту

№ этапа	Содержание работ	Ожидаемый результат	Отчетные документы	Сроки выполнения работ	
				Дата начала	Дата окончания
1	Разработка конструктивных решений изделия и его основных составных частей	Комплект документов ТП	ТП	С даты заключения договора, но не ранее 01.01.2023	01.06.2023
2	Выполнение необходимых расчетов, в том числе подтверждающих технико-экономические показатели, установленные ТЗ	Комплект документов ТП
3	Разработка и обоснование технических решений, обеспечивающих показатели надежности, установленные ТЗ и предшествующими стадиями разработки (если эти стадии разрабатывались)	Комплект документов ТП	Технический акт
4	Разработка и обоснование технических решений, обеспечивающих показатели надежности, установленные ТЗ и предшествующими стадиями разработки (если эти стадии разрабатывались)	Комплект документов ТП
5

Перечень работ является список работ верхнего уровня (без детализации, подэтапов). Детализация работ уточняется в следующем разделе «Уточнение сетевого плана-графика».

Уточнение сетевого плана графика

Сетевой план-график формируется на основе этапов работ с их детализацией и сроков реализации каждого из них. Форма представления может быть по образцу таблицы ниже.

Таблица №4. Пример сетевого плана графика

Вид работ	2022 год		2023 год						
	ноябрь	декабрь	январь	февраль	март	апрель	май	июнь	июль
Этап 1									
Эт. 1.1									
Эт. 1.2									
...									
Этап 2									
...			

Перечень литературы и документов, используемых при разработке пояснительной записке к техническому проекту

Нюансы оформления научной работы прописаны в едином Межгосударственном стандарте – ГОСТ 7.1-2003 и ГОСТ 7.0.100-2018. На основании данного нормативного документа должен оформляться каждый список литературы.

Рекомендации по оформлению пояснительной записке к техническому проекту

Несмотря на то, что основную роль в разработке ТП играет содержание невозможно игнорировать вопрос оформления. Вникнуть в содержание при неграмотном оформлении становится сложнее и, например, процесс согласования с заказчиком технических решений может затягиваться и уводить от изначального замысла, что может негативно сказаться на результате в целом. Поэтому в технических документах принято строгое содержание и ограниченная терминология, без художественных выражений и оборотов. Заказчик должен сосредоточиться на сути, а не художественных оборотах [5,6].

Приведем несколько принципов оформления ТП.

1. В ТП обязательно необходимо использовать стили и кроме как для подчеркивания или выделения внутри абзаца не менять настройки шрифта и абзаца только для одного фрагмента. Если данные настройки необходимо изменить, то изменяют конфигурацию стиля.
2. Второй обязательный момент – автособираемое оглавление, список терминов и сокращений, титульная страница. Желательно также приводить список версий документа, список изменений: очень легко потом отследить, в какие даты была отправлена та или иная версия.
3. Каждое отдельное положение должно быть изложено в отдельном пронумерованном абзаце.
4. Расстановка ссылок. При прочтении абзаца, где упоминается какая-то функция или требование, непросто понять, из этого же документа или из другого. Если из этого, – то в каком разделе. Поэтому необходимо ссылаться на другие разделы, если они упоминаются в текущем тексте. Естественно, ссылки должны быть автоматическими.
5. Также, необходимо ознакомиться с правилами оформления учебных работ на сайте ГУАП (<https://guap.ru/standart/doc>).
6. ТП оформляется как отдельный документ и после интегрируется в итоговый отчет о курсовой работе.
7. Оформление ТП проводится в соответствии с единой системой программной документации.

Заметим, что не во всех ГОСТ требуется наличие рамки, однако в данной курсовой работе ее наличие обязательно.

Подготовка к защите курсового проекта (курсовой работы)

Каждый студент по результатам проведенной работы должен подготовить устный доклад на 5-8 минут и презентационные материалы в объеме 10-12 слайдов по структуре пояснительной записки ТП, где должна быть изложена основная суть разработки.

Каждый студент должен быть готов ответить на следующие вопросы.

1. Какие проблемы в предметной области были идентифицированы в процессе разработки ТП?
2. В чем состоит обоснование классификации требований, полученной в результате анализа?
3. Какие ограничения предметной области учитываются?
4. Какое назначение языков моделирования при проектировании моделей предметной области?
5. По каким принципам проводилась разработка программной архитектуры?
6. В чем заключается обоснование выбранного варианта архитектуры?
7. Какие требования использовались для разработки архитектуры?
8. По какому принципу формировался состав модулей архитектуры программной системы?
9. Опишите разработанную модель работ по проекту.
10. Перечислите критерии выбора программных средств развертывания и конфигурирования.

Список литературы

1. IEEE Swebok V 3.0 Guide to the Software Engineering. Body of Knowledge. A Project of the IEEE Computer Society. Inetrnet resource. URL: www.computer.org (дата обращения: 10.03.2022).
2. *Басс Л., Клементс П., Кацман Р.* Архитектура программного обеспечения на практике. 2-е издание. – СПб.: Питер, 2006. – 575 с.: ил. ISBN 5-469-00494-5.
3. *Брукс Ф.* Мифический человеко-месяц или как создаются программы. – Пер. с англ. – СПб.: Символ-Плюс, 2007. – 304 с.: ил. ISBN 5-93286-005-7.
4. *Вигерс К., Битти Д.* Разработка требований к программному обеспечению/Пер. с англ. – М.: Издательство "БХВ-Петербург", 2014. – 736 с.: ил.
5. ГОСТ 2.103-68 Единая система конструкторской документации. Стадии разработки.
6. ГОСТ 2.120-73 Единая система конструкторской документации. Технический проект.
7. *Карпенко С.Н.* Введение в программную инженерию. Учебно-методические материалы по программе повышения квалификации «Информационные технологии и компьютерное моделирование в прикладной математике». Нижний Новгород, 2007, 103 с.
8. *Купер А., Рейман Р., Кронин Д.* Алан Купер об интерфейсе. Основы проектирования и взаимодействия. – Пер. с англ. – СПб.: Символ-Плюс, 2009. – 688 с., ил.
9. *Леффингуэл, Дин, Уидриг, Дон.* Принципы работы с требованиями к программному обеспечению. Унифицированный подход.: Пер. с англ. – М.: Издательский дом «Вильямс», 2002. – 448 с.: ил. – Парал. Тит. Англ. ISBN 5-8459-0275-4 (рус.).
10. *Липаев В.В.* Программная инженерия. Методологические основы [Текст]: Учеб. / В.В. Липаев ; Гос. Ун-т – Высшая школа экономики. – М.: ТЕИС, 2006. – 608 с. – 1000 экз. – ISBN 5-7598-0424-3 (в пер.).
11. *Маглинец Ю.А.* Разработка информационных систем. Часть 1, Структурные методы. – Красноярск.: Кларитеанум, 2004. – 120 с.
12. *Мацяшек Л. А.* Практическая программная инженерия на основе учебного примера / Л. А. Мацяшек, Б. Л. Лионг ; пер. с англ. – 4-е изд., электрон. – М. : Лаборатория знаний, 2020. – 959 с. – (Программисту). – Систем. требования: Adobe Reader XI ; экран 10". – Загл. с титул. экрана. – Текст : электронный. ISBN 978-5-00101-783-7.
13. *Микони С. В., Соколов Б. В. Юсупов Р. М.* Квалиметрия моделей и полимодельных комплексов : монография С. В. Микони, Б. В. Соколов, Р. М. Юсупов. – М. : РАН, 2018. – 314 с. ISBN 978-5-907036-32-1. DOI:10.31857/S9785907036321000001.

14. *Назаров С. В.* Архитектура и проектирование программных систем [Электронный ресурс] : Монография / С.В. Назаров. - М.: НИЦ ИНФРАМ, 2014. - 351 с. – Режим доступа: <http://znanium.com/bookread2.php?book=542562> – Загл. с экрана.
15. *Орлик С.* Программная инженерия. Программные требования. Перевод Swebok 2004. 2010 г. Интернет-ресурс. URL: <http://swebok.sorlik.ru>. Дата обращения: 10.01.2022.

Список сокращений

АС	Автоматизированная система
ЖЦ	Жизненный цикл
ОКР	Опытно-конструкторская работа
ПО	Программное обеспечение
РКД	Рабочая конструкторская документация
ТЗ	Техническое задание
ТП	Технический проект
ТТЗ	Тактико-техническое задание

Список иллюстраций

<i>Рис. 1 Место актуальности в структуре работы</i>	<i>9</i>
<i>Рис. 2 Место цели в структуре работы.....</i>	<i>11</i>
<i>Рис. 3 Уровни требований по Вигерсу.....</i>	<i>18</i>
<i>Рис. 4 Пример контекстной диаграммы ИС для заказа химикатов</i>	<i>27</i>
<i>Рис. 5 Диаграмма перехода состояний для экземпляра задачи.....</i>	<i>32</i>
<i>Рис. 6 Обобщенный пример цепочки.....</i>	<i>35</i>
<i>Рис. 7 Физическое представление системы управления воздушным движением</i>	<i>40</i>
<i>Рис. 8 Клиент-серверное представление архитектуры ПО системы управления воздушным движением.....</i>	<i>41</i>
<i>Рис. 9 Многоуровневое представление ПО системы управления воздушным движением..</i>	<i>42</i>
<i>Рис. 10 Представление «компонент-соединитель» отказоустойчивого функционирования системы управления воздушным движением</i>	<i>43</i>
<i>Рис. 11 Сравнительный анализ CASE-систем в задаче автоматизации информационно-аналитической поддержки ЖЦ сложных организационно-технических объектов (СИАП ЖЦ СОТО как пример разрабатываемого изделия).....</i>	<i>46</i>

Варианты заданий для выполнения лабораторных работ

Номер варианта задания	Задание на лабораторную работу
1	Разработка программного обеспечения для автоматизированной/информационной системы железной дороги
2	Разработка программного обеспечения для автоматизированной/информационной системы авиакомпании
3	Разработка программного обеспечения для автоматизированной/информационной системы аэропорта
4	Разработка программного обеспечения для автоматизированной/информационной системы морского порта
5	Разработка программного обеспечения для автоматизированной/информационной системы автобусного вокзала
6	Разработка программного обеспечения для автоматизированной/информационной системы школы
7	Разработка программного обеспечения для автоматизированной/информационной системы библиотеки
8	Разработка программного обеспечения для автоматизированной/информационной системы университета
9	Разработка программного обеспечения для автоматизированной/информационной системы службы занятости
10	Разработка программного обеспечения для автоматизированной/информационной системы службы социальной защиты
11	Разработка программного обеспечения для автоматизированной/информационной системы поликлиники
12	Разработка программного обеспечения для автоматизированной/информационной системы обязательного медицинского страхования
13	Разработка программного обеспечения для автоматизированной/информационной системы пенсионного фонда
14	Разработка программного обеспечения для автоматизированной/информационной системы выставочного комплекса
15	Разработка программного обеспечения для автоматизированной/информационной системы для организации НИОКР

16	Разработка программного обеспечения для автоматизированной/информационной системы издательства
17	Разработка программного обеспечения для автоматизированной/информационной системы редакции газеты
18	Разработка программного обеспечения для автоматизированной/информационной системы типографии
19	Разработка программного обеспечения для автоматизированной/информационной системы гостиницы
20	Разработка программного обеспечения для автоматизированной/информационной системы киноцентра
21	Разработка программного обеспечения для автоматизированной/информационной системы фирмы по прокату автомобилей
22	Разработка программного обеспечения для автоматизированной/информационной системы букмекерской фирмы
23	Разработка программного обеспечения для автоматизированной/информационной системы фондовой биржи
24	Разработка программного обеспечения для автоматизированной/информационной системы банка
25	Разработка программного обеспечения для автоматизированной/информационной системы лизинговой компании
26	Разработка программного обеспечения для автоматизированной/информационной системы туристического агентства
27	Разработка программного обеспечения для автоматизированной/информационной системы фильмотеки
28	Разработка программного обеспечения для автоматизированной/информационной системы агентства недвижимости
29	Разработка программного обеспечения для автоматизированной/информационной системы страховой организации
30	Разработка программного обеспечения для автоматизированной/информационной системы автошколы
31	Разработка программного обеспечения для автоматизированной/информационной системы оператора связи
32	Разработка программного обеспечения для автоматизированной/информационной системы автосервиса
33	Разработка программного обеспечения для автоматизированной/информационной системы для оказания госуслуг

34	Разработка программного обеспечения для автоматизированной/информационной системы фирмы по сборке и продаже компьютеров и комплектующих
35	Разработка программного обеспечения для автоматизированной/информационной системы транспортной фирмы
36	Разработка программного обеспечения для автоматизированной/информационной системы супермаркета
37	Разработка программного обеспечения для автоматизированной/информационной системы книжного магазина
38	Разработка программного обеспечения для автоматизированной/информационной системы ломбарда
39	Разработка программного обеспечения для автоматизированной/информационной системы ГИБДД
40	Разработка программного обеспечения для автоматизированной/информационной системы спортивного клуба
41	Разработка программного обеспечения для автоматизированной/информационной системы интернет-провайдера
42	Разработка программного обеспечения для автоматизированной/информационной системы интернет-магазина
43	Разработка программного обеспечения для автоматизированной/информационной системы интернет-аукциона
44	Разработка программного обеспечения для автоматизированной/информационной системы почтовой службы
45	Разработка программного обеспечения для автоматизированной/информационной системы предприятия ЖКХ
46	Разработка программного обеспечения для автоматизированной/информационной системы рекламного агентства
47	Разработка программного обеспечения для автоматизированной/информационной системы курьерской фирмы
48	Разработка программного обеспечения для автоматизированной/информационной системы ресторанного комплекса
49	Разработка программного обеспечения для автоматизированной/информационной системы службы такси
50	Разработка программного обеспечения для автоматизированной/информационной системы службы технической поддержки
51	Разработка программного обеспечения для автоматизированной/информационной системы <свой вариант> (необходимо сформулировать тему)