

В.А. Гончаренко, С.А. Лепешкин, Л.Ю. Шарабаева, А.А. Шульгин

# Моделирование и проектирование систем

*Учебное издание «Моделирование и проектирование систем.*

*Часть 2. Проектирование систем» утверждено в качестве учебного пособия и рекомендовано ученым советом факультета автоматизированных систем управления войсками Военно-космической академии имени А.Ф. Можайского для обучающихся по основным профессиональным образовательным программам высшего образования – программам специалитета по специальности «Применение и эксплуатация автоматизированных систем специального назначения»,  
протокол от 25 мая 2017 года, № \_*

Часть 2

## Проектирование систем



Санкт-Петербург  
2017

Рецензенты:  
доктор технических наук, профессор **Ю.С. Мануйлов**,  
доктор технических наук, профессор **А.Г. Басыров**

**Моделирование и проектирование систем. Ч. 2. Моделирование систем:** учебное пособие / В.А. Гончаренко, С.А. Лепешкин, Л.Ю. Шарабаева, А.А. Шульгин. – СПб.: ВКА имени А.Ф. Можайского, 2017. – 124 с.

В предлагаемом учебном пособии изложен материал второго раздела «Проектирование систем» учебной дисциплины «Моделирование и проектирование систем», ориентированный на обучающихся по основным профессиональным образовательным программам высшего образования.

Рассматриваются методологические основы проектирования автоматизированных информационных систем, стандарты и другие нормативные документы в области проектирования, основы проектирования программного и аппаратного обеспечения информационных систем. Подробно изложены современные технологии проектирования информационных систем (каноническое, автоматизированное, типовое), а также принципы и методы проектирования компьютерных сетей, как важнейшей инфраструктурной составляющей автоматизированных информационных систем.

Подразд. 1.1, 1.4, разд. 3 и заключение разработал и подготовил к изданию кандидат технических наук доцент В.А. Гончаренко, предисловие, подразд. 2.2, 2.3 – кандидат технических наук доцент С.А. Лепешкин, подразд. 1.2, 1.3 – кандидат физико-математических наук доцент Л.Ю. Шарабаева, подразд. 2.1, 3.3 – кандидат технических наук А.А. Шульгин.

© ВКА имени А.Ф. Можайского,

---

Подписано к печ. **01.09.2017**  
Гарнитура Times New Roman  
Уч.-печ. л. 15,50

Формат печатного листа 445×300/8  
Уч.-изд. л. 7,50  
Заказ **3313**

Бесплатно

---

Типография ВКА имени А.Ф. Можайского

## ОГЛАВЛЕНИЕ

Список сокращений .....	5
Предисловие .....	6
1. Основы системного проектирования автоматизированных систем .....	8
1.1. Классификация и архитектура проектируемых систем .....	8
1.1.1. Понятие автоматизированной информационной системы .....	8
1.1.2. Классификации автоматизированных информационных систем .....	10
1.1.3. Архитектура автоматизированных информационных систем .....	14
1.2. Жизненный цикл и стандарты проектирования систем .....	17
1.2.1. Понятие и структура проектирования .....	17
1.2.2. Процессы и модели жизненного цикла систем .....	19
1.2.3. Стадии проектирования систем .....	26
1.3. Требования, предъявляемые к проектируемым объектам .....	30
1.3.1. Основные виды требований к автоматизированным системам .....	30
1.3.2. Проектные и производственные требования .....	32
1.3.3. Эксплуатационные требования .....	36
1.4. Технологии проектирования информационных систем .....	40
1.4.1. Классы технологий проектирования ИС .....	40
1.4.2. Каноническое проектирование ИС .....	42
1.4.3. Типовое проектирование ИС .....	48
1.4.4. Управление процессом проектирования .....	51
2. Автоматизированное проектирование информационных систем .....	53
2.1. Автоматизация проектирования ИС на основе CASE-технологии .....	53
2.1.1. Методологии моделирования систем семейства IDEF .....	53
2.1.2. Основные понятия и классификация CASE-технологий .....	54
2.1.3. Прототипная технология быстрой разработки приложений RAD ....	59
2.1.4. CASE-технология RUP .....	60
2.2. Функционально-ориентированное проектирование информационных систем .....	61
2.1.1. Методология структурно-функционального моделирования .....	61
2.1.2. Стандарт моделирования IDEF0 .....	63
2.1.3. Моделирование потоков данных в нотации DFD .....	70
2.3. Объектно-ориентированное проектирование информационных систем на основе языка UML .....	76

2.3.1. Основы унифицированного языка моделирования UML .....	76
2.3.2. Проектирование логической модели ИС и моделей баз данных .....	77
2.3.3. Проектирование физической модели ИС .....	82
3. Проектирование сетевой компьютерной инфраструктуры информационных систем .....	84
3.1. Основы проектирования компьютерных сетей.....	84
3.1.1. Этапы проектирования компьютерных сетей .....	84
3.1.2. Теоретические основы проектирования компьютерных сетей .....	87
3.1.3. Анализ требований к компьютерным сетям .....	89
3.1.4. Основные модели и характеристики проектируемых компьютерных сетей	91
3.2. Проектирование локальных и глобальных сетей.....	95
3.2.1. Общие вопросы проектирования локальных и глобальных сетей.....	95
3.2.2. Планирование сетевой архитектуры .....	96
3.2.3. Принципы проектирования локальных сетей .....	100
3.2.4. Принципы проектирования глобальных сетей .....	103
3.3. Порядок разработки и оформления результатов проектирования сетевой инфраструктуры .....	105
3.3.1. Общие требования стандартов по проектированию сетевой инфраструктуры .....	105
3.3.2. Проектирование горизонтальной подсистемы локальной сети .....	108
3.3.3. Проектирование магистральной подсистемы локальной сети.....	111
3.3.4. Планирование телекоммуникационных пространств и помещений в проекте компьютерной сети.....	113
Заключение.....	119
Список литературы .....	122

## СПИСОК СОКРАЩЕНИЙ

АИС	–	автоматизированная информационная система
АОС	–	автоматизированная обучающая система
АРМ	–	автоматизированное рабочее место
АС	–	автоматизированная система
АСУ	–	автоматизированная система управления
БД	–	база данных
ВВСТ	–	вооружение, военная и специальная техника
ВКР	–	выпускная квалификационная работа
ВС	–	вычислительная система
ГВС	–	глобальная вычислительная сеть
ГОСТ	–	государственный стандарт
ГРП	–	главный распределительный пункт
ЕИП	–	единое информационное пространство
ЖЦ	–	жизненный цикл
ИБП	–	источник бесперебойного питания
ИВС	–	информационно-вычислительная система
ИКТ	–	информационно-коммуникационная технология
ИС	–	информационная система
ИТ	–	информационная технология
КПД	–	коэффициент полезного действия
ЛВС	–	локальная вычислительная сеть
НИР		научно-исследовательская работа
ОКР	–	опытно-конструкторская работа
ОС	–	операционная система
ПО	–	программное обеспечение
ППП	–	пакет прикладных программ
СВТ	–	средство вычислительной техники
СКС	–	структурированная кабельная система
СУБД	–	система управления базами данных
ТЗ	–	техническое задание
ТПР	–	типовое проектное решение
ТТХ	–	тактико-технические характеристики
ЭВМ	–	электронная вычислительная машина

## ПРЕДИСЛОВИЕ

Проектирование сложных объектов основано на применении идей и принципов, изложенных в ряде теорий и подходов. Наиболее общим подходом является *системный подход*, идеями которого пронизаны различные методики проектирования сложных систем [30, 43, 53].

Основной общий принцип системного подхода заключается в рассмотрении частей исследуемого явления или сложной системы с учетом их взаимодействия. Системный подход включает в себя выявление структуры системы, типизацию связей, определение атрибутов, анализ влияния внешней среды, формирование модели системы, исследование модели и, возможно, *оптимизации* ее структуры и функционирования.

Автоматизированные системы (АС) относятся к числу наиболее сложных современных искусственных систем [43]. Их проектирование и сопровождение невозможны без системного подхода.

Середина прошлого столетия ознаменовалась началом активного развития информационных технологий. Прежде всего, военные ведомства и передовые предприятия многих стран понимают важность и ценность создания и развития автоматизированных информационных систем (АИС). С появлением вычислительной техники обработка больших объемов информации и автоматизация основных производственных процессов и органов управления на всех уровнях становятся наиважнейшей задачей для обеспечения военного превосходства наиболее развитых государств и конкурентных преимуществ коммерческих компаний. Разработчики национальных и крупномасштабных АИС стали первыми осознавать необходимость создания специальных средств проектирования и моделирования процессов деятельности, позволяющими сделать их работу более эффективной и сократить не только сроки создания АИС, но минимизировать ошибки [38]. Ошибки и неточности встречаются постоянно, чем раньше они диагностируются и локализуются, тем меньше стоимость переделки. Известно, что стоимость выявления и устранения ошибки на стадии проектирования обходится дороже в два раза, на стадии тестирования АИС – в десять раз, на стадии эксплуатации – в сто раз, чем на стадии анализа процессов деятельности и разработки технического задания. При создании сложных АС зачастую очень трудно понять требования персонала заказчика. Они могут быть сформулированы некорректно, а в процессе анализа конкретных процессов деятельности даже измениться. Поэтому появление методологий современного проектирования и моделирования АС было насущной задачей, над которой работали специалисты разных стран.

В современном обществе, получившем название «информационное», стратегическим ресурсом становится *информация*, а *информационно-коммуникационные*

*технологии* из вспомогательного инструмента превратились в критический фактор эффективности и развития организации. Большие объемы постоянно изменяющейся информации необходимо оперативно анализировать и принимать верные управленческие решения. Успешность деятельности любой организации в серьезной степени зависит от уровня развития ИКТ, скорости и качества обработки информации, обоснованности и взвешенности принимаемых решений.

Автоматизация всех типов информационных процессов реализуется в рамках АИС как в информационном, так и в информационно-управляющем режимах. Требуется постоянная серьезная работа не только ИТ-специалистов, но и руководителей верхнего уровня по согласованию или точнее – синхронизации всех усилий по стратегическому развитию организации и её информационных систем.

Поэтому процесс проектирования АИС в настоящее время становится обязательным. Если этот процесс не впервые осуществляется организацией, то термин проектирование приравнивается к понятию развития информационной системы. Этим объясняется бурное развитие технологий проектирования информационных систем в последние годы. Прежде всего, создание CASE-технологий, которые во много раз сокращают сроки проектирования ИС, позволяют организовать одновременную коллективную работу, оперативно вносить изменения и быстро реагировать на изменение обстановки на предприятии.

Проектирование и развитие АИС невозможно без знания основных методологий и программных средств, позволяющих в кратчайшие сроки и без ошибок управлять этими процессами. Организации получают колоссальные конкурентные преимущества, если уровень развития информационных систем соответствует уровню развития предприятия. Прежде всего, основные процессы деятельности становятся прозрачными и понятными для контроля и управления, исключаются ошибки и связанные с ними потери времени и средств.

Актуальность и важность раздела «Проектирование систем» в структуре дисциплины «Моделирование и проектирование систем» определяется необходимостью изучения теоретических положений, связанных с нормативно - технической документацией на разработку и проектирование систем вообще, а также получения практических навыков разработки основных проектных документов для информационных систем, в частности. Рассматриваются вопросы моделирования и анализа процессов деятельности, применения современных CASE-средств.

Настоящее учебное пособие содержит полный перечень теоретических сведений по разделу «Проектирование систем» дисциплины «Моделирование и проектирование систем».

# 1. ОСНОВЫ СИСТЕМНОГО ПРОЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СИСТЕМ

## 1.1. Классификация и архитектура проектируемых систем

### 1.1.1. Понятие автоматизированной информационной системы

Методологическую основу проектирования сложных организационно-технических систем составляет системный подход, в соответствии с которым любая система представляет собой совокупность взаимосвязанных объектов, функционирующих совместно для достижения общей цели.

В современной литературе по проектированию систем наибольшее количество публикаций отводится так называемым *автоматизированным системам (АС)* и *информационным системам (ИС)*. Разберем подробнее эти понятия.

Под **автоматизированной системой** понимают совокупность управляемого объекта и автоматических управляющих устройств, в которых часть функций управления выполняет человек. В зависимости от объекта обработки данные системы делят на *автоматизированные производственные системы* (объект управления – материальные и энергетические ресурсы) и *автоматизированные информационные системы* (объект управления – информация).

**Автоматизированная производственная система** представляет собой организационно-техническую систему, состоящую из средств автоматизации определенного вида или нескольких видов деятельности людей, и персонала, осуществляющего эту деятельность.

Определение **автоматизированной информационной системы**, содержанием работы которой является переработка информации, дано в ГОСТ 34.002-90 [5]:

**Автоматизированная система** – система, состоящая из *персонала* и *комплекса средств автоматизации* его деятельности, реализующая *информационную технологию* выполнения установленных функций.

Напомним, приведенное в [1] определение информационных технологий.

**Информационные технологии (ИТ)** – процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов.

Современный этап развития информационного общества предполагает использование ИТ во взаимосвязи с телекоммуникационными каналами и средствами связи, то есть **информационно-коммуникационных технологий (ИКТ)**.

Понятие «*информационной системы*» имеет несколько трактовок в зависимости от контекста. В Федеральном законе РФ от 27.07.2006 г. N 149-ФЗ «Об информации, информационных технологиях и о защите информации» определено:

**Информационная система** – совокупность содержащейся в базах данных информации и обеспечивающих ее обработку *информационных технологий* и *технических средств* [1].



В этом определении ИС рассматривается, как совокупность программно-аппаратного обеспечения, информационных технологий и данных (рис. 1.1).



Рис. 1.1. Структура ИС

Второй подход к трактовке ИС базируется на принципах системологии и в определении системы учитывает не только ее внутреннюю *архитектуру* (состав элементов и подсистем), но и ее отношения со *средой* (принцип целеполагания).

Под *системой* понимают любой объект, который одновременно рассматривается и как единое целое, и как объединенная в интересах достижения поставленных целей совокупность разнородных элементов. Добавление к понятию «система» слова «информационная» отражает цель ее создания и функционирования: обеспечение сбора, хранения, обработки, поиска, выдачи информации, необходимой в процессе получения решений задач из любой предметной области. Современное понимание ИС предполагает использование в качестве основных средств переработки информации *ИКТ*, различных типов современных *компьютеров*, а также *персонала*, взаимодействующего с компьютерами и телекоммуникациями. Таким образом, можно дать следующее определение ИС:

**Информационная система** – взаимосвязанная совокупность *средств, методов и персонала*, используемых для сбора, хранения, обработки, поиска и выдачи информации в интересах достижения поставленной цели.

В ГОСТ РВ 51987-02 под ИС подразумевается *автоматизированная система, результатом функционирования которой является представление выходной информации для последующего использования*, т.е. ИС – разновидность АС.

В более узкой трактовке ИС рассматривается в деятельности организаций, как *программное обеспечение*, реализующее деловую стратегию организации.

Таким образом, понятие «*информационная система*» в зависимости от контекста использования может означать разновидность, часть или компонент (подсистему) АС, либо быть синонимом понятия «*автоматизированная система*».

Можно представить базовую модель АИС (рис. 1.2), включающую пять видов основных ресурсов и обобщенную схему деятельности любой АИС. Важно отметить, что создание АИС не сводимо к разработке программно-аппаратного решения. Это, прежде всего деятельность по изменению деятельности всей организации, в которой проводится автоматизация процессов.

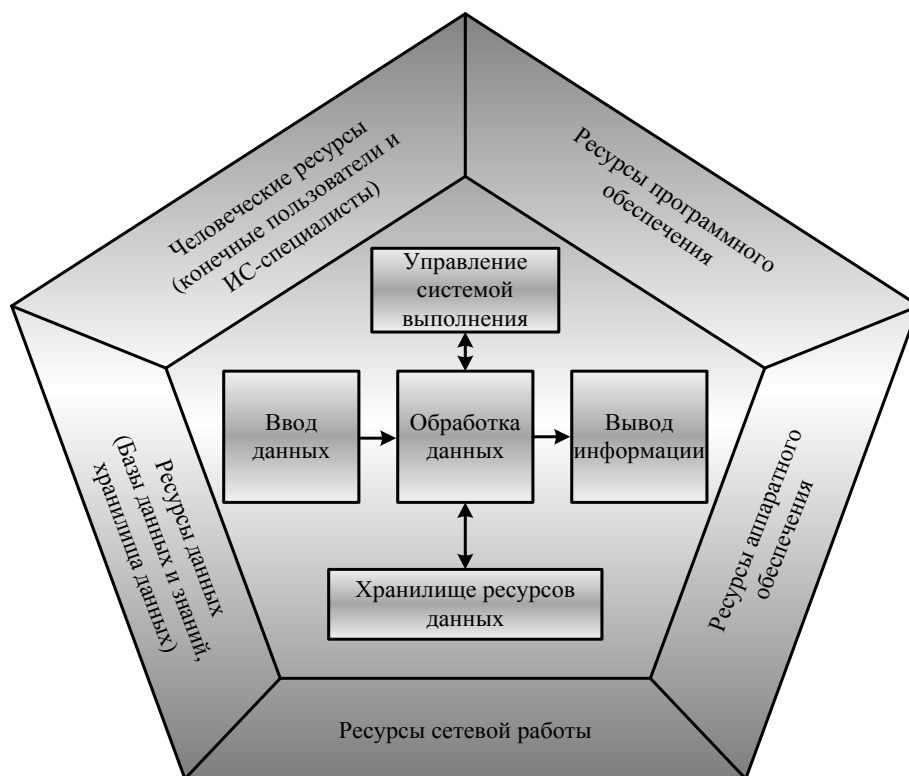


Рис. 1.2. Базовая модель АИС

### 1.1.2. Классификации автоматизированных информационных систем

В научно-методической литературе предлагаются различные варианты классификации ИС [26, 32, 42]. Рассмотрим классификацию по следующим признакам: степень автоматизации, режим работы, распределение вычислительных ресурсов, характер информации, сфера применения, масштаб.

В зависимости от **степени автоматизации** информационных процессов АИС делят на собственно автоматизированные и автоматические.

**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» вкладывается обязательно понятие **автоматизированной системы**.

В **автоматических** ИС автоматизация является полной, т.е. все операции по переработке информации выполняются без участия человека либо с его эпизодическим участием в целях контроля.

По **режиму работы** ИС делятся на пакетные, запросно-ответные, диалоговые и реального времени.

**Пакетные** ИС работают в пакетном режиме: вначале данные накапливаются и формируется пакет данных, а затем пакет последовательно обрабатывается рядом программ (например, системы информационного обслуживания, учебные системы). Недостаток этого режима – низкая оперативность принятия решений и обособленность пользователя от системы.

**Запросно-ответные** ИС работают в режиме запрос пользователя – ответ системы (например, системы продажи билетов, информационно-поисковые системы, библиотечные системы).

**Диалоговые** (интерактивные) ИС работают в режиме обмена сообщениями между пользователями и системой (например, САПР, АСНИ, обучающие системы). Этот режим особенно удобен, когда пользователь может выбирать перспективные варианты из числа предлагаемых системой.

Системы **реального времени** используются для управления быстропротекающими процессами и непрерывными технологическими процессами (например, испытательными стендами, подвижными объектами).

По способу распределения вычислительных ресурсов ИС делятся на локальные и распределенные.

**Локальные** ИС используют одну ЭВМ, а в **распределенных** ИС взаимодействуют несколько ЭВМ, связанных компьютерной сетью. Отдельные узлы сети обычно территориально удалены друг от друга, решают разные задачи, но используют общую информационную базу.

По характеру использования информации АИС делятся на информационно-поисковые, информационно-аналитические и информационно-решающие.

**Информационно-поисковые** системы производят ввод, систематизацию, хранение, выдачу информации по запросу пользователя без сложных преобразований данных. Например, информационно-поисковая система в библиотеке, в железнодорожных и авиакассах продажи билетов. Подразделяются на *фактографические* и *документографические*.

**Информационно-аналитические** системы предназначены для аналитической обработки данных с использованием баз знаний и экспертных систем.

**Информационно-решающие** системы осуществляют все операции переработки информации по определенному алгоритму. По степени воздействия выработанной результатной информации на процесс принятия решений среди них выделяют два класса систем: *управляющие* и *советующие*. *Управляющие АИС* вырабатывают информацию, на основании которой человек принимает решение (например, АСУП). Для этих систем характерны: тип задач расчетного характера и обработка больших объемов данных. *Советующие АИС* вырабатывают информацию, которая принимается человеком к сведению и не превращается немедленно в серию конкретных действий. Эти системы обладают более высокой степенью интеллекта, так как для них характерна обработка знаний, а не данных.

По сфере применения различают следующие классы АИС:

- автоматизированные системы управления (АСУ):
  - АС управления предприятием (АСУП);
  - АС управления технологическими процессами (АСУ ТП);

- системы поддержки принятия решений (СППР);
- автоматизированные информационно-вычислительные системы (АИВС);
  - информационно-расчетные системы (ИРС);
  - системы автоматизированного проектирования (САПР);
  - АС научных исследований (АСНИ);
- автоматизированные обучающие системы (АОС);
  - АС программированного обучения (АСПО);
  - АС обучения деловым играм (АСОДИ);
  - тренажеры и тренажерные комплексы (ТТК);
- автоматизированные информационно-справочные системы (АИСС);
- интегрированные ИС (ИИС).

**АСУ** представляет собой АИС, предназначенную для автоматизации всех или большинства задач управления.

**АСУП** предназначены для автоматизации функций управленческого персонала предприятия. Учитывая наиболее широкое применение и разнообразие этого класса систем, часто любые АС понимают именно в данном толковании. К этому классу относятся АСУ различными организациями.

**АСУ ТП** решают задачи оперативного управления и контроля технических объектов в промышленности, энергетике, на транспорте. Они широко используются для поддержания технологического процесса в металлургической, химической и машиностроительной промышленности.

**СППР** – это ИС, целью которых является помощь лицам, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности. СППР возникли в результате слияния управленческих информационных систем и систем управления базами данных. Если в основе работы СППР лежат методы искусственного интеллекта, то говорят об интеллектуальной СППР. Близкие к СППР классы систем — это экспертные системы и автоматизированные системы управления.

**АИВС** предназначены для решения сложных в математическом отношении задач, требующих больших объемов самой разнообразной информации. Эти системы используются для обеспечения научных исследований и разработок, а также как подсистемы АСУ и СППР.

**ИРС** предназначены для обеспечения оперативных расчетов и автоматизации обмена информацией между рабочими местами в пределах организации или системы организаций. ИРС могут выступать в качестве подсистемы АСУ.

**САПР** – АИС, предназначенные для автоматизации функций инженеров-проектировщиков, конструкторов, архитекторов, дизайнеров при создании новой техники или технологии.

**АСНИ** – программно-аппаратные комплексы, предназначенные для научных

исследований и испытаний.

**АОС** – комплексы программно-технических, учебно-методической литературы и электронные учебники, обеспечивающих учебную деятельность.

**АСПО** ориентированы на обучение в основном по теоретическим разделам курсов и дисциплин. Учебный материал разделяется на порции и в зависимости от действий и ответов обучаемого на вопросы формируется очередная порция предоставляемой информации.

**АСОДИ** предназначена для подготовки и проведения деловых игр, сущность которых заключается в имитации принятия решений должностными лицами. в различных проблемных ситуациях путем игры по заданным правилам.

**ТТК** предназначены для обучения практическим навыкам работы на конкретных рабочих местах (постах).

**АИСС** предназначены для сбора, хранения, поиска и выдачи в требуемом виде потребителям информации справочного характера.

**ИИС** используются для автоматизации всех функций организации и охватывают весь цикл *работ* от планирования деятельности до сбыта продукции. Они включают в себя ряд модулей (подсистем), работающих в **едином информационном пространстве** и выполняющих функции поддержки соответствующих направлений деятельности.

Анализ современного состояния рынка АС показывает устойчивую тенденцию роста спроса на **АСУ предприятием**. Причем спрос продолжает расти именно на **интегрированные системы управления**.

По **масштабу** АС подразделяются на одиночные, групповые и корпоративные.

**Одиночные АИС** реализуются, как правило, на автономном персональном компьютере (сеть не используется). Такая система может содержать несколько простых приложений, связанных общим информационным фондом, и рассчитана на работу одного пользователя или группы пользователей, разделяющих по времени одно рабочее место. Подобные приложения создаются с помощью локальных систем управления базами данных (СУБД).

**Групповые АИС** ориентированы на коллективное использование информации членами рабочей группы и чаще всего строятся на базе локальной вычислительной сети. При разработке таких приложений используются серверы баз данных (SQL-серверы) для рабочих групп.

**Корпоративные АИС** являются развитием систем для рабочих групп, они ориентированы на крупные компании и могут поддерживать территориально разнесенные узлы или сети. Для таких систем характера архитектура клиент-сервер со специализацией серверов или же многоуровневая архитектура. При разработке таких систем могут использоваться те же серверы баз данных, что и при разработке групповых ИС.

### 1.1.3. Архитектура автоматизированных информационных систем

Рассматривая архитектуру крупных организаций, принято использовать понятие «корпоративная архитектура». Её можно представить в виде совокупности нескольких типов архитектур:

- техническая архитектура (Hardware architecture);
- программная архитектура (Software architecture);
- архитектура данных (Data architecture);
- ИТ-архитектура (Information Technology architecture);
- бизнес-архитектура (Business architecture).

**Техническая архитектура** является первым уровнем архитектуры автоматизированной информационной системы. Она описывает все аппаратные средства, использующиеся при выполнении заявленного набора функций, а также включает средства обеспечения сетевого взаимодействия и надёжности.

**Программная архитектура** представляет собой совокупность компьютерных программ, предназначенных для решения конкретных задач. Данный тип архитектуры необходим для описания приложений, входящих в состав АИС.

**Архитектура данных** объединяет в себе как физические хранилища данных, так и средства управления данными. На этом уровне описываются логические и физические модели данных, определяются правила целостности, составляются ограничения для данных.

**ИТ-архитектура** формирует базовый набор сервисов, которые используются как на уровне программной архитектуры, так и на уровне архитектуры данных. Основной функцией ИТ-архитектуры является обеспечение функционирования важных бизнес-приложений для достижения обозначенных бизнес-целей.

**Бизнес-архитектура** определяет стратегии ведения бизнеса, способы управления, принципы организации и ключевые процессы, представляющие для бизнеса огромную важность.

Можно выделить пять различных подходов к проектированию:

- календарный подход;
- подход, за основу которого взят процесс управления требованиями;
- подход, основанный на процессе разработки документации;
- подход, в основе которого лежит система управления качеством;
- архитектурный подход.

**Календарный подход** подразумевает составление графика предстоящих работ с их поэтапным выполнением.

**Подход на основе процесса управления требованиями** предполагает выделение большей части времени всего процесса разработки на функциональные характеристики системы, а нефункциональные, такие как масштабируемость, например,

практически не рассматриваются. Все решения в ходе проекта формируются исходя из локальных целей по реализации конкретного функционала.

При применении *подхода на основе процесса разработки документации*, большое количество времени тратится на формирования пакета документов, из-за чего может страдать качество самой разрабатываемой системы. Данный подход используется в правительственных организациях и крупных компаниях.

*Подход на основе системы управления качеством* включает в себя большое количество разноплановых мер для отслеживания на всех стадиях разработки системы наиболее значимых для функционирования системы параметров.

*Архитектурный подход* к проектированию АИС можно считать наиболее зрелым. Его ключевым аспектом является создание каркаса, адаптация которого под нужды конкретной системы будет легко осуществима. В соответствии с этим задача проектирования разбивается на две: *разработка многократно используемого каркаса* и *создание системы на его основе*. Данные подзадачи могут решаться различными группами специалистов. При использовании каркасов появляется возможность довольно быстро изменять функциональность системы за счёт итеративности процесса проектирования. Архитектурный подход призван ликвидировать недостатки, возникающие в процессе проектирования, основанном на управлении требованиями.

Одним из важнейших этапов проектирования архитектуры АИС является распределение функциональных компонентов программной системы по выбранной платформенной архитектуре. Можно выделить три направления развития платформенных архитектур: *автономные, централизованные и распределённые*.

*Автономная архитектура* подразумевает наличие всех функциональных компонентов системы на одном физическом устройстве, например, компьютере и не должна иметь связей с внешней средой.

*Централизованная архитектура* подразумевает выполнение всех требуемых задач на специально отведённом узле, мощности которого достаточно, чтобы удовлетворить потребности всех пользователей. Компоненты системы в данном случае распределяются между вычислительным узлом (мэйнфреймом), и терминальной станцией, за которой работает пользователь.

В *распределённой архитектуре* функциональные компоненты АИС распределяются по имеющимся узлам в зависимости от поставленных целей и задач.

Существуют следующие виды распределённых архитектур АИС:

- архитектура «файл-сервер»;
- архитектура «клиент-сервер»;
- многоуровневой архитектура;
- архитектура Интернет/интранет-технологий;
- смешанная архитектура.

**Архитектура «файл-сервер»** подразумевает наличие выделенного сетевого ресурса для хранения данных. При такой архитектуре все функциональные компоненты системы расположены на пользовательском компьютере, который называется «клиентом», а сами данные находятся на сервере. Дополнительные пользователи и приложения добавляют лишь незначительную нагрузку на центральный процессор. Каждый новый клиент добавляет вычислительную мощность к сети.

**Архитектура клиент-сервер** представляет собой сетевую инфраструктуру, в которой серверы являются поставщиками определённых сервисов (услуг), а клиентские компьютеры выступают их потребителями. Компоненты приложения разделяются и размещаются там, где они будут функционировать наиболее эффективно. Особенностью архитектуры клиент-сервер является использование выделенных серверов баз данных, понимающих запросы на языке структурированных запросов SQL и выполняющих поиск, сортировку и агрегирование информации. В настоящее время архитектура клиент-сервер получила признание и широкое распространение как способ организации приложений для рабочих групп и корпоративных АИС. Подобная организация работы повышает эффективность выполнения приложений за счет использования возможностей сервера БД, разгрузки сети и обеспечения контроля целостности данных.

**Многоуровневая архитектура АИС** стала развитием архитектуры клиент-сервер, в которой функция обработки данных вынесена на один или несколько отдельных серверов. Это позволяет разделить функции хранения, обработки и представления данных для более эффективного использования возможностей серверов и клиентов. В классической форме состоит из трех уровней: нижний уровень – приложения клиентов; средний уровень – сервер приложений; верхний уровень – удаленный специализированный сервер базы данных. Трехуровневая архитектура позволяет еще больше сбалансировать нагрузку на разные узлы и сеть, а также способствует специализации инструментов для разработки приложений и устраняет недостатки двухуровневой модели клиент-сервер.

**Архитектура Интернет/интранет-технологии** включает Web-узлы с интерактивным информационным наполнением, реализованных при помощи технологий Java, JavaBeans и JavaScript, взаимодействующих с предметной базой данных, с одной стороны, и с клиентским местом с другой. База данных, в свою очередь, является источником информации для интерактивных приложений реального времени.

**Смешанная архитектура**, эффективно работающая с БД, объединяет Интернет/интранет-технологии с многоуровневой архитектурой. При этом структура информационного приложения приобретает следующий вид: браузер – сервер приложений – сервер баз данных – сервер динамических страниц – web-сервер.



## 1.2. Жизненный цикл и стандарты проектирования систем

### 1.2.1. Понятие и структура проектирования

В соответствии со стандартом ISO 24765 [21] «**Проектирование** - процесс определения принципиальной организации системы, воплощенной в её элементах, их взаимоотношениях друг с другом и со средой, а также других характеристиках системы или её части. Результатом проектирования является **проект** – целостная совокупность моделей, свойств или характеристик, описанных в форме, пригодной для реализации системы.

Проектирование системы направлено на представление системы, соответствующее предусмотренной цели, принципам и замыслам; оно включает оценку и принятие решений по выбору таких компонентов системы, которые отвечают её архитектуре и укладываются в предписанные ограничения.

Фактически, **проектирование** – это деятельность человека или организации/ий по созданию проекта, то есть прототипа, прообраза предполагаемого или возможного объекта, состояния; комплекта документации, предназначенной для создания определённого объекта, его эксплуатации, ремонта и ликвидации, а также для проверки или воспроизведения промежуточных и конечных решений, на основе которых был разработан данный объект.

Проектирование может включать несколько этапов от подготовки технического задания до испытания опытных образцов. Объектом проектирования является проект материального предмета.

Понятие «проектирование» не включает в себя стадию реализации проекта.

Проектирование обладает своей методологией [43], которая включает структуру деятельности, принципы и нормы деятельности, субъектов, объект и его модели, методы и др.

Для всех подходов к проектированию сложных систем характерны также следующие особенности [53]:

1. Структуризация процесса проектирования, выражаемая декомпозицией проектных задач и документации, выделением стадий, этапов, проектных процедур. Эта структуризация является сущностью блочно-иерархического подхода к проектированию.

2. Итерационный характер проектирования.

3. Типизация и унификация проектных решений и средств проектирования.

4. Использование знаний фундаментальных, технических и социально-экономических наук, подчинение всей деятельности удовлетворению интересов, прежде всего, человека (покупателя, производителя, разработчика). Слово «проект» еще применяется в значении «программа», «план действий».

Проектирование связано не только с техническими объектами. Так, имеется

социальное проектирование, проектирование программного обеспечения и другие. Отличительной особенностью проектирования является его практическая направленность (обязательное наличие практических результатов) и персональная ответственность за полученные и переданные заказчику результаты.

Внутри процесса проектирования, наряду с расчетными этапами и экспериментальными исследованиями, часто выделяют процесс *конструирования*.

**Конструирование** – деятельность по созданию материального образа разрабатываемого объекта, ей свойственна работа с физическими моделями и их графическими изображениями. Эти модели и изображения, а также некоторые виды изделий называют конструкциями.

Современному уровню развития техники стали присущи не только сложность проектируемых объектов, но и их интенсивное воздействие на общество и окружающую среду, тяжесть последствий аварий из-за ошибок разработки и эксплуатации, высокие требования к качеству и цене, сокращению сроков выпуска новой продукции. При создании подобных объектов их уже необходимо рассматривать в виде систем, т.е. комплекса взаимосвязанных внутренних элементов с определенной структурой, широким набором свойств и разнообразными внутренними и внешними связями.

Жизненная необходимость учета этих обстоятельств, а также необходимость рассмотрения разрабатываемых объектов в виде систем, постепенно заставляла вносить изменения в традиционный характер и методологию проектной деятельности. К настоящему времени сформировалась новая проектная идеология, получившая название системного проектирования.

**Системное проектирование** комплексно решает поставленные задачи, принимает во внимание взаимодействие и взаимосвязь отдельных объектов-систем и их частей, как между собой, так и с внешней средой, учитывает социально-экономические и экологические последствия их функционирования. Системное проектирование основывается на тщательном совместном рассмотрении объекта проектирования и процесса проектирования, которые в свою очередь включают еще ряд важных частей, показанных на рис. 1.3.

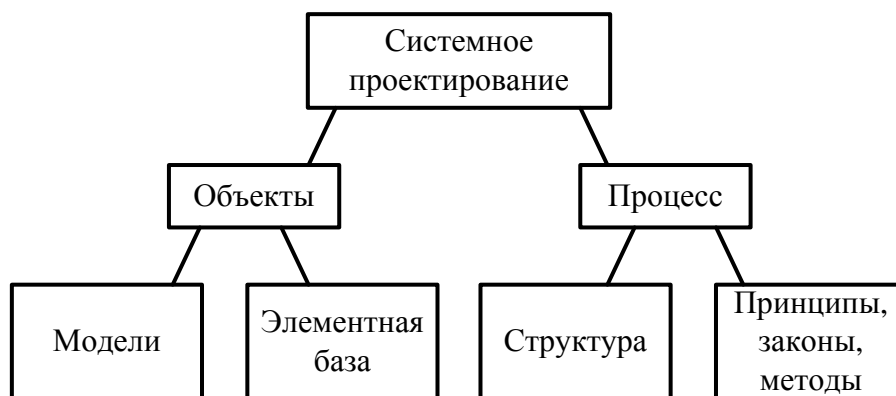


Рис. 1.3. Основные части проектирования

Очевидно, что в настоящее время проектирование все более и более становится системным. Поэтому словосочетание «системное проектирование» ниже будет использоваться только в случаях, если потребуется подчеркнуть важность системности разработки.

Проектирование – это один из видов работ, результатом которых является комплект проектной документации на материальный объект, выполнение работы, или оказание услуги. Поэтому участников этих работ можно разделить на *потребителей* (заказчиков проектных работ) и *поставщиков* (исполнителей этих работ). Исполнителя-специалиста по разработке проекта обобщенно называют проектировщиком или разработчиком. Если продукция создается для собственного потребления, то возможно соединение в одном лице заказчика и исполнителя.

Существует еще один участник этих работ — государство, которым создана система мер по защите потребителя посредством контроля, лицензирования, выпуска нормативной документации.

Проектирование, как осознанная целенаправленная деятельность, обладает определенной структурой, т.е. последовательностью и составом стадий и этапов разработки проекта, совокупностью процедур и привлекаемых технических средств, взаимодействием участников процесса.

### 1.2.2. Процессы и модели жизненного цикла систем

Каждая система имеет свой жизненный цикл (ЖЦ). Жизненный цикл может быть описан с использованием абстрактной функциональной модели, представляющей концептуализацию потребности в системе, ее реализации, применения, развития и ликвидации [11]. Система развивается на протяжении ЖЦ в результате действий, осуществляемых и управляемых людьми, работающими в организациях и использующими определенные процессы в своей деятельности. Детали модели жизненного цикла выражаются в терминах этих процессов, их результатов, взаимосвязи и возникновения.

Процессы ЖЦ, определенные стандартом ГОСТ Р ИСО/МЭК 15288-2005, могут применяться любой организацией при приобретении и использовании или создании и поставке системы. Они распространяются на любой уровень системной иерархии и на любую стадию жизненного цикла. Процессы жизненного цикла основываются на принципах *модульности* (максимальная слаженность функций процесса и минимальная связь между процессами) и *собственности* (процесс связывается с ответственностью). Функции, которые осуществляются данными процессами, определяются в зависимости от конкретных целей, результатов и набора действий, составляющих данный процесс. Процессы, описанные в настоящем стандарте, не препятствуют и не исключают использование дополнительных процессов, которые организация считает необходимыми.

Процессы ЖЦ сложных систем в соответствии с [11] подразделяются на четыре группы процессов (рис. 1.4):

- процессы соглашения;
- процессы предприятия;
- процессы проекта;
- технические процессы.

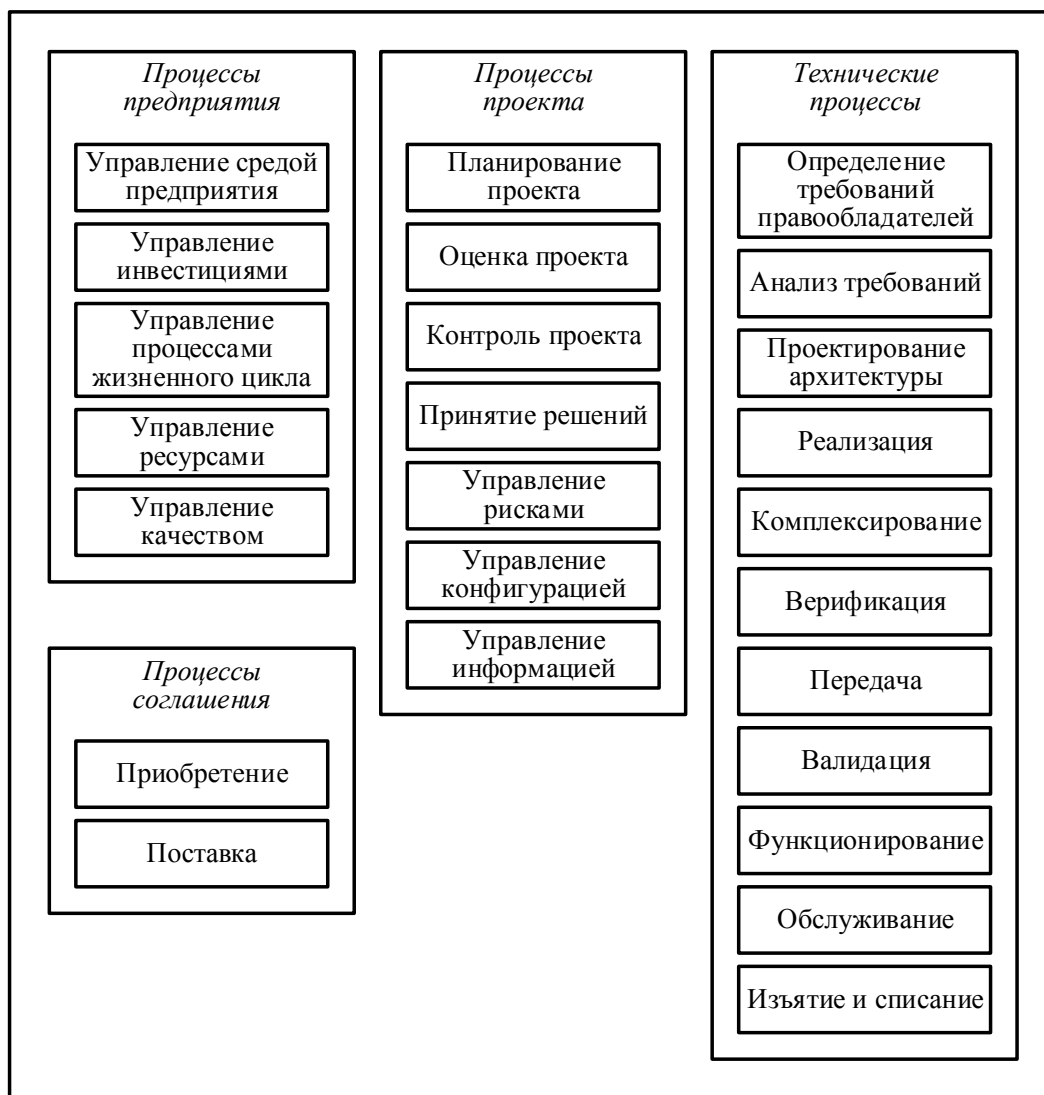


Рис. 1.4. Процессы жизненного цикла системы [11]

**Процессы соглашения** определяют действия, необходимые для достижения соглашения между двумя организациями, и включают процесс приобретения и процесс поставки.

**Процессы предприятия** управляют способностью организации приобретать и поставлять продукцию или услуги посредством запуска проектов, их поддержки и контроля. Данные процессы обеспечивают ресурсы и инфраструктуру, необходимые для осуществления проектов, и гарантируют достижение целей и исполнение обязательств организации по соглашениям.

**Процессы проекта** используются для установления и выполнения планов,

оценки фактических достижений и продвижений проекта в соответствии с планами и для контроля выполнения проекта вплоть до его завершения.

**Технические процессы** используются для определения требований к системе, преобразования этих требований в эффективный продукт, позволяющий осуществлять при необходимости устойчивое воспроизводство этого продукта, использовать его для обеспечения требуемых услуг, поддерживать обеспечение этими услугами и удалять продукт, когда он изымается из обращения.

Процессы ЖЦ программного обеспечения (ПО) АС регламентируются другим стандартом – ГОСТ Р ИСО/МЭК 12207.

В соответствии с ГОСТ Р ИСО/МЭК 12207- 99 процессы жизненного цикла включают в себя работы, которые могут выполняться в жизненном цикле программных средств и распределены по пяти основным, восьми вспомогательным и четырем организационным процессам.

**Основные процессы жизненного цикла ПО** [9] реализуются под управлением основных сторон, вовлеченных в жизненный цикл программных средств. Под основной стороной понимают одну из тех организаций, которые иницируют или выполняют разработку, эксплуатацию или сопровождение программных продуктов. Основными сторонами являются заказчик, поставщик, разработчик, оператор и персонал сопровождения программных продуктов.

**Основными процессами** являются:

1. **Процесс заказа.** Определяет работы заказчика, то есть организации, которая приобретает систему, программный продукт или программную услугу.
2. **Процесс поставки.** Определяет работы поставщика, то есть организации, которая поставляет систему, программный продукт или услугу заказчику.
3. **Процесс разработки.** Определяет работы разработчика, то есть организации, которая проектирует и разрабатывает программный продукт.
4. **Процесс эксплуатации.** Определяет работы оператора, то есть организации, которая обеспечивает эксплуатационное обслуживание вычислительной системы в заданных условиях в интересах пользователей.
5. **Процесс сопровождения.** Определяет работы персонала сопровождения, то есть организации, которая предоставляет услуги по сопровождению программного продукта, состоящие в контролируемом изменении программного продукта с целью сохранения его исходного состояния и функциональных возможностей. Данный процесс охватывает перенос и снятие с эксплуатации программного продукта.

**Вспомогательные процессы жизненного цикла ПО** [9] состоят из восьми процессов. Вспомогательный процесс является целенаправленной составной частью другого процесса, обеспечивающей успешную реализацию и качество вы-

полнения программного проекта. Вспомогательный процесс, при необходимости, инициируется и используется другим процессом.

**Вспомогательными процессами** являются:

1. **Процесс документирования.** Определяет работы по описанию информации, выдаваемой в процессе жизненного цикла.

2. **Процесс управления конфигурацией.** Определяет работы по управлению конфигурацией.

3. **Процесс обеспечения качества.** Определяет работы по объективному обеспечению того, чтобы программные продукты и процессы соответствовали требованиям, установленным для них, и реализовывались в рамках утвержденных планов. Совместные анализы, аудиторские проверки, верификация и аттестация могут использоваться в качестве методов обеспечения качества.

4. **Процесс верификации.** Определяет работы (заказчика, поставщика или независимой стороны) по верификации программных продуктов по мере реализации программного проекта.

5. **Процесс аттестации.** Определяет работы (заказчика, поставщика или независимой стороны) по аттестации программных продуктов программного проекта.

6. **Процесс совместного анализа.** Определяет работы по оценке состояния и результатов какой-либо работы. Данный процесс может использоваться двумя любыми сторонами, когда одна из сторон (проверяющая) проверяет другую сторону (проверяемую) на совместном совещании.

7. **Процесс аудита.** Определяет работы по определению соответствия требованиям, планам и договору. Данный процесс может использоваться двумя сторонами, когда одна из сторон (проверяющая) контролирует программные продукты или работы другой стороны (проверяемой).

8. **Процесс решения проблемы.** Определяет процесс анализа и устранения проблем (включая несоответствия), независимо от их характера и источника, которые были обнаружены во время осуществления разработки, эксплуатации, сопровождения или других процессов.

**Организационные процессы жизненного цикла** [9] применяются в какой-либо организации для создания и реализации основной структуры, охватывающей взаимосвязанные процессы ЖЦ и соответствующий персонал, а также для постоянного совершенствования данной структуры и процессов. Эти процессы, как правило, являются типовыми, независимо от области реализации конкретных проектов и договоров; однако уроки, извлеченные из таких проектов и договоров, способствуют совершенствованию организационных вопросов.

Организационными процессами являются:

1. **Процесс управления.** Определяет основные работы по управлению, включая управление проектом, при реализации процессов жизненного цикла.

2. **Процесс создания инфраструктуры.** Определяет основные работы по созданию основной структуры процесса жизненного цикла.

3. **Процесс усовершенствования.** Определяет основные работы, которые организация (заказчика, поставщика, разработчика, оператора, персонала сопровождения или администратора другого процесса) выполняет при создании, оценке, контроле и усовершенствовании выбранных процессов ЖЦ

4. **Процесс обучения.** Определяет работы по соответствующему обучению персонала. Данный ГОСТ Р ИСО/МЭК 12207-99 охватывает жизненный цикл программных средств от концепции замыслов через определение и объединение процессов для заказа и поставки программных продуктов и услуг. Кроме того, данная структура предназначена для контроля и модернизации данных процессов. Настоящий стандарт следует адаптировать для конкретной организации, проекта или приложения. Настоящий стандарт предназначен для использования как в случае отдельно поставляемых программных средств, так и для программных средств, встраиваемых или интегрируемых в общую систему [9].

**Модель жизненного цикла информационной системы** может включать:

- 1) стадии;
- 2) основные результаты выполнения работ на каждой стадии;
- 3) ключевые события.

Стадии представляют собой основные периоды ЖЦ, связанные с системой и относящиеся к состоянию описания системы или непосредственно к системе. Стадии отображают значимый прогресс и достижение запланированных этапов развития системы на протяжении всего жизненного цикла и дают начало важнейшим решениям относительно своих входов и выходов. Эти решения используются организациями для учета неопределенностей и рисков, непосредственно связанных с затратами, сроками и функциональностью при создании или применении системы. Таким образом, стадии обеспечивают организации структурой работ, в рамках которых управление предприятием обладает высокой способностью для обзора и контроля проекта и технических процессов.

ЖЦ АИС характеризуется периодом времени от идеи создания АИС до момента вывода ее из эксплуатации и включает в себя следующие стадии [11]: *предпроектное обследование, проектирование, создание системы, ввод в эксплуатацию, эксплуатация системы, вывод из эксплуатации.*

Наибольшее распространение получили следующие модели жизненного цикла информационных систем (ИС): *каскадная* (классическая или водопадная), *итерационная* и *спиральная*.

#### ***Каскадная (классическая, водопадная) модель ЖЦ АИС***

Модель была предложена в 1970 году Уинстоном Ройсом. Переход на следующий этап осуществляется после полного окончания работ по предыдущему

этапу, при этом оформляется полный комплект рабочей документации. Все этапы выполняются в строгой последовательности с утвержденными сроками и четкими затратами. Это основные достоинства каскадной модели ЖЦ АИС, которая применялась в условиях полной определенности решаемых задач и совершенно неприемлема, когда и разработчики, и заказчики не имеют четкого видения всех особенностей проектируемой АИС. Кроме того, невозможно идти дальше, пока не сдан предыдущий этап, а после сдачи нельзя возвращаться к нему для устранения обнаруженных недочетов, что серьезно затрудняет работы по совершенствованию и доработке создаваемой АИС. Эта модель нравится и заказчикам, и разработчикам по причине жесткой дисциплины финансирования этапов только после их предъявления. Но полностью отсутствует гибкость в работе над созданием АИС.

Каскадная модель представлена на рис. 1.5. На практике все же приходится возвращаться к предыдущим этапам и в этом случае, в последнее время наиболее востребованной стала итерационная модель ЖЦ АИС.

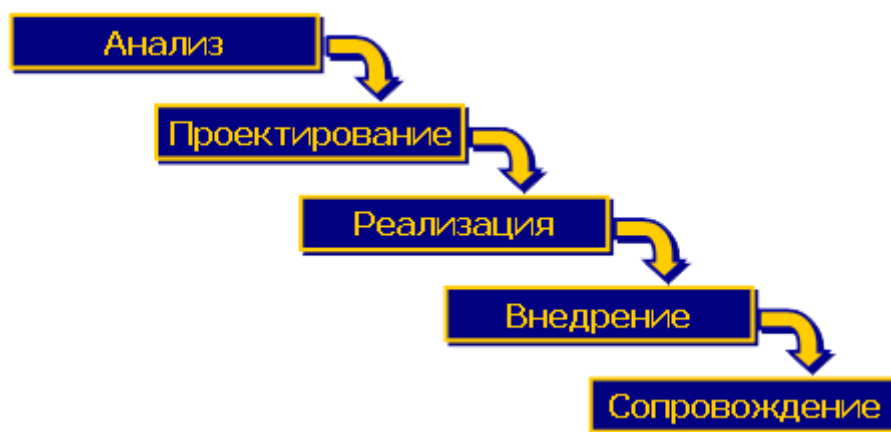


Рис. 1.5. Каскадная модель ЖЦ АИС

### ***Итерационная модель ЖЦ АИС***

Поэтапная модель с промежуточным контролем — итерационная модель разработки информационной системы. Каждый этап имеет обратные связи в процессе корректировки и создает условия для корректировки ранее созданных этапов. При этом трудоемкость работ и временные затраты существенно сокращаются по сравнению с водопадной моделью жизненного цикла. Итерационная модель ЖЦ информационной системы представлена на рис. 1.6.

Создание АИС – это организованный процесс построения и последовательного преобразования согласованных моделей на всех этапах жизненного цикла. При этом все разработанные модели находятся в репозитории проекта и доступны всем разработчикам, что позволяет эффективно вести одновременную работу над проектированием и созданием информационной системы.





Рис. 1.6. Итерационная модель ЖЦ АИС

### ***Спиральная модель ЖЦ АИС***

Спиральная модель предложена Барри Бозм в 1988 году и определяет, в основном стартовые этапы жизненного цикла информационной системы. При этом обосновывается и проверяется возможность реализации спроектированных технических решений. На каждом витке создается прототип проектируемой информационной системы, который на следующих витках спирали ЖЦ АИС совершенствуется, дополняется и доводится до полного внедрения. При этом не обязательно дожидаться окончания каждого этапа, данная модель позволяет переходить на следующие витки спирали и решать проблемы или недоделки на следующем уровне, что делает работу над проектом более эффективной, гибкой и завершить в более сжатые сроки. Новый виток спирали соответствует поэтапной модели создания фрагмента информационной системы.

При использовании спиральной модели ЖЦ:

- происходит ориентация на модернизацию информационной системы;
- осуществляется аккумулярование всех решений в процессе проектирования и создания моделей и прототипов информационной системы;
- проводится анализ издержек и всех рисков в процессе проектирования ИС.

Спиральный процесс состоит из следующей повторяющейся последовательности:

1. Определение требований.
2. Анализ.
3. Проектирование.
4. Реализация и тестирование.
5. Интеграция.
6. Внедрение.

Этот многократный цикл, завершающийся созданием новой версии информационной системы, представлен на рис. 1.7.

Для применения спиральной модели ЖЦ АИС может быть несколько причин.

Это необходимость минимизации рисков и возможность представления заказчику прототипа или эскизной версии проекта для конкретизации пожеланий и учета их в следующих циклах. А также в случае если разрабатываемая информационная система достаточно сложна и существует реальная необходимость создавать промежуточные версии продукта, не откладывая эту работу на финишные этапы, как это предписывает водопадная модель.

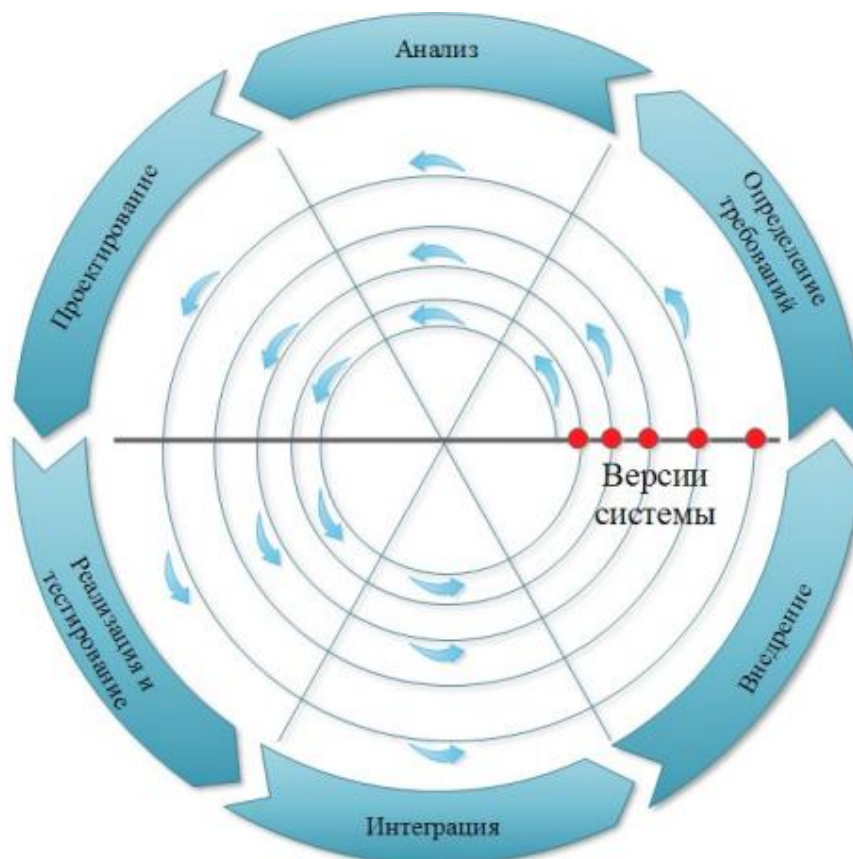


Рис. 1.7. Спиральная модель ЖЦ АИС

Основная задача спиральной модели ЖЦ АИС заключается в том, чтобы на каждой итерации создавать очередную версию системы, используя разработанный прототип предыдущих этапов. Такая модель позволяет более гибко работать с заказчиком, постоянно учитывать его замечания и предложения, совершенствовать проектируемую систему в процессе каждого нового витка спирали.

### 1.2.3. Стадии проектирования систем

В настоящее время существуют два представления структуры проектирования, подобные по форме, но различные по целям и подходам к деятельности. Это – структура в виде *стадий разработки проектной документации* (стадий проектирования) и структура *процесса проектирования*.

Соответствующая им структура регламентирована стандартом (ГОСТ 2.103, 15.201) и используется при официальных взаимоотношениях между заказчиком и исполнителем или между соисполнителями работ.

Данная структура устанавливает *стадии разработки конструкторской документации* для изделий всех отраслей промышленности и этапы выполнения работ. Эта документация необходима для отчета перед заказчиком о проделанной работе, возможности проверки или повторения разработок другими исполнителями, подготовки производства и обслуживания изделия в период эксплуатации. Основные стадии (этапы) структуры представлены на рис. 1.8.

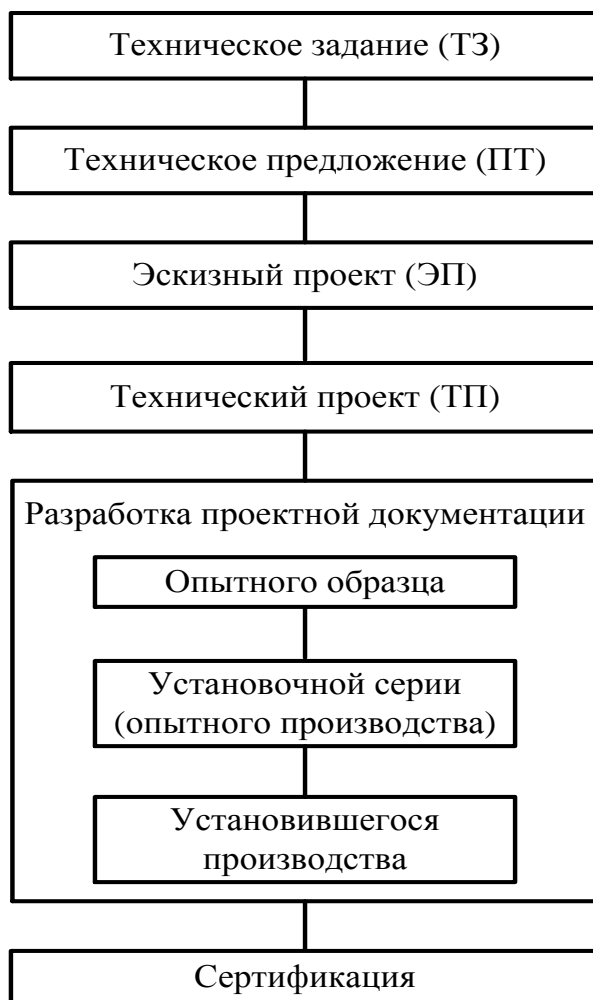


Рис. 1.8. Стадии разработки проектной документации

**Техническое задание (ТЗ)** устанавливает основное назначение, технические и тактико-технические характеристики, показатели качества и технико-экономические требования к разрабатываемому объекту, предписание по выполнению необходимых стадий создания документации и ее состав, а также специальные требования к изделию.

**Техническое предложение (ПТ)** – совокупность документов, содержащих техническое и технико-экономическое обоснование (ТЭО) целесообразности разработки проекта. Такое заключение дается на основании анализа ТЗ заказчика и различных вариантов возможных решений, их сравнительной оценки с учетом особенностей разрабатываемого и существующих изделий, а также патентных материалов.

Согласованное и утвержденное в установленном (на предприятии, в министерстве и т.п.) порядке ПТ является основанием для разработки эскизного проекта.

**Эскизный проект (ЭП)** – совокупность документов, содержащих принципиальные решения и дающих общее представление об устройстве и принципе работы разрабатываемого объекта, а также данные, определяющие его назначение, основные параметры и габаритные размеры. В случае большой сложности объекта этому этапу может предшествовать аванпроект (предпроектное исследование), обычно содержащий теоретические исследования, предназначенные для обоснования принципиальной возможности и целесообразности создания данного объекта. При необходимости на стадии ЭП проводят изготовление и испытание макетов разрабатываемого объекта.

**Технический проект (ТП)** – совокупность документов, которые должны содержать окончательные технические решения, дающие полное представление об устройстве проектируемого объекта, исходные данные для разработки рабочей документации.

На стадии **рабочего проекта (РП)** сначала разрабатывают подробную документацию для изготовления опытного образца и последующего его испытания. Испытания проводят в ряд этапов (от заводских до приемо-сдаточных), по результатам которых корректируют проектные документы. Далее разрабатывают рабочую документацию для изготовления установочной серии, ее испытания, оснащения производственного процесса основных составных частей изделия. По результатам этого этапа снова корректируют проектные документы и разрабатывают рабочую документацию для изготовления и испытания головной (контрольной) серии. На основе документов окончательно отработанных и проверенных в производстве изделий, изготовленных по зафиксированному и полностью оснащеному технологическому процессу, затем разрабатывается завершающая рабочая документация установившегося производства.

В процессе разработки проектной документации (рис. 1.8) в зависимости от сложности решаемой задачи допускается объединять между собой ряд этапов. Этапы постановки ТЗ и технического предложения могут входить в цикл научно-исследовательских работ (НИР), а этапы эскизного и технического проектирования – образовывать цикл опытно-конструкторских работ (ОКР).

Завершает цикл работ этап, подводящий итог проектной деятельности – **сертификация**. Ее назначение – определение уровня качества созданного изделия и подтверждение его соответствия требованиям тех стран, где предполагается его последующая реализация. Необходимость выделения этого этапа в виде самостоятельного вызвана тем, что в настоящее время экспорт продукции или ее реализация внутри страны во многих случаях недопустимы без наличия у нее сертификата качества. Сертификацию осуществляют специальные организации

(органы по сертификации), зарегистрированные в Госстандарте России. На основе анализа протоколов испытаний и состояния производства продукции и других факторов орган по сертификации дает оценку соответствия продукции установленным требованиям, оформляет и регистрирует сертификат. В течение всего срока действия сертификата за сертифицированной продукцией ведется инспекционный контроль. Сертификация может быть обязательной или добровольной. *Обязательной сертификации* подлежат товары, на которые законами или стандартами установлены требования, обеспечивающие безопасность жизни и здоровья потребителей, охрану окружающей среды, предотвращение причинения вреда имуществу потребителя. *Добровольная сертификация* проводится по инициативе предприятий. Обычно это делается с целью официального подтверждения характеристик продукции, изготавливаемой предприятием, и, как следствие, повышения доверия к ней у потребителей.

Базовым набором стандартов в РФ при проектировании АС являются стандарты ГОСТ 34 серии. Несмотря на то, что данные стандарты разработаны в конце 80х- начале 90х годов, они представляют собой целостную систему нормативного обеспечения проектирования и используются до сих пор.

Основные термины и определения в области АС установлены в ГОСТ 34.003-90, распространяющемся на АС, используемые в различных сферах деятельности (управление, исследования, проектирование и т. п., включая их сочетание), содержанием которых является переработка информации.

Стадии создания автоматизированных систем и содержание работ на стадиях определены ГОСТ 34.601-90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Стадии создания»:

- 1) формирование требований к АС;
- 2) разработка концепции АС;
- 3) техническое задание;
- 4) эскизный проект;
- 5) технический проект;
- 6) рабочая документация;
- 7) ввод в действие;
- 8) сопровождение АС.

Стандартом допускается при выполнении несложных проектов исключать стадию «Эскизный проект» и отдельные этапы, а также объединять стадии «Технический проект» и «Рабочая документация» в одну стадию «Технорабочий проект». Проектирование небольших ЛВС как раз относится к такого рода проектам.

Технические требования к АС являются основным техническим документом на разработку системы, на основе которого разрабатывается **техническое задание** на создание объекта автоматизации в соответствии с ГОСТ 34.602-89.

Полный перечень документации, разрабатываемый на этапах создания АС «Эскизный проект», «Технический проект» и «Рабочая документация» приводится в ГОСТ 34.201-89 «Основные виды, комплектность и обозначение документов при создании автоматизированных систем». Зачастую пакет рабочей документации ограничивается следующими документами:

- 1) Руководство пользователя (администратора);
- 2) Инструкция по эксплуатации комплекса технических средств;
- 3) Общее описание системы (в случае отсутствия документа «Пояснительная записка к техническому (эскизному) проекту» данный документ нецелесообразен так большинство разделов дублируются);
- 4) Программа и методика испытаний.

Окончательное описание проектируемого объекта представляет собой полный комплект схемной, конструкторской и технологической документации, оформленной в соответствии с требованиями ГОСТов:

- ЕСКД (единая система конструкторской документации);
- ЕСТД (единая система технологической документации);
- ЕСПД (единая система программной документации).

Этот комплект документации предназначен для использования в процессе изготовления и эксплуатации объекта проектирования.

### **1.3. Требования, предъявляемые к проектируемым объектам**

#### **1.3.1. Основные виды требований к автоматизированным системам**

К типовым требованиям к научно-технической продукции относят требования функциональные (показатели назначения), надежности, технологичности, стандартизации и унификации, ограничения вредных воздействий (эргономичность и экологичность), эстетичность, экономичность, патентно-правовые.

Систематизированный перечень требований представлен на рис. 1.9. При этом существует еще две группы требований, явно не выделяемые, но способные сильно повлиять на принимаемое решение и ход работ. Это – политические цели и морально-этические принципы и устои. Они связаны с общественным сознанием и изучаются соответствующими науками. Часто их учитывают косвенно, предусматривая социальные последствия от внедрения результатов разработки на ближнюю и дальнюю перспективы.

В ГОСТ 34.602-89 изложен перечень требований к системе, которые указываются в техническом задании на проектирование:

- 1) требования к системе в целом;
- 2) требования к функциям (задачам), выполняемым системой;
- 3) требования к видам обеспечения.

В подразделе «*Требования к системе в целом*» указывают:

1) *требования к структуре и функционированию системы* (перечень подсистем, уровни иерархии, степень централизации, способы информационного обмена, режимы функционирования, взаимодействие со смежными системами, перспективы развития системы);

2) *требования к персоналу* (численность и квалификация персонала системы, порядок подготовки и контроля знаний, требуемый режим его работы);

3) *требования к показателям назначения* (степень приспособляемости системы к изменениям процессов управления и значений параметров);

4) *требования к показателям* (надежности, безопасности, эргономике, транспортабельности, эксплуатации, техническому обслуживанию и ремонту, защите и сохранности информации, защите от внешних воздействий, к патентной чистоте, по стандартизации и унификации).

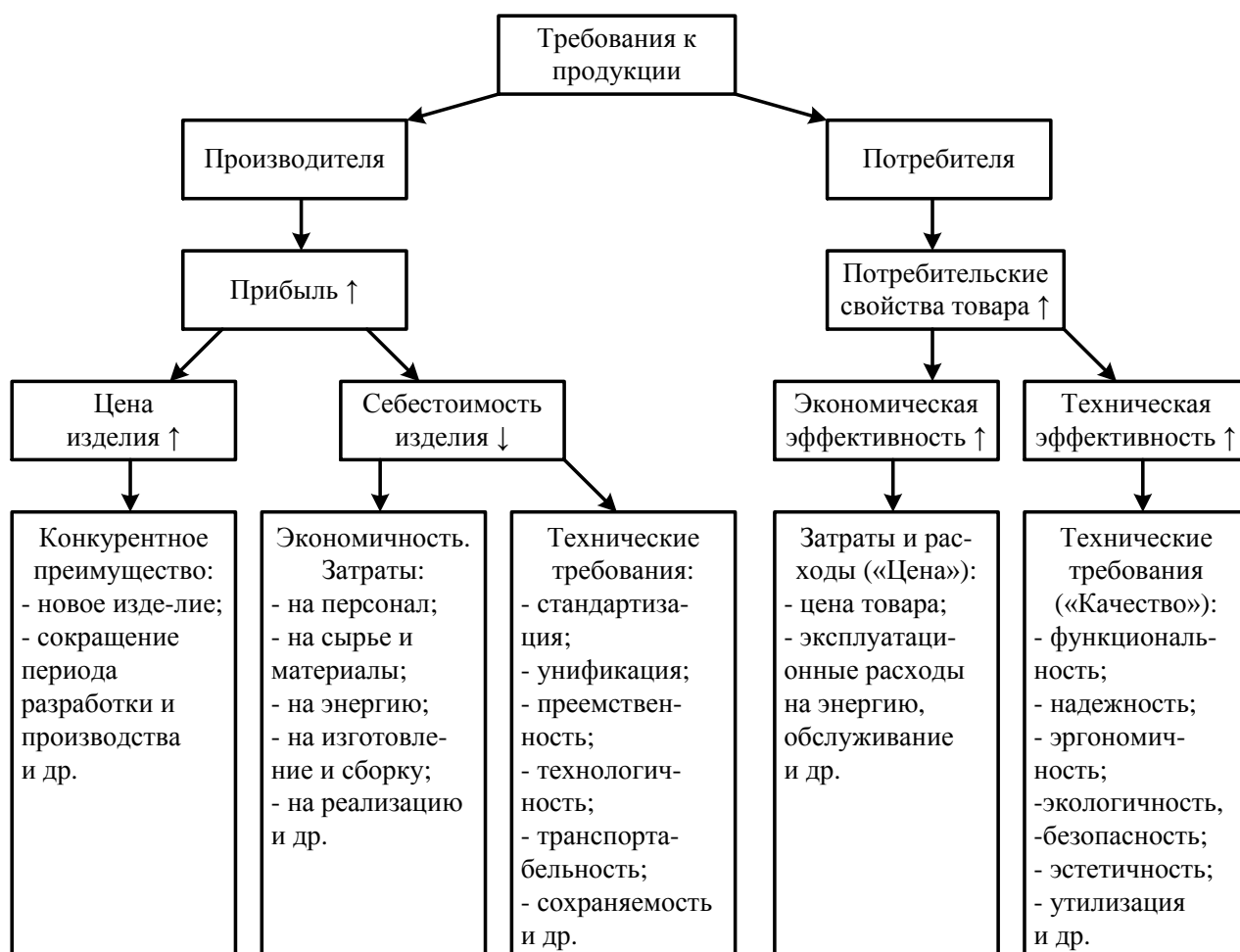


Рис. 1.9. Состав возможных требований, предъявляемых к проектируемым объектам

В подразделе «*Требование к функциям (задачам)*», выполняемым системой, приводят:

1) перечень подлежащих автоматизации функций и задач или их комплексов (по каждой подсистеме);

2) временной регламент реализации каждой функции, задачи (комплекса);

3) требования к качеству реализации каждой функции (задачи, комплекса задач), к форме представления выходной информации, характеристики точности, достоверности выдачи результатов;

4) перечень и критерии отказов для каждой функции.

В подразделе «**Требования к видам обеспечения**» в зависимости от вида системы приводят требования к видам обеспечения системы:

1) **математическому** (состав и область применения мат. моделей и методов, типовых и разрабатываемых алгоритмов);

2) **информационному** (состав, структура и организация данных, обмен данными между компонентами системы, информационная совместимость со смежными системами, используемые классификаторы, СУБД, контроль данных и ведение информационных массивов, процедуры придания юридической силы выходным документам);

3) **лингвистическому** (языки программирования, языки взаимодействия пользователей с системой, системы кодирования, языки ввода-вывода);

4) **программному** (независимость программных средств от платформы, качество программных средств и способы его контроля, использование фондов алгоритмов и программ);

5) **техническому** (виды технических средств, функциональные, конструктивные и эксплуатационные характеристики средств технического обеспечения системы);

6) **метрологическому** (требования к точности измерений параметров и (или) к метрологическим характеристикам измерительных каналов, к метрологической совместимости технических средств систем и др.);

7) **организационному** (структура и функции эксплуатирующих подразделений, защита от ошибочных действий персонала);

8) **методическому** (состав нормативно-технической документации).

### 1.3.2. Проектные и производственные требования

#### 1. Стандартизация, унификация, преемственность.

В различных видах деятельности существуют документы, которые с позиции правовых норм регламентируют общие принципы, правила или характеристики. Это — нормативные документы, охватывающие такие понятия, как стандарты, нормы, правила, своды правил, регламенты и другие подобные документы.

Закон РФ «О техническом регулировании» определяет **стандартизацию** как деятельность по установлению правил и характеристик в целях их добровольного многократного использования, направленную на достижение упорядоченности в сферах производства и обращения и повышение конкурентоспособности продукции, работ, услуг. Комитет ИСО по изучению научных принципов стандартизации (СТАКО) определяет **стандартизацию** как установление и применение правил с целью упорядочения деятельности в определенной области на пользу и при участии всех заинтересованных сторон для достижения оптимальной экономии при соблюдении норм безопасности.

Стандартизацию проводят органы стандартизации, наделенные законным правом руководить разработкой и утверждать нормативные документы и другие



правила, придавая им статус стандартов. В России компетентными органами в области стандартизации являются Госстандарт России и Госстрой. Основопологающим нормативным документом по стандартизации Госстандарта России установлена «Государственная система стандартизации» (ГСС).

Комплекс стандартов ГСС РФ (ГОСТ Р1.0, ГОСТ Р1.1, ГОСТ Р1.2 и др.) представляет собой систему взаимоувязанных правил и положений, определяющих цели и задачи стандартизации, организацию и методику проведения работ по стандартизации во всех производственных отраслях России. ГСС устанавливает порядок разработки, оформления, согласования, утверждения, издания, обращения стандартов разных уровней стандартизации и других нормативных документов, а также контроля за их внедрением и соблюдением.

Стандарт имеет распространение в пределах компетенции органа стандартизации, и соответственно этим пределам различают следующие уровни стандартизации:

1. Международная стандартизация. Органом по стандартизации является ИСО (ISO). Нормативным документом ИСО являются стандарты ИСО.

2. Межрегиональная стандартизация. Охватывает ряд независимых государств (СНГ, ЕЭС). Нормативным документом стран СНГ является межрегиональный стандарт.

3. Национальная стандартизация. Это – стандартизация в пределах одного государства. Нормативным документом по национальной стандартизации в России установлен государственный стандарт России – ГОСТ Р, в ФРГ – DIN, в Великобритании – BS, и т.д.

4. Правила, нормы и рекомендации в области стандартизации, общероссийские классификаторы технико-экономической и социальной информации.

5. Стандарты организаций – отраслевые стандарты (ОСТ), стандарты предприятий (СТП), стандарты обществ и т.п. Это – низший уровень стандартизации.

Государства стремятся к согласованию национальных стандартов и выпуску международных стандартов (например, стандарт ИСО на допуски и посадки), что упрощает обслуживание и ремонт экспортной продукции, облегчает продвижение товара на внешние рынки.

ОСТы имеют применение в выпустивших их отраслях промышленности. Обычно в виде ОСТов оформляются типовые ситуации, которые после дальнейшей практической проверки и подтверждения своей важности служат основой для выпуска соответствующего ГОСТа.

СТП имеют применение только на выпустившем их предприятии. Часто оформляются в виде нормалей, которые устанавливают ограничения на применяемую номенклатуру (перечень) деталей, материалов, норм и т.п., что вызывается особенностями снабжения и производства. Требования стандартов предприятий могут быть обязательными и для других предприятий, если между ними

существуют договорные отношения, в том числе устанавливающие обязательность исполнений определенного круга стандартов одного из предприятий.

Наряду со стандартами другими нормативными документами, регулирующими отношения в области установления, применения и исполнения требований к продукции, процессам производства, эксплуатации, хранения, перевозки, реализации и утилизации, являются технические регламенты.

**Технический регламент** - документ, который принят международным договором РФ, ратифицирован и устанавливает обязательные для применения и использования минимально необходимые требования к объектам технического регулирования, обеспечивающие безопасность (химическую, биологическую, пожарную, механическую и т.д.) и единство измерений.

На переходный период, до принятия необходимых технических регламентов, с указанными целями должны применяться соответствующие требования ранее принятых ГОСТ, санитарных и строительных норм и правил (СанПиН, СНиП).

Объектами стандартизации являются продукция, работа (процесс создания продукции) и услуга (процесс создания нематериальной продукции), подлежащие или подвергшиеся стандартизации. Стандарты разрабатывают на наиболее распространенные и типовые ситуации. В частности, объектом стандартизации может быть конкретное изделие, параметрические ряды однотипных изделий, отдельные свойства изделия, меры и единицы измерения, нормы, правила, требования, условия, термины и определения, обозначения и т.д., имеющие перспективу многократного применения в различных сферах деятельности человека. Все правила до оформления их в стандарты проходят длительную проверку на практике, и поэтому заключают в себе богатый опыт предшествующих поколений инженеров и ученых.

Стандартизация ведет к снижению себестоимости продукции, поскольку:

- позволяет экономить время и средства за счет применения уже разработанных типовых ситуаций и объектов;
- повышает надежность изделия или результатов расчетов, поскольку применяемые технические решения уже неоднократно проверены на практике;
- упрощает ремонт и обслуживание изделий, так как стандартные узлы и детали - взаимозаменяемые (при условии, что сборка осуществлялась без пригоночных операций).

Эффективность стандартизации может быть незначительной или же вообще отрицательной в случае, когда основной целью разработки изделия является достижение очень высоких функциональных характеристик. Это также заметно в процессе оптимального проектирования, когда наилучшие значения параметров могут не соответствовать стандартным значениям. Различают стандартизацию

конструктивную и технологическую. Конструктивная стандартизация подразумевает использование стандартных конструктивных решений параметров, деталей и узлов. Технологическая стандартизация основана на применении стандартного инструмента и оборудования, технологического процесса.

## **2. Технологичность.**

Возможность изготовления изделия — обязательное условие реализуемости конструктивного решения. Технологичность же заключается в возможности выпуска изделия (изготовления и сборки с заданным уровнем качества) с наименьшими производственными затратами и в кратчайшие сроки. Технологичность закладывается в конструкцию при соответствующем назначении параметров деталей (материала, размеров и их отклонений, шероховатости и т.п.), форм и взаимного расположения поверхностей их элементов. Технологичность базируется на стандартизации, унификации и преемственности. Во многих случаях только возможности технологии (воплощающей в себе достижения науки и техники) позволяют достичь уникальных результатов и высоких потребительских свойств.

Технологичность — понятие относительное, на нее влияют место изготовления, серийность, доступное оборудование, привлекаемый персонал, развитость межхозяйственных связей и многое другое.

## **3. Транспортабельность.**

Транспортабельность — это свойство изделия, позволяющее с минимальными затратами перемещать его в пространстве (внутри производственных цехов, от производителя к продавцу и, далее, к потребителю). Сложности с транспортировкой возникают у изделий громоздких и тяжелых, чувствительных к перемещению (хрупких, высокоточных и т.п.).

Для тяжелых изделий предусматривают возможность такелажирования, т.е. наличия приспособлений для подъема и перемещения груза. При этом конструкция изделия должна быть прочной и жесткой, допускающей его подъем и перенос, а такелажные приспособления — не увеличивать размеры изделия и не ухудшать его вид. Транспортировка изделий, чувствительных к перемещению, требует применения специальных приспособлений и тары.

Размеры и формы изделия должны проектироваться, в том числе, под возможный способ его доставки на место назначения, что подразумевает:

- ориентацию на определенный вид транспорта;
- если габариты уменьшить нельзя, то конструкцию выполняют составной с последующей сборкой на месте эксплуатации посредством простых монтажных операций (принцип агрегатности) и с минимумом затрат на регулировку;
- обеспечение жесткости и виброустойчивости;
- введение специальных частей для стопорения изделия и осуществления погрузочно-разгрузочных (такелажных) работ;

- защиту полостей и выступающих частей, предохранение от коррозии;
- создание соответствующей упаковки.

#### **4. Сохраняемость.**

Сохраняемость - это способность изделия не зависеть (быть защищенным) от неблагоприятных воздействий внешней среды (климатических, случайных или преднамеренных). Достигается нанесением покрытий или использованием упаковки (тары). Форма изделия должна обеспечивать минимальные габариты и простоту формы тары, возможность легкого удаления защитных покрытий, необходимую степень герметичности и использования объема внутри тары.

### **1.3.3. Эксплуатационные требования**

#### **1. Функциональные требования.**

Функциональные требования (показатели назначения) характеризуют наиболее важные свойства товара, способность использовать его по назначению. Эти требования можно разделить на следующие группы:

- *требования производительности* (Включают показатели необходимой мощности, грузоподъемности, развиваемой скорости и другие, которые характеризуют выполняемую функцию);
- *требования эффективности* (характеризуют степень эффективности использования изделия по назначению, например, показатели энергетические (КПД, потери), кинематические (точность перемещения), силовые (стабильность нагрузки) и т.п.);
- *конструктивные требования* (характеризуют достоинства выбранной конструкции, например, масса и габариты).

Функциональные требования обязательно должны присутствовать в ТЗ. Часто их приводят в виде относительных величин.

#### **2. Надежность.**

Надежность — это свойство изделия сохранять во времени в установленных пределах значения всех параметров, которые характеризуют способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, ремонтов, хранения и транспортирования. Степень надежности закладывается на этапах проектирования, изготовления и сборки.

С течением времени значения параметров изменяются. Если изменения некоторых параметров превышают допустимые пределы, то первоначально исправное изделие, состояние которого соответствовало всем требованиям проектной документации, становится неисправным. При этом, если показатели назначения остались в установленных пределах (изделие и дальше способно выполнять заданные функции), то состояние оценивается как работоспособное. Событие, ко-

торое вызвало нарушение исправного состояния изделия при сохранении его работоспособности, называется повреждением. Так, например, изделие может быть работоспособным, но неисправным при ухудшении внешнего вида, не препятствующем его дальнейшему применению по назначению.

Состояние изделия, которое влечет за собой временное или окончательное прекращение его применения по назначению, называется предельным. Такое состояние возникает из-за нарушения требований техники безопасности, недопустимого снижения эффективности эксплуатации, морального старения и других причин. Временное прекращение применения изделия вызывается отправкой его в капитальный или средний ремонт.

Безотказность — свойство изделия непрерывно сохранять работоспособное состояние в течение некоторого времени или наработки. Под наработкой понимается продолжительность или объем выполняемой работы. Долговечность — свойство изделия сохранять работоспособное состояние до наступления предельного состояния при установленной системе технического обслуживания и ремонта. Безотказность используется для характеристики надежности неремонтируемых изделий (например, электролампы) или изделий, отказ которых вызывает катастрофические последствия (например, ущерб от выхода из строя гидротурбины). Долговечность характеризует большой период времени нормальной работы изделия, поскольку предполагается восстановление его работоспособности, но в продолжительность работы остановки на ремонт не засчитываются.

При характеристике надежности изделия предполагается определенная вероятность достижения предельного состояния отдельными его частями. Вероятность того, что в пределах заданной наработки или заданного отрезка времени отказ не возникает, составляет основной показатель теории надежности — вероятность безотказной работы  $P(t)$ . Этот показатель зависит от большого числа параметров (случайных факторов) и изменяется во времени в соответствии с некоторым законом распределения. Надежность связана с экономическими показателями изделия и сроком его эксплуатации: повышение надежности достигается увеличением затрат на изготовление и проектирование, но снижает расходы на ремонт и обслуживание, потери от простоя.

Надежность изделий изменяется и в перерывах между периодами эксплуатации (например, старение материалов). Способность изделий противостоять отрицательному влиянию на безотказность, ремонтпригодность и долговечность условий и продолжительности хранения и транспортирования характеризуется сохраняемостью. Основным ее показателем является средний срок сохраняемости (гамма-процентный срок), который включает срок сохраняемости в упаковке или законсервированном виде и срок монтажа. Продолжительное хранение и транспортирование могут отрицательно повлиять на поведение изделия не

только в этот период, но и при последующей его эксплуатации.

### **3. Эргономичность.**

На основе сочетания возможностей человека и машины эргономичность содействует созданию изделий с высокими социальными результатами: сохранение здоровья людей посредством повышенного удобства эксплуатации, всестороннее развитие человеческой личности (человек — не придаток машины, а машина — помощник человека, глубже реализующая его способности), обеспечение высоких потребительских свойств и эффективности деятельности человека.

Важной является задача не подгонять проектируемый объект под человека, а сразу проектировать человеко-машинную систему. Изделие должно быть эргономичным не только в процессе эксплуатации, но и в процессе изготовления и сборки.

### **4. Безопасность.**

Безопасность предусматривает исключение возможных несчастных случаев при нормальной и неквалифицированной работе, при случайных действиях человека и воздействии внешней среды, в аварийных и экстремальных ситуациях, а также в процессе изготовлении изделия (на обычном и, особенно, опасном производстве).

Безопасность повышается при установке или встраивании систем предохранения, блокировки, предупреждения и т.д.

Безопасность подразумевает безопасность самого изделия и используемых расходных материалов, пожаро- и электробезопасность, отсутствие токсичности, учет особенностей работы (высокая температура, давление) и т.д.

Изделие должно быть безопасным в течение всего жизненного цикла и установленного срока службы. При этом Декларация Генеральной Ассамблеи ООН 1985 г. предусматривает, что изделия, предназначенные для потребления отдельными гражданами, должны быть безопасны не только при обычных условиях их использования по назначению, транспортирования, хранения и утилизации, но также и при любом ином разумном использовании.

Минимально приемлемый уровень безопасности устанавливают нормативные документы федеральных органов исполнительной власти, осуществляющих контроль за качеством и безопасностью товаров, такие как Госгортехнадзор, Госсанэпидемнадзор и другие. В ряде случаев для эксплуатации изделия или при использовании отдельных видов производства необходимо получение соответствующих лицензий и разрешений, наличие инструкций по эксплуатации и действию при аварийных ситуациях.

### **5. Экологичность.**

Экологичность характеризует приспособленность разработанного изделия к сосуществованию с окружающей природой и средой обитания живых организмов, к обмену с ними энергией (например, отдача в окружающее пространство

тепла), веществом (например, засорение среды продуктами износа, утечками смазочных масел) и сигналами (например, издавание свиста, шума). Изделие также должно быть экологичным в процессе производства и утилизации: безвредные отходы, технологические процессы, расходные материалы. Идеальным является изделие, полностью изолированное от внешней среды или работающее по замкнутому циклу (по отношению к отходам и потерям). Ряд требований экологичности регламентируется ГОСТ и СНиП (санитарные нормы и правила).

С каждым годом роль требований экологичности все более возрастает, увеличиваются налоги, отчисления и штрафы за загрязнение окружающей среды, на восстановление природных ресурсов. Предполагают, что скоро эти требования станут одними из самых важных.

## **6. Эстетичность.**

Эстетичность характеризует проявление прекрасного во внешних образах изделия. Обеспечивает социальную эффективность новой продукции. С другой стороны, по степени изящности изделия судят о степени совершенства самой конструкции. Эстетичность достигается методами *дизайна* (художественного проектирования), цель которого — включить разрабатываемое изделие в круг вещей как элемент человеческой культуры, расширить его функции не только на удовлетворение материальных потребностей, но и на потребности духовной жизни, достичь единства формы (художественного восприятия) и содержания (функционального назначения и технической целесообразности).

## **7. Утилизация.**

Утилизация характеризует способы ликвидации изделия по завершении его эксплуатации. Включает два основных этапа: демонтаж и утилизация.

На этапе демонтажа возможны разборка изделия на части или удаление его целиком. Обычно разборка предусматривается с целью удобства последующего транспортирования, либо извлечения из изделия деталей и узлов (впоследствии используемых в качестве запасных частей), либо разделения на однородные фракции (неметаллы, черные и цветные металлы и т.д. для удобства реализации как вторичного сырья).

Требования утилизации должны предусматривать, где и как будут собраны, а затем переплавлены, или захоронены, или уничтожены и т.д. отдельные части или все изделие в целом при целенаправленных действиях, а также при неорганизованных, но удобных для потребителя поступках. Наиболее целесообразно выведение изделия из эксплуатации, используя его в качестве вторичных ресурсов (для той же или других областей промышленности и хозяйства), т.е. чтобы жизненный цикл был замкнутым (сырье — производство — эксплуатация — утилизация — сырье -...). В ином случае изделие должно самоуничтожаться естественным образом, не нарушая экологии.

## 1.4. Технологии проектирования информационных систем

### 1.4.1. Классы технологий проектирования ИС

Под **проектом ИС** будем понимать проектно-конструкторскую и технологическую документацию, в которой представлено описание проектных решений по созданию и эксплуатации ИС в конкретной программно-аппаратной среде.

Под **проектированием ИС** будем понимать процесс преобразования входной информации об объекте и методах проектирования, об опыте проектирования аналогичных объектов в соответствии с ГОСТом в проект ИС. Тогда проектирование ИС будет сводиться к последовательной формализации проектных решений на различных стадиях ЖЦ ИС.

Осуществление проектирования ИС предполагает использование проектировщиками определенной **технологии проектирования**, соответствующей масштабу и особенностям разрабатываемого проекта.

**Технология проектирования** – это совокупность **методологии** и **инструментальных средств** проектирования ИС, а также методов и средств **организации проектирования** (рис.1.10).

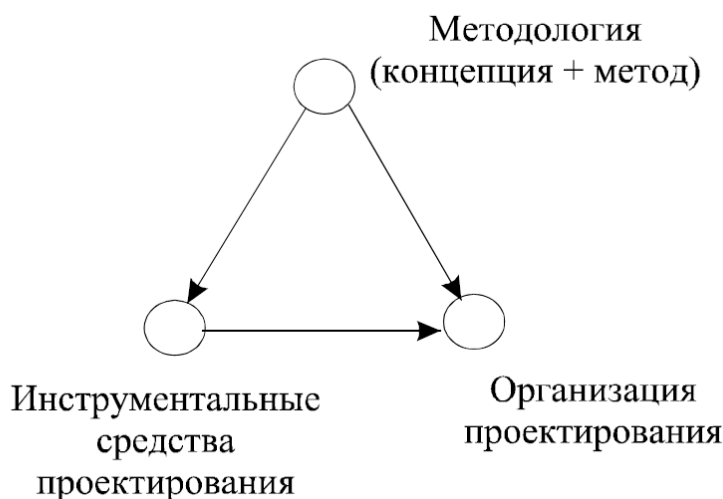


Рис. 1.10. Состав компонентов технологии проектирования

В основе технологии проектирования лежит технологический процесс, который определяет действия, их последовательность, состав исполнителей, средства и ресурсы, требуемые для выполнения этих действий. Предметом любой выбираемой технологии проектирования должно служить отражение взаимосвязанных процессов проектирования на всех стадиях жизненного цикла ИС.

Основу технологии проектирования ИС составляет методология, которая определяет сущность, основные отличительные технологические особенности.

**Методология проектирования** предполагает наличие некоторой концепции, принципов проектирования, реализуемых набором методов проектирования, которые должны поддерживаться некоторыми средствами проектирования.



**Организация проектирования** предполагает определение методов взаимодействия проектировщиков между собой и с заказчиком в процессе создания проекта ИС, которые могут также поддерживаться набором специфических средств.

Методы проектирования ИС можно классифицировать по степени использования средств автоматизации, типовых проектных решений, адаптивности к предполагаемым изменениям.

По *степени автоматизации* методы проектирования разделяются на методы: *ручного проектирования*, при котором проектирование компонентов ИС осуществляется без использования специальных инструментальных программных средств;

*компьютерного проектирования*, которое генерирует проектные решения на основе использования специальных инструментальных программных средств.

По *степени использования* типовых проектных решений различают методы: *оригинального проектирования*, когда проектные решения разрабатываются с «нуля» в соответствии с требованиями к ИС;

*типового проектирования*, предполагающего конфигурацию ИС из готовых типовых проектных решений (программных модулей).

По *степени адаптивности проектных решений* выделяют методы:

*реконструкции*, когда адаптация проектных решений выполняется путем переработки соответствующих компонентов;

*параметризации*, когда проектные решения настраиваются (перегенерируются) в соответствии с изменяемыми параметрами;

*реструктуризации* модели, когда изменяется модель проблемной области, на основе которой автоматически регенерируются проектные решения.

Сочетание различных признаков классификации методов проектирования обуславливает характер используемой технологии проектирования ИС, среди которых выделяются два основных класса: каноническая и промышленная технологии. Промышленная технология, в свою очередь, разбивается на автоматизированное (с использованием CASE-технологий) и типовое (параметрически или модельно-ориентированное) проектирование (табл.1.1).

Таблица 1.1.

**Характеристики классов технологий проектирования**

Класс технологии проектирования	Степень автоматизации	Степень типизации	Степень адаптивности
Каноническое проектирование	Ручное проектирование	Оригинальное проектирование	Реконструкция
Промышленное автоматизированное проектирование	Компьютерное проектирование	Оригинальное проектирование	Реструктуризация модели (генерация ИС)
Промышленное типовое проектирование	Компьютерное проектирование	Типовое сборочное проектирование	Параметризация и реструктуризация модели ((конфигурация ИС)

**Инструментальные средства проектирования** ИС применяются для конкретных видов технологий проектирования.

Средства проектирования ИС можно разделить на два класса: без использования ЭВМ и с использованием ЭВМ.

**Средства проектирования ИС без использования ЭВМ** применяются на всех стадиях и этапах проектирования. К ним относятся средства организационно-методического обеспечения операций проектирования; стандарты, регламентирующие процесс проектирования систем; единая система классификации и кодирования информации; унифицированная система документации; модели описания и анализа потоков информации и т.п.

**Средства проектирования ИС с использованием ЭВМ могут** применяться как на отдельных, так и на всех стадиях и этапах проектирования ИС и соответственно поддерживают разработку элементов проекта системы, разделов проекта системы, проекта системы в целом.

Средства проектирования с использованием ЭВМ делят на четыре подкласса: средства поддержки проектирования операций обработки информации; средства поддержки проектирования отдельных компонентов проекта ИС; средства поддержки проектирования разделов проекта ИС; средства поддержки разработки проекта на стадиях и этапах проектирования.

К **первому подклассу** относятся алгоритмические языки, библиотеки стандартных подпрограмм и классов объектов, макрогенераторы, генераторы программ типовых операций обработки данных, системные утилиты, а также простейшие инструментальные средства проектирования (средства для тестирования и отладки программ, поддержки процесса документирования проекта и т.п.)

Ко **второму подклассу** относятся средства общесистемного назначения: СУБД, методоориентированные пакеты прикладных программ (ППП), текстовые и графические редакторы, табличные процессоры, статистические и интегрированные пакеты прикладных программ, оболочки экспертных систем.

К **третьему подклассу** относят функциональные средства проектирования, направленные на разработку АС, реализующих функции, комплексы задач и задачи управления. К ним относят типовые проектные решения, функциональные пакеты прикладных программ, типовые проекты.

К **четвертому подклассу** относятся средства автоматизации проектирования ИС (CASE-средства).

Рассмотрим ниже каноническое и типовое проектирование ИС. Автоматизированному проектированию ИС посвящен раздел 2.

#### **1.4.2. Каноническое проектирование ИС**

Каноническое проектирование ИС направлено на отражение особенностей

технологии индивидуального (оригинального) проектирования. Среди основных характерных особенностей канонического проектирования можно выделить такие особенности, как [38]: отражение особенностей ручной технологии проектирования; ориентация на индивидуальное (оригинальное) проектирование; осуществление на уровне исполнителей; возможность интеграции выполнения элементарных операций; применение, как правило, для сравнительно небольших, локальных ИС; использование инструментальных средств универсальной компьютерной поддержки.

Каноническое проектирование направлено на минимальное использование типовых проектных решений. Адаптация проектных решений при каноническом проектировании осуществляется только путем перепрограммирования соответствующих программных модулей. Организация канонического проектирования ИС основана на использовании каскадной модели жизненного цикла и предусматривает набор определенных стадий и этапов. Принцип деления процесса проектирования на стадии и этапы направлен на то, чтобы проектировать систему «сверху-вниз» и постепенно разрабатывать изначально укрупненные, затем детализированные – проектные решения. Поскольку объекты автоматизации имеют различную сложность и набор задач для создания решения для конкретной ИС, стадии и этапы работ также могут различаться по трудоемкости: существует возможность объединять последовательные этапы, исключать определенные из них на любой стадии проекта, а также до окончания предыдущей стадии начинать выполнение следующей. Стадии и этапы разработки ИС, которые выполняют организации-участники, оформляются в договорах и технических заданиях на выполнение работ.

**Каноническое проектирование** основано на ряде стандартов, таких, как:

1. ГОСТ 34.003 – термины и определения основных понятий в области автоматизированных систем.
2. ГОСТ 34.201 – виды, комплектность и обозначение документов при создании автоматизированных систем.
3. ГОСТ 34.601 – стадии создания автоматизированных систем.
4. ГОСТ 34.602 – техническое задание на создание ИС.
5. ГОСТ 34.603 – виды испытаний автоматизированных систем.
6. РД 50-34.698 – требования к содержанию документов.
7. ГОСТ 2.105 – общие требования к текстовым документам.

По отношению к проекту разработки ИС можно выделить 3 укрупненные стадии проектирования:

- предпроектную (стадии 1-3);
- проектную (стадии 4-6);
- послепроектную (стадии 7-8).

Стадии и этапы создания ИС, выполняемые организациями-участниками, фиксируются в договорах и технических заданиях на выполнение работ.

**Предпроектная** стадия направлена на предпроектное обследование и разработку технического задания на ИС. Характерными результатами этого этапа являются: определение целей и задач системы, формирование общих требований к ее созданию, разработка программы проведения обследования, в ходе которого должны быть изучены структура и процессы организации, модель управления, задачи, подлежащие автоматизации, технико-экономические характеристики, ориентировочный состав технических средств. Ниже представлен перечень этапов работ (стадий), в соответствии с ГОСТ 34.601.

### ***Стадия 1. Формирование требований к ИС:***

- обследование объекта и обоснование необходимости создания ИС;
- формирование требований пользователя к ИС;
- оформление отчета о выполненной работе и заявки на разработку ИС.

На этапе обследования целесообразно выделить две составляющие: *определение стратегии внедрения ИС* и *детальный анализ деятельности организации*.

Основная задача ***определения стратегии внедрения ИС*** – оценка реального объема проекта, его целей и задач на основе выявленных функций и информационных элементов автоматизируемого объекта высокого уровня. Результатом этапа определения стратегии является документ (технико-экономическое обоснование проекта), где четко сформулировано, что получит заказчик, когда он получит готовый продукт (график выполнения работ) и сколько это будет стоить (для крупных проектов должен быть составлен график финансирования на разных этапах работ).

При ***детальном анализе деятельности организации*** изучаются задачи, обеспечивающие реализацию функций управления, организационная структура, штаты и содержание работ по управлению предприятием, а также характер подчиненности вышестоящим органам управления.

На этапе ***обследования*** следует классифицировать планируемые функции системы по степени важности. Один из возможных форматов представления такой классификации – MuSCoW. Эта аббревиатура расшифровывается так:

- **Must have** - необходимые функции;
- **Should have** - желательные функции;
- **Could have** - возможные функции;
- **Won't have** - отсутствующие функции.

Функции *первой категории* обеспечивают критичные для успешной работы системы возможности. Реализация функций *второй и третьей категорий* ограничивается временными и финансовыми рамками: разрабатывается то, что необходимо, а также максимально возможное в порядке приоритета число функций второй и третьей категорий. *Четвертая категория* функций особенно важна, поскольку необходимо четко представлять границы проекта и набор функций, которые будут отсутствовать в системе.

## **Стадия 2. Разработка концепции ИС:**

- изучение объекта;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
- оформление отчета о проделанной работе.

## **Стадия 3. Техническое задание (ТЗ):**

- разработка и утверждение технического задания на создание ИС.

Важным документом, фиксирующим результаты определения стратегии внедрения ИС, является технико-экономическое обоснование проекта. В этом документе должно быть четко определены результаты выполнения проекта для заказчика, а также указаны графики выполнения работ и график финансирования на разных этапах выполнения проекта. Дополнительно в таком документе отражаются сроки, время окупаемости проекта, ожидаемые выгода и экономический эффект проекта.

Ориентировочно технико-экономическое обоснование содержит:

- все риски и ограничения, влияющие на успешность проекта;
- условия эксплуатации будущей системы: архитектурные, программные, аппаратные требования, требования к компонентам ПО и СУБД;
- пользователи системы;
- функции, выполняемые системой;
- интерфейсы и распределение функций между человеком и системой;
- сроки завершения этапов, форма приемки/сдачи работ;
- рамки проекта;
- возможности развития системы.

По результатам обследования формируется **техническое задание на ИС**.

В соответствии с ГОСТ 34.602-89, **ТЗ** – основной документ, определяющий требования и порядок создания (развития или модернизации) АС, в соответствии с которым проводится разработка ИС и ее приемка при вводе в действие.

Разработка ТЗ предусматривает описание следующих разделов:

- общие сведения;
- назначение и цели создания (развития) системы;
- характеристика объектов автоматизации;
- требования к системе;
- состав и содержание работ по созданию системы;
- порядок контроля и приемки системы;
- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

- требования к документированию;
- источники разработки.

**Проектная** стадия главным образом ориентирована на разработку технического и рабочего проектов. Процесс разработки ТЗ включает обследование объекта автоматизации (организации или подразделения) и его систем управления. Для решения задач информационного обеспечения необходимо проанализировать информационные потоки, формы документации, системы кодирования, а также все связанное со структурой баз данных и СУБД, что определяет состав исходных технологических требований.

#### ***Стадия 4. Эскизный проект:***

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ИС и ее части.

Если для ИС конкретного объекта автоматизации проектные решения выбраны ранее или являются очевидными, стадия эскизного проекта может быть исключена из последовательности работ. Таким образом, эта стадия не является строго обязательной.

На этапе эскизного проекта, в том числе, должны быть определены:

- цели, функции ИС и подсистем;
- состав комплексов задач и отдельных задач;
- концепция и структура информационной базы;
- функции СУБД;
- функции и параметры основных программных средств;
- ожидаемый эффект от ее внедрения.

Документация, содержащая результаты работ по совокупности принятых проектных решений, согласовывается, утверждается и используется в дальнейшем для выполнения работ по созданию ИС. На основании ТЗ (в том числе, при наличии эскизного проекта) разрабатывается технический проект ИС.

#### ***Стадия 5. Технический проект:***

- разработка проектных решений по системе и ее частям;
- разработка документации на ИС и ее части;
- разработка и оформление документации на поставку изделий для комплектования ИС и (или) технических требований (ТЗ) на их разработку;
- разработка заданий на проектирование в смежных частях проекта объекта автоматизации.

На этапе технического проекта проводятся работы научно-исследовательского и экспериментального характера для выбора основных проектных решений, а также рассчитывается экономическая эффективность системы.

Важным аспектом разработки **технического проекта** является анализ всей используемой информации на предмет таких характеристик, как полнота, отсутствие дублирования и избыточности, непротиворечивость и т.д., а также определение форм выходных документов. Документация должна быть оформлена в соответствии с требованиями ГОСТ 34-201 и РД 50-34.698.

#### ***Стадия 6. Рабочая документация:***

- разработка рабочей документации на систему и ее части;
- разработка или адаптация программ.

Один из основных этапов стадии **рабочего проектирования** – разработка рабочей документации на информационное обеспечение ИС, в состав которой входят: технический проект ИС, описание баз данных, перечень исходных и выходных данных и документов. Стадия технического проектирования завершается подготовкой и оформлением документации на поставку для комплектования ИС и определением технических требований и составлением ТЗ на разработку ИС.

Стадия «Рабочая документация» предполагает создание, как программного продукта, так и всей сопровождающей документации, которая должна предоставлять все сведения, обеспечивающие выполнение работ на стадиях ввода ИС в действие и эксплуатации ИС, в том числе, сведения для поддержания уровня качества ИС (соблюдения эксплуатационных характеристик).

**Послепроектная** стадия включает в себя реализацию мероприятий по внедрению, подготовку помещений и технических средств, обучение персонала. Также производится эксплуатация системы с решением конкретных задач, анализируются результаты испытаний, реализуются мероприятия по сопровождению ИС.

#### ***Стадия 7. Ввод в действие:***

- подготовка объекта автоматизации к вводу ИС в действие;
- подготовка персонала;
- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);
- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

Основными видами испытаний для ИС являются такие, как: предварительные испытания, опытная эксплуатация и приемочные испытания, которые при необходимости могут быть расширены дополнительными испытаниями ИС и ее составляющих частей.

В ходе предварительных испытаний, регламентируемых соответствующей программой и методикой, главным образом проводятся испытания системы на работоспособность и соответствие ТЗ, а также устранение неисправностей и внесение изменений в документацию на ИС.

На следующем этапе происходит процесс проведения опытной эксплуатации, анализируются ее результаты, и при необходимости проводится доработка ПО и дополнительная наладка технических средств ИС.

В процессе проведения приемочных испытаний реализуются испытания на соответствие ТЗ, анализируются результаты испытания системы, устраняются недостатки, которые были выявлены при испытаниях.

При всех видах испытаний оформляются соответствующие акты о приемке системы в опытную эксплуатацию, ее завершении и о приемке системы в постоянную эксплуатацию.

### ***Стадия 8. Сопровождение ИС:***

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Основными процессами этой стадии являются осуществление работ по устранению недостатков, выявленных при эксплуатации системы в течение гарантийных сроков, а также анализ функционирования системы, выявление отклонений и их причин, устранение причин отклонений и недостатков, обеспечение стабильности эксплуатационных характеристик.

## **1.4.3. Типовое проектирование ИС**

Методы типового проектирования направлены на выполнение проектирования ИС с использованием типовых проектных решений.

**Типовое проектное решение (ТПР)** – проектное решение, пригодное к многократному использованию (тиражируемое проектное решение).

Применение методов типового проектирования имеет свои особенности. Основным условием для использования таких методов является возможность декомпозиции проектируемой ИС на составляющие компоненты (подсистемы, программные модули, комплексы выполняемых задач и т.д.), для реализации которых можно выбрать типовые проектные решения, существующие на рынке, которые будут настроены на нужды конкретного предприятия [38, 42].

Помимо собственно функциональных (программных, аппаратных) элементов, типовое решение подразумевает наличие необходимой документации, в которой дается детальное описание ТПР (в т.ч. процедур настройки), отвечающее требованиям проектируемой системы.

По уровню декомпозиции системы можно выделить такие классы ТПР, как:

- элементные ТПР – ТПР по отдельному элементу (задаче, виду обеспечения);



- подсистемные ТПР – ТПР по отдельным подсистемам;
- объектные ТПР – отраслевые ТПР, включающие весь набор подсистем ИС.

Выделенные классы ТПР имеют свои достоинства и недостатки. Рассмотрим наиболее характерные из них.

К достоинству **элементных ТПР** можно отнести реализацию модульного подхода к проектированию ИС. В то же время это приводит к большим затратам на доработку ТПР конкретных элементов и к затратам на объединение разных элементов вследствие их несовместимости.

**Подсистемные ТПР** также позволяют реализовать модульный подход к проектированию ИС. Кроме того, они позволяют осуществлять параметрическую настройку компонентов на объекты различных уровней управления; взаимосвязанные компоненты и высокая степень интеграции элементов ИС приводят к минимизации затрат на проектирование и программирование. Однако в случае нескольких производителей программного обеспечения появляются проблемы в объединении различных функциональных подсистем; помимо этого, с точки зрения непрерывного реинжиниринга процессов адаптивность ТПР является недостаточной.

**Объектные ТПР** имеют такие преимущества, как:

- масштабируемость (допускаются конфигурации ИС для различного числа рабочих мест);
- методологическое единство компонентов ИС;
- совместимость компонентов ИС;
- открытость архитектуры (возможность развертывания ТПР на платформах разного типа);
- конфигурируемость (возможность использовать необходимое подмножество компонентов системы).

К недостаткам объектных ТПР можно отнести проблемы реализации типового проекта в оригинальном объекте управления, что приводит в определенных ситуациях к необходимости смены организационной структуры объекта автоматизации.

При реализации типового проектирования применяются такие подходы, как параметрически-ориентированное и модельно-ориентированное проектирование.

Этапами **параметрически-ориентированного** проектирования являются:

- постановка задач и определение пригодности пакетов прикладных программ (ППП) для их решения через систему критериев оценки;
- анализ доступных пакетов прикладных программ, исходя из критериев;
- выбор и приобретение подходящего ППП;
- настройка параметров приобретенного ППП.

Среди критериев оценки ППП выделяют [32] следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета;
- факторы финансового порядка;
- особенности установки пакета;
- особенности эксплуатации пакета;
- помощь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Отметим, что каждая из перечисленных групп критериев может быть детализирована на совокупность частных показателей, дающих дополнительную информацию для каждого аспекта анализа выбранного ППП. Значения критериев определяются с использованием методов экспертного оценивания.

Другим подходом реализации типового проектирования является **модельно-ориентированное проектирование**, сущность которого состоит в адаптации существующих характеристик типовой ИС, исходя из модели объекта автоматизации, построение которой предполагает использование специального программного инструментария.

При таком подходе технология проектирования должна иметь средства как для работы с моделью конкретного предприятия, так и с моделью типовой ИС.

В репозитории типовой ИС содержится модель объекта автоматизации, которая является основой для конфигурирования программного обеспечения. Кроме того, в репозитории содержится базовая (ссылочная) модель ИС и типовые (референтные) модели ее определенных классов.

Базовая модель ИС описывает процессы, организационную структуру, объекты, функции, для поддержки которых предназначены программные модули типовой ИС. Типовые модели предназначены для описания конфигурации ИС для тех или иных отраслей, типов производства.

Модель конкретного предприятия может быть построена либо в результате выбора фрагментов типовой модели с учетом особенностей объекта автоматизации (BAAN Enterprise Modeler), либо с использованием автоматизированной адаптации этих модулей с учетом мнений экспертов (SAP Business Engineering Workbench). Модель предприятия, на основе которой осуществляется автоматическое конфигурирование и настройка ИС, хранится в репозитории и может быть откорректирована в случае необходимости.

Внедрение типовой ИС начинается с анализа результатов предпроектного обследования предприятия, сформированных в виде требований к конкретной ИС, для оценки которых может быть использована методика оценки ППП.

На следующем этапе необходимо построить предварительную модель ИС, которая должна полно отражать особенности реализации ИС для конкретного объекта автоматизации. Предварительная модель – основа для выбора типовой модели системы, а также для формирования перечня компонентов, для реализации которых потребуются другие программные средства или инструментальные средства, имеющиеся в составе типовой ИС.

При реализации типового проекта выполняются следующие операции [32]:

- установка глобальных параметров системы;
- задание структуры объекта автоматизации;
- определение структуры основных данных;
- задание перечня реализуемых функций и процессов;
- описание интерфейсов; описание отчетов; настройка авторизации доступа; настройка системы архивирования.

Типовое проектирование широко представлено в современных средствах.

#### **1.4.4. Управление процессом проектирования**

Правильное ведение документации облегчает взаимодействие участников процесса проектирования. Осознанный подход к проектной деятельности позволяет быстро находить эффективные решения и управлять этой деятельностью.

В настоящее время на основе исследований сущности процесса проектирования разработаны рекомендации по ведению этой деятельности. Предложен ряд структур и алгоритмов проектирования, совпадающих в основных чертах и различающихся только в содержании или названии отдельных этапов. В результате их анализа и обобщения предложена структура, представленная на рис. 1.10.

Решение любой задачи начинается с ее осмысления и уточнения исходных данных. Технические требования (ТТ), которые выдаются заказчиком, формулируются на языке потребителя-неспециалиста и не всегда бывают технически четкими и исчерпывающими. Перевести требования на язык предметной области, сформулировать задачу максимально полно и грамотно, обосновать необходимость ее решения, т.е. сформулировать ТЗ – первый и обязательный этап работы. Исполнитель выполняет его в тесном контакте с заказчиком.

На этапе *синтеза принципа действия* отыскивают принципиальные положения, физические, социальные и т.п. эффекты, которые составят основу функционирования будущего изделия. Это могут быть основополагающие нормы, фундаментальные законы и правила, их частные случаи или следствия. Работа ведется с принципиальными моделями и их графическим представлением — блок-схемами. Этому этапу соответствует заключительная стадия ТЗ и стадия технического предложения структуры проектирования по ГОСТ 2.103.

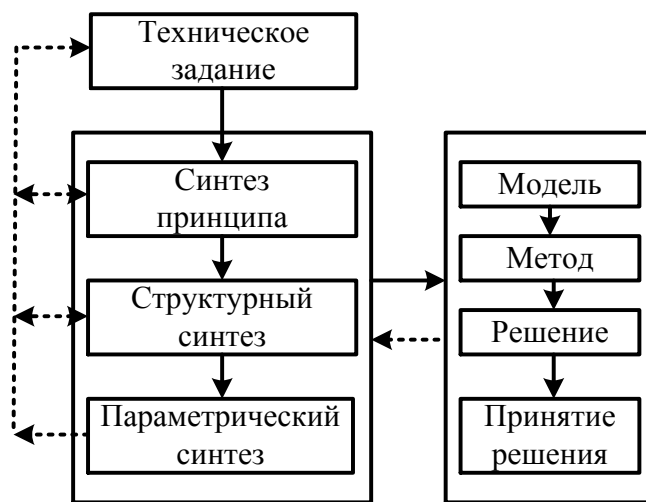


Рис. 1.10. Структура процесса решения задачи проектирования

На этапе **структурного синтеза** на основе выбранного принципа действия создаются варианты начального графического представления объекта - структуры, схемы, алгоритмы, упрощенные эскизы. В соответствии с ГОСТ 2.103 этот этап включает стадию эскизного проектирования.

На этапе **параметрического синтеза** отыскиваются значения параметров объекта, находится численное решение проектной задачи, создается подробная документация или описание объекта, чертежи изделия и его частей. Этот этап соответствует стадиям технического и рабочего проектирования.

Вследствие неполноты начальных знаний процесс проектирования итерационен, что на рис. 1.10 отражается стрелками обратных движений.

Замечено, что эффективность проектируемого объекта определяется: в первую очередь — выбранным принципом действия, во вторую — предложенной структурой и в третью — соотношением параметров.

Ведение разработки последовательно от общих черт проектируемого объекта к детальным частным называется нисходящим проектированием. Его результатом будут требования к отдельным частям и узлам.

Возможен ход разработки от частного к общему, что образует процесс восходящего проектирования. Такое проектирование встречается, если одна или несколько частей уже являются готовыми (покупными или уже разработанными) изделиями. Результатом проектирования будет частная документация на узлы.

Нисходящее и восходящее проектирование обладают своими достоинствами и недостатками. Так, при **нисходящем проектировании** возможно появление требований, впоследствии оказывающихся нереализуемыми по технологическим, экологическим или иным соображениям. При **восходящем проектировании** возможно получение объекта, не соответствующего заданным требованиям. В реальной жизни, вследствие итерационного характера проектирования, оба его вида взаимосвязаны.

## 2. АВТОМАТИЗИРОВАННОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

### 2.1. Автоматизация проектирования ИС на основе CASE-технологии

#### 2.1.1. Методологии моделирования систем семейства IDEF

Методологическую основу автоматизации проектирования ИС составляет *семейство стандартов IDEF* – семейство совместно используемых методов для процесса моделирования. IDEF-технология используется, начиная с конца 1980-х годов. Основные пользователи – Министерство обороны США и некоторые крупные корпорации. Включает 15 основных (методов) IDEF0 – IDEF14.

**IDEF0 (Function Modeling)** – метод для создания функциональной модели, которая является структурированным отображением функций производственной системы или среды, а также информации и объектов, связывающих эти функции.

**IDEF1 (Information Modeling)** – метод для построения информационной модели, которая представляет собой структурированную информацию, необходимую для поддержки функций производственной системы или среды.

**IDEF2 (Simulation Model Design)** – метод позволяет построить динамическую модель меняющегося во времени поведения функций, информации и ресурсов производственной системы или среды. Данная модель используется редко. В основном востребована на предприятиях, где необходимо описать непрерывную деятельность на конвейерах или аналогичные функции.

**IDEF3 (Process Description Capture)** – метод для сбора информации о состоянии моделируемой системы. Это структурный метод, показывающий причинно-следственные связи и события, а также, как организована работа и какие пользователи работают с системой. IDEF3 состоит из двух методов.

**Process Flow Description (PFD)** – описание процессов, с описанием того, как организована работа между различными элементами моделируемой системы.

**Object State Transition Description (OSTD)** – описание переходов состояний объектов, с описанием того, какие существуют промежуточные состояния у объектов в моделируемой системе.

**IDEF4 (Object-Oriented Design)** – метод объектно-ориентированного планирования был разработан для поддержки объектно-ориентированной идеологии. Подробнее - *Технология UML*.

**IDEF5 (Ontology Description Capture)** – метод позволяет разрабатывать, изучать и поддерживать онтологию моделируемой системы. Термин «онтология» включает в себя каталог терминов области знаний; правила, объясняющие, как термины могут комбинироваться, создавая при этом корректные ситуации в области знаний и согласованные выводы, используемые в моделируемой системе.

**IDEF6 (Design Rational Capture Method)** – метод позволяет использовать рациональный опыт проектирования.

**IDEF7 (Information System Auditing)** – метод описывает проведение методологии аудита информационной системы.

**IDEF8 (User Interface Modeling)** – метод для разработки необходимых моделей Графического Интерфейса Пользователя (HumanSystem Interaction Design). Предназначен для проектирования взаимодействия человека и технической системы.

**IDEF9 (Business Constraint Discovery)** – модель предназначена для анализа имеющихся условий и ограничений (в том числе физических, юридических или других) и их влияния на принимаемые решения в процессе реинжиниринга.

**IDEF10 - Implementation Architecture Modeling.**

**IDEF11 - Information Artifact Modeling.**

**IDEF12 - Organization Modeling.**

**IDEF13 - Three Schema Mapping Design.**

**IDEF14 (Network Design)** - метод позволяет *моделировать вычислительные сети*. Модель предназначена для представления и анализа данных при проектировании вычислительных сетей на графическом языке с описанием конфигураций, очередей, сетевых компонентов, требований к надежности.

Наиболее популярные методологии моделирования процессов и их анализа:

- моделирование процессов в нотации IDEF0 предназначено для функционального описания процессов.
- моделирование рабочих потоков в нотации IDEF3 предназначено для описания рабочих процессов.
- моделирование потоков данных в нотации DFD предназначено для описания потоков информации в процессе выполнения работ.

### **2.1.2. Основные понятия и классификация CASE-технологий**

Учитывая постоянный рост требований и сильно растущую сложность современных ИС, можно выделить следующие особенности крупных ИС [26, 32]:

- сложность описания;
- наличие совокупности тесно взаимодействующих компонентов;
- отсутствие прямых аналогов, ограничивающее возможность использования каких-либо типовых проектных решений и прикладных систем;
- необходимость интеграции существующих и разрабатываемых приложений;
- функционирование в неоднородной среде на нескольких аппаратных платформах;
- разобщенность и разнородность отдельных групп разработчиков по уровню

квалификации и сложившимся традициям использования тех или иных инструментальных средств;

- существенная временная протяженность проекта разработки.

Создание всей требуемой проектной документации вручную является крайне сложной задачей, а редактирование созданного пакета документов влечёт за собой ещё большие трудности.

В связи с этим, можно выявить следующие проблемы, возникающие при ручном процессе проектирования:

- неадекватная спецификация требований;
- неспособность обнаруживать ошибки в проектных решениях;
- низкое качество документации, снижающее эксплуатационные качества;
- затяжной цикл и неудовлетворительные результаты тестирования.

Существенное подспорье в решении подобных проблем вносят **CASE-технологии** (Computer Aided System/Software Engineering). Термин CASE в настоящее время используется в довольно широком смысле и охватывает процесс разработки не только программного обеспечения, но и сложных ИС в целом.

CASE-технология в рамках методологии программирования включает в себя *методологии проектирования ИС*, с помощью которых на основе графической *нотации* строятся диаграммы, поддерживаемые *инструментальными средствами* анализа и проектирования ИС.

**Методология** – это процедура генерации описаний компонентов ИС.

**Нотация** – отображение структуры системы, элементов данных этапов обработки с помощью специальных графических символов диаграмм, а также описание проекта системы на формальных и естественных языках.

**Инструментальные CASE-средства** – специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования ИС.

CASE-средство поддерживает следующие процессы создания и сопровождения ИС: анализ и формулировка требований, проектирование прикладного программного обеспечения и баз данных, генерация кода, тестирование, документирование, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

Полная среда разработки ИС представляет из себя совокупность используемых CASE-средств, системного программного обеспечения и технических средств. Следует заметить, что CASE-средства далеко не всегда сразу же дают ожидаемый эффект, кроме того, реальный бюджет, требуемый на внедрение, в большинстве случаев существенно превышает их рыночную стоимость.

Чтобы увеличить шансы на успешное внедрение CASE-средства, необходимо руководствоваться тремя аспектами:

- технология (ограниченность существующих возможностей);

- культура (готовность к внедрению новых процессов);
- управление (четкое руководство важными этапами и процессами внедрения).

Недостаток внимания к какому-либо аспекту может негативным образом сказаться на успешности процесса внедрения, даже, несмотря на скрупулёзное следование существующим рекомендациям.

CASE-средства обладают мощными графическими средствами описания и документирования ИС, обеспечивают управляемость процесса разработки за счёт интеграции некоторых компонентов, а также позволяют централизованно хранить данные при помощи репозитория. Архитектуру CASE-средства можно представить в виде совокупности шести компонентов (рис. 2.1) [26]:

1. Репозиторий данных.
2. Графический редактор диаграмм.
3. Верификатор диаграмм.
4. Генератор отчётов.
5. Администратор проекта.
6. Сервис.

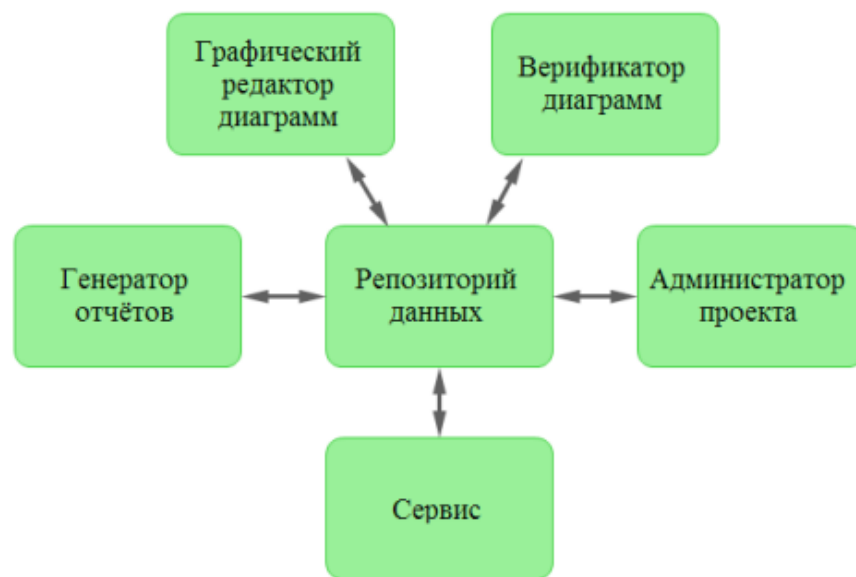


Рис. 2.1. Компоненты CASE-средства

**Репозиторий** представляет собой базу данных, предназначенную для обмена информацией между компонентами CASE-средства, а также для хранения сведений обо всех объектах проектируемой системы.

**Графический редактор** диаграмм предназначен для отображения проектируемой ИС в заданной графической нотации. Выполняет следующие действия:

- создавать элементы диаграмм и взаимосвязи между ними;
- задавать описания элементов диаграмм; задавать описания связей между элементами диаграмм;



- редактировать элементы диаграмм, их взаимосвязи и описания.

**Верификатор диаграмм** выявляет несоответствия разрабатываемой диаграммы методологии проектирование.

Среди его функций можно выделить:

- мониторинг правильности построения диаграмм;
- диагностику и выдачу сообщений об ошибках;
- выделение на диаграмме ошибочных элементов.

**Генератор отчётов** позволяет получать информацию о состоянии проекта в виде, формируемых по различным признакам, отчётов.

**Администратор проекта** представляет собой набор инструментальных средств, необходимых для выполнения административных функций.

К таким функциям относятся:

- инициализация проекта;
- задание начальных параметров проекта;
- назначение и изменение прав доступа к элементам проекта;
- мониторинг выполнения работ.

**Компонент сервиса** представляет собой набор системных утилит для обслуживания репозитория данных. Используется для архивации данных, восстановления данных и создания нового репозитория.

В зависимости от области применения различные CASE-средства могут содержать следующие компоненты:

- репозиторий;
- графические средства анализа и проектирования;
- средства разработки приложений;
- средства конфигурационного управления;
- средства документирования; средства тестирования;
- средства управления проектом; средства реинжиниринга.

Современные CASE-системы классифицируются по следующим признакам:

1) *по поддерживаемым методологиям проектирования*: функционально (структурно)-ориентированные, объектно-ориентированные и комплексно-ориентированные (набор методологий проектирования);

2) *по поддерживаемым графическим нотациям построения диаграмм*: с фиксированной нотацией, с отдельными нотациями, с наиболее распространенными нотациями;

3) *по степени интегрированности*: **tools** (отдельные локальные средства), **toolkit** (набор неинтегрированных средств) **workbench** (полностью интегрированные средства, связанные общей базой проектных данных);

4) *по типу и архитектуре вычислительной техники*: ориентированные на отдельную ЭВМ, на ЛВС, на ГВС, смешанного типа;

5) *по режиму коллективной разработки проекта*: не поддерживающие коллективную разработку, ориентированные на режим реального времени разработки проекта, ориентированные на режим объединения подпроектов;

6) *по типу ОС*: работающие под управлением ОС Windows, под управлением Unix-подобных ОС, кроссплатформенные.

Выделим следующие **типы CASE-средств** [32]:

- *средства моделирования предметной области* (построения и анализа моделей предметной области);
- *средства анализа и проектирования* (создание спецификации компонентов, интерфейсов системы, архитектуры системы, алгоритмов и структур данных);
- *средства проектирования баз данных* (построение моделей данных);
- *средства разработки приложений*;
- *средства реинжиниринга* (анализ программных кодов и схем баз данных и формирование на их основе различных моделей и проектных спецификаций);
- *средства планирования и управления*;
- *средства конфигурационного управления*;
- *средства тестирования*;
- *средства документирования*.

В исторической ретроспективе выделяют CASE-системы двух поколений:

**Первое поколение** обеспечивает:

- поддержку графических моделей;
- проектирование спецификаций;
- проектирование экранных редакторов;
- проектирование словарей данных.

**Второе поколение** обеспечивает:

- поддержку графических представлений требований к системе;
- поддержку представлений спецификаций проектирования;
- поддержку контроля и анализа системной информации,
- информационную поддержку управления проектированием,
- построение прототипов и моделей системы;
- автоматическую кодогенерацию;
- поддержку тестирования, верификации и анализа сгенерированных программ;
- генерацию документов по проекту;
- контроль на соответствие стандартам по всем этапам ЖЦ.

Большая часть подобных технологий основывается на *методологиях структурного и объектно-ориентированного анализа*. Представление полученных данных производится при помощи текстов и диаграмм.

В качестве примера можно выделить следующие популярные CASE-средства:

- коммерческие: Process Modeler (BPwin), Data Modeler (Erwin), ARIS Express (IDS Scheer), BusinessStudio, Rational Rose, Visual Paradigm for UML;
- распространяемые по открытой лицензии: Ramus Educational, StarUML.

### 2.1.3. Прототипная технология быстрой разработки приложений RAD

Одним из возможных подходов к разработке ИС в рамках *спиральной модели* ЖЦ является получивший широкое распространение способ так называемой *быстрой разработки приложений*, или *RAD (Rapid Application Development)*.

Принципы RAD сформулированы в 1980 году сотрудником компании IBM Джеймсом Мартином. При этом методология реализовывалась в кратчайшие сроки небольшой группой разработчиков с использованием инкрементного прототипирования. Это управление стадией проектирования ИС позволяет продемонстрировать заказчику действующую интерактивную модель системы-прототипа, уточнить проектные решения, оценить эксплуатационные характеристики.

В настоящее время методология RAD стала общепринятой схемой для проектирования и разработки ИС. Средства разработки на основе RAD очень популярны за счет использования таких сред разработки: IBM Lotus Domino Designer, Borland Delphi, Borland C++ Builder, Microsoft Visual Studio, Macromedia Flash и др. Быстрая разработка приложений в RAD достигается за счет использования компонентно-ориентированного конструирования и применяется если [38]:

- бюджет проектируемой ИС ограничен;
- нечетко определены требования к ИС;
- требуется реализация проекта ИС в минимальные сроки;
- интерфейс пользователя можно продемонстрировать в прототипе;
- проект можно функционально разделить на составляющие элементы.

Методология RAD имеет *следующие стадии*:

1. Моделирование информационных потоков между функциями.
2. Моделирование данных.
3. Преобразование объектов данных, обеспечивающих реализацию функций.
4. Генерация приложений.
5. Тестирование и объединение.

Итерационное использование прототипного подхода RAD к разработке ИС обеспечивает экономию ресурсов на проектирование и резкое сокращение времени на разработку и внедрение готовой к эксплуатации системы.

Недостатки методологии:

1. Для больших ИС требуются большой коллектив разработчиков.
2. Применима для ИС, которые могут декомпозироваться на отдельные модули и в которых производительность не является критической величиной.
3. Не используется в случае применения новых технологий.

## 2.1.4. CASE-технология RUP

Одна из самых популярных CASE-технологий - *Rational Unified Process* (RUP), разработанная компанией Rational Software, которая в настоящее время входит в состав IBM. Она ориентирована на использование языка объектно-ориентированного моделирования UML, являющегося стандартом в этой области.

Среди всех фирм-производителей CASE-средств компания Rational Software одной из первых осознала стратегическую перспективность развития объектно-ориентированных технологий анализа и проектирования программных систем. Эта компания выступила инициатором унификации языка визуального моделирования, что привело к появлению первых версий UML. Она же первой разработала инструментальное объектно-ориентированное CASE-средство, в котором был реализован язык UML, как базовая нотация визуального моделирования.

На рис. 2.2 [38] показан общий вид процесса RUP в двух измерениях:

- *горизонтальное измерение* отражает динамические аспекты процессов и оперирует такими понятиями, как циклы, фазы, итерации и контрольные точки;
- *вертикальное измерение* отражает статические аспекты процессов и оперирует такими понятиями, как действия, результаты деятельности, исполнители и рабочие процессы.

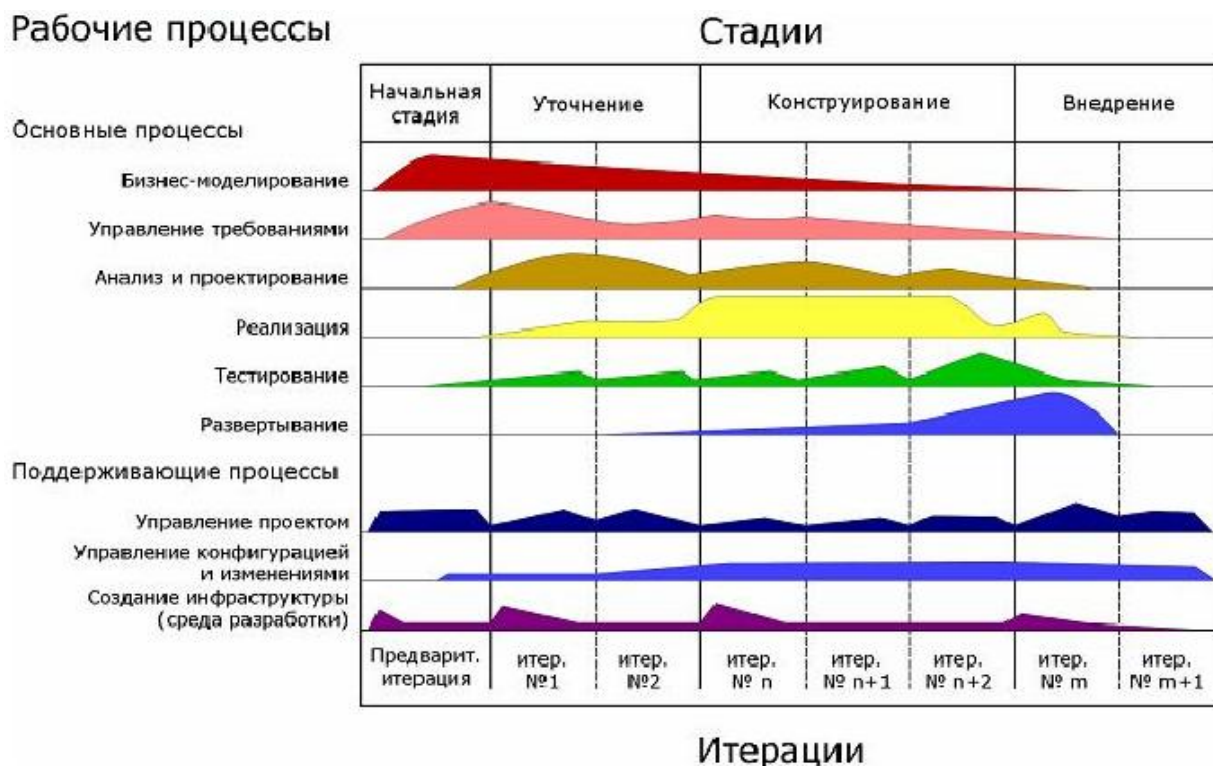


Рис. 2.2. Процесс разработки в методологии RUP

Согласно технологии RUP жизненный цикл ПО разбивается на отдельные классы, в каждом из которых создается новое поколение продукта. Каждый цикл, в свою очередь, разбивается на четыре последовательных стадии:

- начальная стадия (inception);
- стадия уточнения (elaboration);
- стадия конструирования (construction);
- стадия ввода в действие (transition).

Каждая стадия завершается в четко определенной контрольной точке (milestone). В этот момент времени должны достигаться важные результаты и приниматься критически важные решения о дальнейшей разработке.

*Статический аспект RUP* характеризуют следующие элементы:

- **исполнители**, определяющие поведение и ответственность личности или группы личностей, составляющих проектную компанию;
- **действия** конкретного исполнителя, как единицы выполняемой им работы;
- **результаты деятельности**, на получение или модификацию которых направлены действия исполнителей;
- **рабочие процессы**, представляющие собой последовательности, действий, приводящие к получению значимого результата.

В методологии RUP реализуются следующие подходы:

- *итерационный* и *инкрементный* (наращиваемый);
- построение системы на базе архитектуры ИС;
- управление проектом на основе функциональных требований к ИС.

## 2.2. Функционально-ориентированное проектирование информационных систем

### 2.1.1. Методология структурно-функционального моделирования

Технология описания процессов реализуется через процесс подробного описания всех процессов организации в виде графических моделей с обозначением всех взаимных связей. Для максимальной эффективности этой технологии необходимо иметь утвержденную методологию, квалифицированный персонал и средство описания процессов [31]. Технология описания процессов включает в себя определенную методологию, которая включает в себя [42]:

1. **Описание организационной структуры компании.** Строится иерархическая модель организации по принципу подчиненности и взаимосвязанности.
2. **Описание ИС организации.** Разрабатывается модель ИС, обозначающая все модули и применяемое программное обеспечение.
3. **Описание функций подразделений организации.** Разрабатывается функциональная модель организации с подробным описанием всех функций структурных подразделений.
4. **Описание выбранных процессов организации.** Разрабатываются модели всех процессов организации.

5. **Описание продуктов и услуг организации.** Строится модель продуктов и услуг, которые производятся или реализуются организацией как на внешнем, так внутреннем рынке.

6. **Определение и построение дерева процессов организации.** Осуществляется классификация всех процессов на основные, обеспечивающие и управления с последующим построением модели процессов организации.

7. **Описание информационных потоков всех процессов.** Происходит процесс идентификации информационных потоков выявленных процессов с последующим построением модели информационных потоков.

Методология SADT (Structured Analysis and Design Technique – методология структурного анализа и проектирования), разработанная Дугласом Т. Россом в 1969-1973 годах, базируется на структурном анализе систем и графическом представлении организации в виде системы функций, которые имеют три класса структурных моделей: *функциональную, информационную и динамическую*.

Процесс моделирования по методологии SADT состоит из следующих этапов:

1. Сбор информации и анализ информации о предметной области.
2. Документирование полученной информации.
3. Моделирование (стандарт IDEF0).
4. Корректурная модель в процессе итеративного рецензирования.

Методология моделирования SADT (IDEF0) предназначена для анализа всей системы как множества взаимодействующих взаимосвязанных функций.

Ориентация исключительно на анализ функций позволяет рассматривать функции независимо от объектов, которые их выполняют. *Структурно-функциональный подход* позволяет четко разделить проблемы анализа и проектирования от проблем реализации. Поэтому такой подход является чрезвычайно удобным на этапе анализа и проектирования, поскольку аналитики имеют дело с процессами деятельности, по сути, являющимися функциями или группами функций.

Методология в настоящее время более известна как стандарт IDEF0, использует формализованный процесс моделирования ИС и имеет следующие стадии: *анализ, проектирование, реализация, объединение, тестирование, установка, функционирование*. Проектирование ИС по стандарту IDEF0 сводится к декомпозиции основных функций организации на отдельные процессы, работы или действия. В результате разрабатывается иерархическая модель анализируемой организации, при этом декомпозицию можно проводить многократно, до четкого и детального описания всех процессов. Диаграммы IDEF0 верхнего уровня принято называть *родительскими*, а нижнего уровня – *дочерними*.

Исторически IDEF0 как стандарт был разработан в 1981 году в рамках обширной программы автоматизации промышленных предприятий по инициативе BBC США, которая носила обозначение ICAM (Integrated Computer Aided

Manufacturing – Интеграция компьютерных и промышленных технологий). В 2000 г. IDEF0 был принят в качестве стандарта и в РФ (РД IDEF0-2000).

В значительной мере успех методологии SADT объясняется графической нотацией стандарта IDEF0, хотя не менее ценным является сам процесс моделирования, который подсказывает вполне определенный путь выполнения согласованной и достоверной структурной декомпозиции. SADT уникальна в своей способности обеспечить как графический язык, так и процесс создания непротиворечивой и полезной системы описаний. Именно то, что SADT является не просто графической нотацией, но, прежде всего, системным подходом к анализу и синтезу сложных систем, делает ее полноценной методологией, широко распространенной не только в сфере проектирования ИС, но и в областях, связанных с оптимизацией организационно-экономических и производственно-технических систем.

### 2.1.2. Стандарт моделирования IDEF0

Целью является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы. В основе методологии лежат четыре основных понятия: функциональный блок, интерфейсная дуга (стрелка), декомпозиция, глоссарий.

**Функциональный блок (Activity Box)** представляет собой некоторую конкретную функцию в рамках рассматриваемой системы. По требованиям стандарта название каждого функционального блока должно быть сформулировано в глагольном наклонении (например, "производить услуги"). На диаграмме функциональный блок изображается прямоугольником (рис. 2.3). Каждая из четырех сторон функционального блока имеет свое определенное значение (роль):

- верхняя сторона имеет значение "Управление"(Control);
- левая сторона имеет значение "Вход"(Input);
- правая сторона имеет значение "Выход"(Output);
- нижняя сторона имеет значение "Ресурсы"(Mechanism).

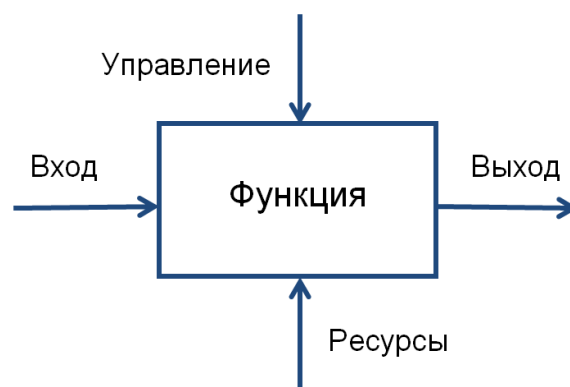


Рис. 2.3. Функциональный блок

**Интерфейсная дуга (стрелка) (Arrow)** отображает элемент системы, который обрабатывается функциональным блоком или оказывает иное влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги часто называют *потоками* или *стрелками*. С помощью интерфейсных дуг отображают различные объекты, в той или иной степени определяющие процессы, происходящие в системе. Такими объектами могут быть элементы реального мира (детали, вагоны, сотрудники и т.д.) или потоки данных и информации (документы, данные, инструкции и т.д.). В зависимости от того, к какой из сторон функционального блока подходит данная интерфейсная дуга, она носит название «входящей», «исходящей» или «управляющей». Любой функциональный блок по требованиям стандарта должен иметь, по крайней мере, одну управляющую интерфейсную дугу и одну исходящую, т.к. каждый процесс должен происходить по каким-то правилам (отображаемым управляющей дугой) и выдавать некоторый результат (выходящая дуга), иначе его рассмотрение не имеет никакого смысла. Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий - DFD и WFD.

1. Интерфейсная стрелка «**вход**» (**Input**) - это материалы, предметы или информация, которые трансформируются в процессе выполнения функции с целью получения результата. Стрелки входа соединяются с левой стороной блока. Некоторые блоки могут не иметь стрелок входа, поскольку не каждая функция преобразует или изменяет что-либо.

2. Интерфейсная стрелка «**управление**» (**Control**) определяет как, когда и в каком случае выполняется функция, и какой результат от нее ожидается. Каждая функция (IDEF0-блока) должна иметь как минимум один вход управления. Управление часто представляется в виде правил, норм, процедур, стандартов. Они оказывают влияние на выполнение функции, не изменяясь при этом сами. Управление – это особый тип входных данных функции. Часто даже возникает вопрос, какого типа должна быть стрелка: вход или управление.

3. Интерфейсная стрелка «**ресурс**» или «**механизм**» (**Mechanism**) обозначает те ресурсы, при помощи которых выполняется функция. В качестве механизма выступают люди, машины, оборудование, которые обеспечивают все необходимое для реализации функции. IDEF0-блок может не содержать стрелок механизма. Это объясняется тем, что знание механизма, осуществляющего функцию, зачастую не является целью моделирования системы.

4. Интерфейсная стрелка «**выход**» (**Output**) – это материалы, предметы, информация, производимые функцией, это результат выполнения функции. Каждый блок обязательно имеет хотя бы одну стрелку выхода. При моделировании непродуцированных процессов выходом функции часто являются данные, которые были обработаны по алгоритму, определяемому функцией.



5. Интерфейсные стрелки **ссылки (Call Arrow)** используются для указания на другие модели или диаграммы внутри модели, которые помогают лучше понять модель. Интерфейсная стрелка ссылки может ссылаться на другую диаграмму внутри модели или на специфическое дочернее действие другой модели.

**Декомпозиция (Decomposition)** является основным понятием стандарта IDEF0. Принцип декомпозиции применяется при разбиении сложного процесса на составляющие его функции. Уровень детализации процесса определяется непосредственно разработчиком модели. Декомпозиция позволяет постепенно и структурированно представлять модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Последним из понятий IDEF0 является **глоссарий (Glossary)**. Для каждого из элементов IDEF0 — диаграмм, функциональных блоков, интерфейсных дуг — существующий стандарт подразумевает создание и поддержание набора соответствующих определений, ключевых слов, повествовательных изложений и т.д., которые характеризуют объект, отображенный данным элементом. Этот набор называется глоссарием и является описанием сущности данного элемента. Глоссарий гармонично дополняет наглядный графический язык, снабжая диаграммы необходимой дополнительной информацией.

Модель IDEF0 всегда начинается с представления системы как единого целого — одного функционального блока с интерфейсными дугами, простирающимися за пределы рассматриваемой области.

Такая диаграмма с одним функциональным блоком называется **контекстной диаграммой**. В пояснительном тексте к контекстной диаграмме должна быть указана **цель (Purpose)** построения диаграммы в виде краткого описания и зафиксирована **точка зрения (Viewpoint)**.

**Цель (Purpose).** Определение и формализация цели разработки IDEF0-модели является крайне важным моментом, т.к. цель определяет соответствующие области в исследуемой системе, на которых необходимо фокусироваться в первую очередь. Модели создаются для получения ответа на ряд вопросов. Данные вопросы должны подготавливаться заранее и будут служить основой для создания цели модели. Примерные вопросы могут звучать следующим образом: каковы основные задачи сотрудника; кто отвечает за произведенную продукцию; кто управляет начальной стадией производства; какой требуется инструмент для каждого этапа и т.п.

**Точка зрения (Viewpoint).** Особенно важно включать в процесс разработки модели представителей различных мнений, однако сама модель должна базироваться на единой точке зрения. Чаще всего разнообразные точки зрения кратко фиксируют на диаграмме FEO (For Exposition Only — только для комментариев). Эти диаграммы используются только в качестве материалов для презентаций.

Точка зрения должна формулироваться исходя из цели построения диаграммы. При построении модели важно придерживаться одной точки зрения, которая должна содержать наименование должности, структурного подразделения или описание должностных обязанностей работника. Модели могут содержать разнообразные точки зрения с целью детальной фиксации всех действий (функций).

Точка зрения определяет основное направление развития модели и уровень необходимой детализации. Четкое фиксирование точки зрения позволяет разгрузить модель, отказавшись от детализации и исследования отдельных элементов, не являющихся необходимыми, исходя из выбранной точки зрения на систему.

Диаграммы IDEF0 верхнего уровня принято называть родительскими, а нижнего уровня – дочерними. Диаграммы IDEF0 второго уровня называются дочерней (Child diagram) по отношению к первому (каждый из функциональных блоков, принадлежащих дочерней диаграмме соответственно называется дочерним блоком – Child Box). В свою очередь, функциональный блок – предок называется родительским блоком по отношению к дочерней диаграмме (Parent Box), а диаграмма, к которой он принадлежит – родительской диаграммой (Parent Diagram).

При декомпозиции функционального блока все стрелки, входящие в данный блок, или исходящие из него фиксируются на дочерней диаграмме. Этим достигается структурная целостность IDEF0-модели. Каждый блок имеет свой уникальный порядковый номер на диаграмме (цифра в правом нижнем углу), а обозначение под правым углом указывает на номер дочерней для этого блока диаграммы. Отсутствие этого обозначения говорит о том, что декомпозиции для данного блока не существует.

**Туннели (Tunnels).** Связывание интерфейсных стрелок используется в моделях для определения уровня детализации. Когда интерфейсная стрелка не возникает на базовой диаграмме, но не связана с другими стрелками, туннель используется для указания того, что интерфейсная стрелка входит или выходит из системы. Туннель используется, чтобы не загромождать базовую диаграмму. В других случаях туннель может быть использован в интерфейсной стрелке, ведущей к базовому действию. Это указывает, что взаимоотношения интерфейсной стрелки с дочерними действиями не определены.

Далее каждый подпроцесс тоже можно декомпозировать и подробно описывать все связи до необходимого предела. Основным достоинством этой методологии являются простота и наглядность. В качестве недостатка – невозможность описать реакцию описываемого процесса на изменяющиеся внешние факторы. Для этих целей служат другие методологии.

### **Варианты взаимодействия функциональных блоков**

Дуги позволяют отражать взаимосвязи между блоками. По типу этой связи вы-

деляют пять видов взаимодействия: «выход - вход», «выход - управление», «выход – механизм», «обратная связь выход – управление», «выход – вход обратная связь». На рисунках 2.4-2.8 представлены примеры различного взаимодействия функциональных блоков.



Рис. 2.4. Взаимодействие типа «выход-вход»



Рис. 2.5. Взаимодействие типа «выход-управление»

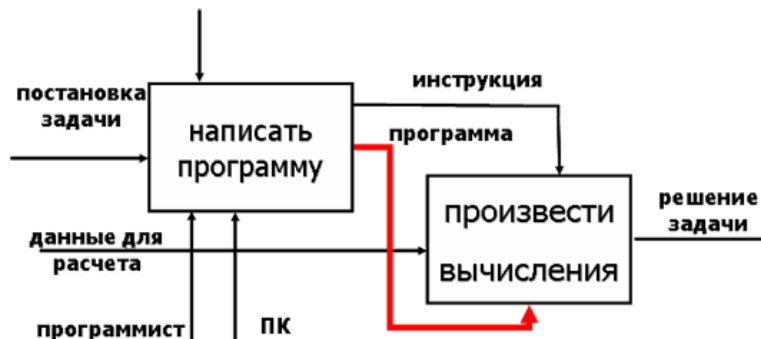


Рис. 2.6. Взаимодействие типа «выход-механизм»

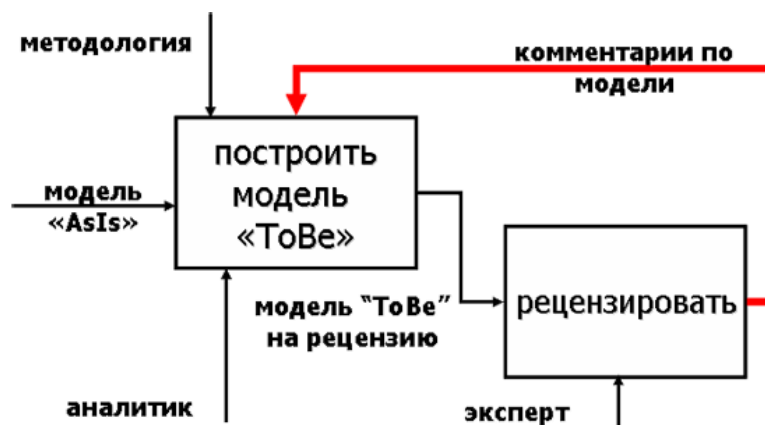


Рис. 2.7. Взаимодействие типа «обратная связь выход-управление»

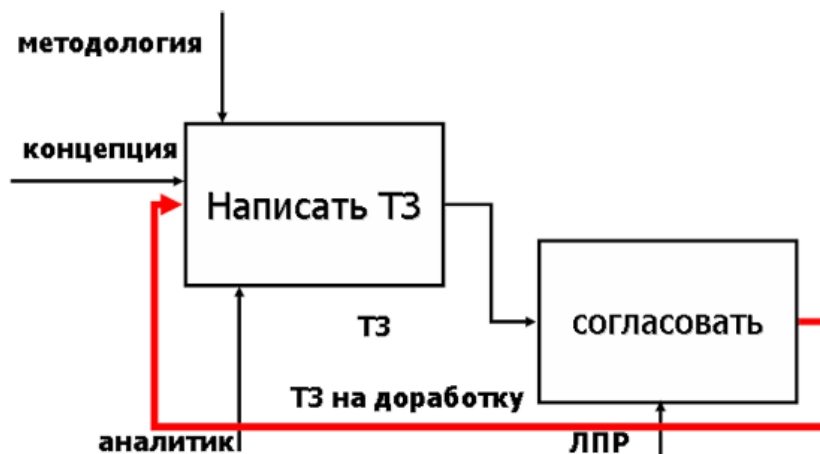


Рис. 2.8. Взаимодействие типа «обратная связь выход-вход»

## Принципы IDEF0 [36]

### 1. Принцип функциональной декомпозиции

Принцип декомпозиции или, как его еще называют принцип «разделяй и властвуй» применяется при разбиении сложного процесса на составляющие его функции. Декомпозиция позволяет представить модель системы в виде иерархической структуры отдельных диаграмм, что делает ее менее перегруженной и легко усваиваемой.

Суть принципа функциональной декомпозиции очень прост (рис. 2.9):

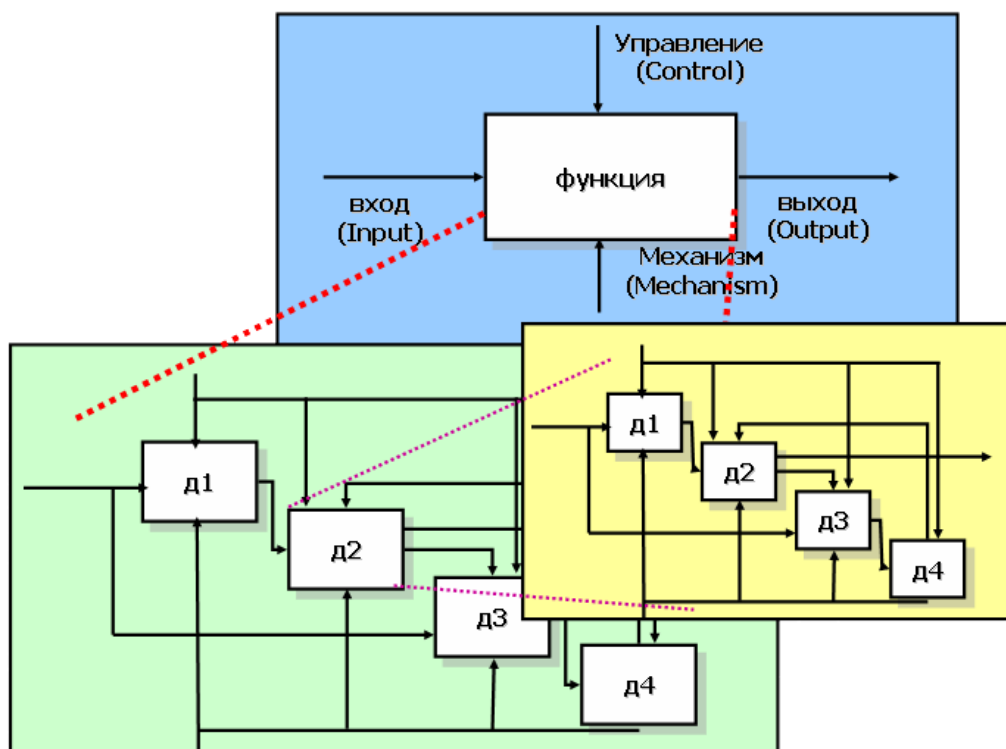


Рис. 2.9. Функциональная декомпозиция [42]

1. Функциональный блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных между собой дугами.

2. Эти блоки представляют основные подфункции (подмодули) единого исходного модуля.

3. Данная декомпозиция выявляет полный набор подмодулей, каждый из которых представлен как блок, границы которого определены дугами.

4. Каждый из этих подмодулей может быть декомпозирован подобным же образом для более детального представления.

## 2. Принцип ограничения сложности

При работе с IDEF0 диаграммами существенным является условие их разборчивости и удобочитаемости, поэтому количество блоков на диаграмме должно быть не менее двух и не более шести. Такой диапазон вызван учетом особенностей психологии человеческого восприятия. Практика показывает, что соблюдение этого принципа приводит к тому, что функциональные процессы, представленные в виде IDEF0 модели, хорошо структурированы, понятны и легко поддаются анализу.

## 3. Принцип контекста (целеполагания)

Как было указано выше любая модель должна отвечать на вопросы о системе («М есть модель системы S, если М может быть использована для получения ответов на вопросы относительно S с точностью А»). Это означает, что не может быть модели вообще. Любая модель – это лишь инструмент и чтобы правильно ее создать, надо иметь однозначное представление о цели моделирования (Purpose), точке зрения (Viewpoint) и границах моделирования (Scope). Принцип целеполагания как раз и означает, что любая модель в IDEF0 – должна быть определена прежде всего по перечисленным трем позициям.

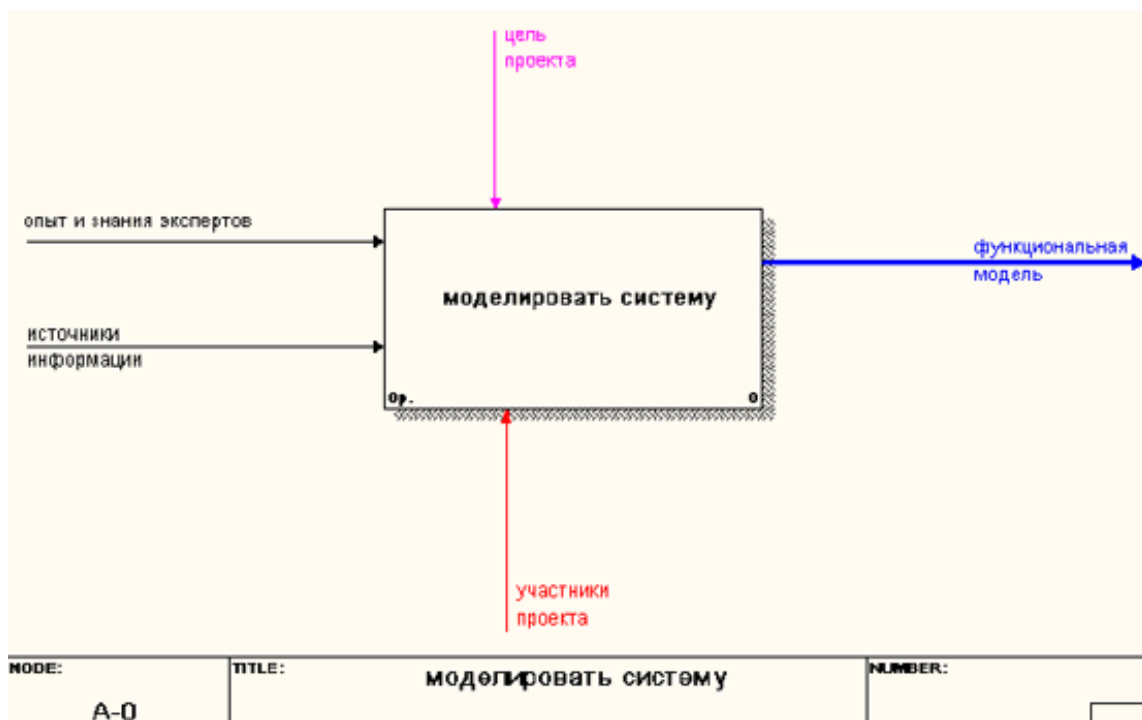


Рис. 2.10. Контекстная диаграмма

Если процесс моделирования, описанный в РД IDEF0-2000 представить в нотации IDEF0, то получим следующие диаграммы: контекстная – рис. 2.10; диаграмма декомпозиции – рис. 2.11.

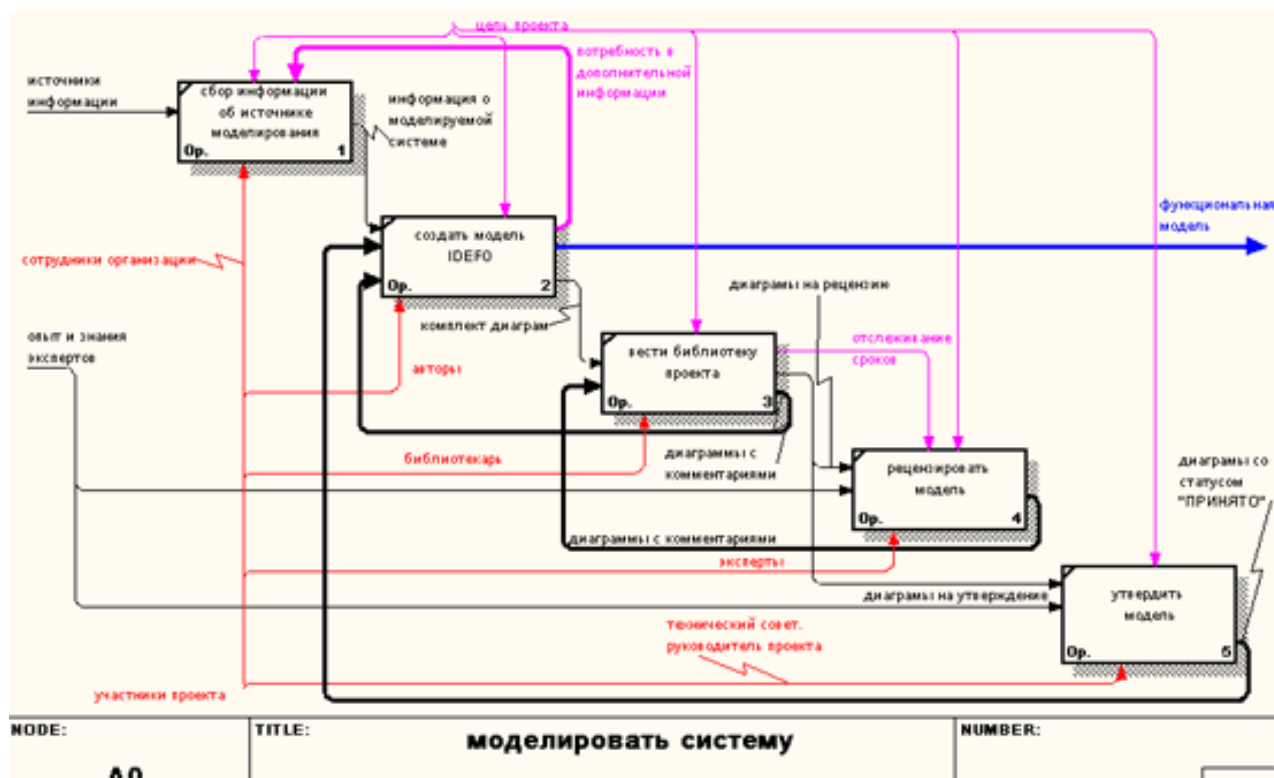


Рис. 2.11. Диаграмма декомпозиции

### 2.1.3. Моделирование потоков данных в нотации DFD

В DFD (Data Flow Diagram) модель системы определяется как иерархия диаграмм потоков данных, описывающих процессы преобразования информации от момента ее ввода в систему до выдачи конечному пользователю. Диаграммы верхних уровней иерархии - контекстные диаграммы, задают границы модели, определяя её окружение (внешние входы и выходы) и основные рассматриваемые процессы. Контекстные диаграммы детализируются при помощи диаграмм следующих уровней.

При проведении проекта создания ИС нотация DFD может использоваться в качестве основной нотации функционального моделирования, однако, часто она применяется как дополнительная по отношению к IDEF0[36]. Следует отметить, что в отличие от SADT (IDEF0), DFD методологией не является. Другими словами, DFD – это всего лишь набор общепринятых обозначений без жестких ограничений к способам моделирования и применения полученных моделей.

Диаграммы потоков данных используются для описания движения документов и обработки информации. В отличие от IDEF0, где система рассматривается как взаимосвязанные функциональные блоки, а дуги представляют собой жест-

кие взаимосвязи, стрелки в DFD показывают лишь то, как объекты (включая данные) движутся от одной работы к другой. DFD отражает функциональные зависимости значений, вычисляемых в системе, включая входные значения, выходные значения и внутренние хранилища данных. DFD - это граф, на котором показано движение значений данных от их источников через преобразующие их процессы к их потребителям в других объектах. DFD содержит процессы, которые преобразуют данные; потоки данных, которые переносят данные; активные объекты, которые производят и потребляют данные; хранилища данных, которые пассивно хранят данные.

Если говорить о выразительной силе нотации и сравнивать DFD с IDEF0, можно сказать, что отсутствие таких понятий как управление и механизм резко сокращают потенциал DFD при анализе модели, выявлении «узких мест», поиске путей усовершенствования и т.д. Все это привело к тому, что DFD достаточно редко применяется как базовая нотация в проектах инжиниринга процессов, построения системы менеджмента качества и т.д.

Так как DFD не является стандартом, на настоящее время нет единой нотации со своими однозначно определенными примитивами. Для представления моделей применяется ряд различных нотаций DFD. Наибольшее распространение среди них получили нотации Гейна-Сарсона и Йодана/де Марко.

Несмотря на существование нескольких разных нотаций DFD все они отличаются только тем набором графических примитивов, которые используются для построения функциональных моделей.

### **Структура DFD модели**

Первым шагом при построении иерархии DFD, также, как и в SADT, является построение контекстных диаграмм. Обычно при проектировании относительно простых ИС строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы.

Если же для сложной системы ограничиться единственной контекстной диаграммой, то она будет содержать слишком большое количество источников и приемников информации, которые трудно расположить на листе бумаги нормального формата, и кроме того, единственный главный процесс не раскрывает структуры распределенной системы. Признаками сложности (в смысле контекста) могут быть:

- наличие большого количества внешних сущностей (десять и более);
- распределенная природа системы;
- многофункциональность системы с уже сложившейся или выявленной группировкой функций в отдельные подсистемы.

Для сложных ИС строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем проектируемой ИС как между собой, так и с внешними входными и выходными потоками данных и внешними объектами (источниками и приемниками информации), с которыми взаимодействует ИС.

Разработка контекстных диаграмм решает проблему строгого определения функциональной структуры ИС на самой ранней стадии ее проектирования, что особенно важно для сложных многофункциональных систем, в разработке которых участвуют разные организации и коллективы разработчиков.

После построения контекстных диаграмм полученную модель следует проверить на полноту исходных данных об объектах системы и изолированность объектов (отсутствие информационных связей с другими объектами).

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи DFD. Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи DFD или миниспецификации. При детализации должно выполняться правило балансировки. Суть этого правила сводится к тому, что при детализации подсистемы или процесса детализирующая диаграмма в качестве внешних источников/приемников данных может иметь только те компоненты (подсистемы, процессы, внешние сущности, накопители данных), с которыми имеет информационную связь детализируемая подсистема или процесс на родительской диаграмме;

Миниспецификация (описание логики процесса) должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

Миниспецификация является конечной вершиной иерархии DFD. Решение о завершении детализации процесса и использовании миниспецификации принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока);
- возможности описания преобразования данных процессом в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи миниспецификации небольшого объема (не более 20-30 строк).



При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных.

Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать в структуре. Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанном диапазоне. Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных может указываться единица измерения (кг, см и т.п.), диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

После построения законченной модели системы ее необходимо верифицировать (проверить на полноту и согласованность). В полной модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Выявленные недетализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

### **Основные элементы DFD и их назначение**

Синтаксис DFD включает четыре основных элемента:

- поток данных;
- процесс;
- хранилище;
- внешняя сущность.

Рассмотрим эти элементы подробнее.

#### ***Поток данных***

Поток данных соединяет выход объекта (или процесса) с входом другого объекта (или процесса). Он представляет промежуточные данные вычислений. Поток данных изображается в виде стрелки между производителем и потребителем данных, помеченной именами соответствующих данных. Упрощенно можно считать, что потоки данных являются механизмами, используемыми для моделирования передачи информации (или физических компонент) из одной части системы в другую.

Потоки на диаграммах изображаются стрелками (обычно именованными), ориентация которых указывает направление движения информации.

В отличие от дуг в IDEF0 потоки данных в DFD могут быть не только односторонними, но и двусторонними.

### ***Процесс***

Процесс преобразует значения данных. Процессы представляют собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. В реальной жизни процесс может выполняться некоторым подразделением организации, выполняющим обработку входных документов и выпуск отчетов, отдельным сотрудником, программой, установленной на компьютере, специальным логическим устройством и тому подобное.

Назначение процесса состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Это имя должно содержать глагол в неопределенной форме с последующим дополнением (например, «выдать пропуск»). Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Как уже говорилось ранее из-за отсутствия единого стандарта, объекты DFD могут иметь разное обозначение.

Особо следует подчеркнуть, что в отличие от SADT, в DFD все стороны блока равнозначны (это очевидно, если посмотреть на обозначение процесса в нотации Йодана/де Марко). Другими словами, в отличие от IDEF0 диаграмм, в DFD диаграммах не используются стрелки управления для обозначения правил выполнения действия и стрелки механизмов для обозначения требуемых ресурсов.

### ***Хранилища (накопители данных)***

1. **Накопители данных** предназначены для изображения неких абстрактных устройств для хранения информации, которую можно туда в любой момент времени поместить или извлечь, безотносительно к их конкретной физической реализации. Накопители данных являются неким прообразом базы данных информационной системы организации.

2. **Хранилище (накопитель данных)** позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет "срезы" потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое и быть существительным. В случае, когда поток данных входит или выходит в/из хранилища, и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое нет необходимости отражать на диаграмме.

Если проводить сравнение DFD с IDEF0, то выясняется, что прямого аналога

хранилища в IDEF0 не предусмотрено. Когда речь идет о разработке программной системы и проектировании базы данных наличие хранилищ, которые при грамотном описании представляют собой по сути готовую ERD, является серьезным достоинством DFD.

### **Внешняя сущность (Терминатор)**

1. **Внешняя сущность (терминатор)** представляет сущность вне контекста системы, являющуюся источником или приемником системных данных. Ее имя должно содержать существительное. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке.

2. Под **внешней сущностью (External Entity)** понимается материальный объект, являющийся источником или приемником информации.

В качестве внешней сущности на DFD диаграмме могут выступать заказчики, поставщики, склад, банк и другие.

Пример использования всех рассмотренных элементов на DFD модели можно проиллюстрировать следующей диаграммой (рис. 2.12).

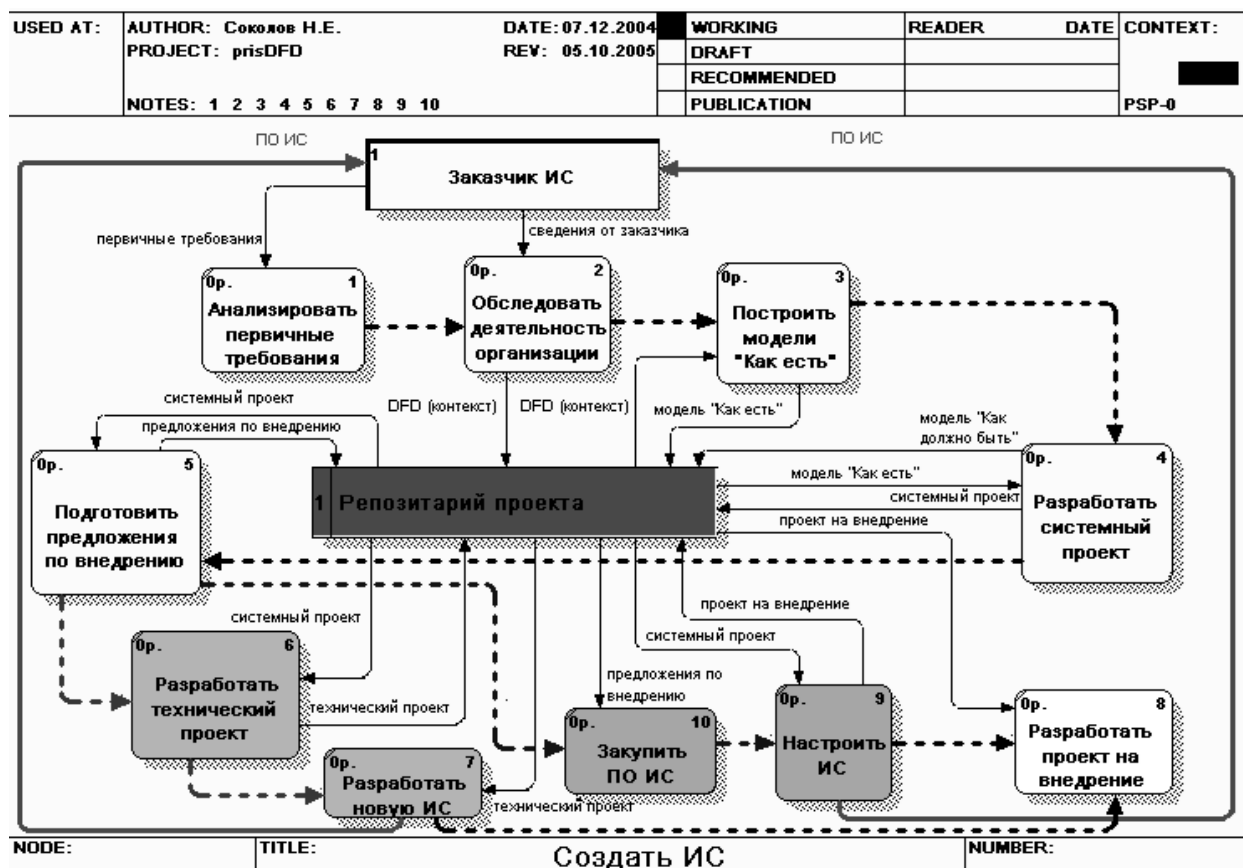


Рис. 2.12. Пример DFD- диаграммы

### **Выводы по нотации DFD.**

В некоторых случаях DFD может рассматриваться в качестве основной нотации функционального моделирования при проектировании ИС. Учитывая то, что IDEF0 также является нотацией, обеспечивающей описание организационно-экономических и производственно-технологических систем, возникает проблема

выбора нотации при проведении конкретного проекта автоматизации. Как следует из проведенного краткого обзора сравниваемых нотаций, DFD имеет преимущество над IDEF0 в части представления на модели структур данных. Фактически, эта нотация позволяет уже на стадии функционального моделирования проектировать базу данных.

Серьезными недостатками DFD является то, что:

- 1) выразительная сила нотации DFD оказывается недостаточной при анализе модели, выявлении «узких мест», поиске путей усовершенствования и т.д.;
- 2) DFD методологией не является, что приводит к возможности неоднозначной трактовки результатов моделирования.

Все это позволяет говорить о том, что применение DFD в качестве базовой нотации функционального моделирования оправдано в случае, когда речь идет о разработке программной системы и предполагается автоматизация существующих процессов без их оптимизации, то есть, когда речь идет о «лоскутной» автоматизации.

В случае комплексной автоматизации, когда основное значение приобретает не программирование, а поиск решений оптимизации деятельности, нотация DFD не выдерживает конкуренции с IDEF0 и может рассматриваться лишь как дополнительная.

## **2.3. Объектно-ориентированное проектирование информационных систем на основе языка UML**

### **2.3.1. Основы унифицированного языка моделирования UML**

Существует большое количество инструментальных средств, используемых для реализации проекта ИС от этапа анализа до создания программного кода. Отдельно выделяют так называемые CASE-средства верхнего уровня (upper CASE tools) и CASE-средства нижнего уровня (lower CASE tools). Среди основных проблем использования CASE-средств верхнего уровня выделяют проблемы их адаптации под конкретные проекты, так как они жестко регламентируют процесс разработки и не дают возможности организовать работу на уровне отдельных элементов проекта. Альтернативой им может стать использование CASE-средства нижнего уровня, но их использование влечет другие проблемы – трудности в организации взаимодействия между командами, работающими над различными элементами проекта. Средством, позволяющим объединить эти подходы, явился унифицированный **язык объектно-ориентированного моделирования** (Unified Modeling Language – UML). К преимуществам языка UML можно отнести разнообразные инструментальные средства, которые как поддерживают жизненный цикл ИС, так и позволяют настроить и отразить специфику деятельности разработчиков различных элементов проекта.

Рассмотрим программный инструмент моделирования StarUML. Данная программная платформа имеет свободную лицензию и доступна для установки с официального сайта StarUML [54].

StarUML поддерживает одиннадцать различных типов диаграмм, принятых в нотации UML 2.0, а также подход MDA (модельно-настраиваемая архитектура), предлагает настройку параметров пользователя для адаптации среды разработки, поддерживает расширения, предоставляет различного рода модули, расширяющие возможности StarUML.

Для поддержки моделирования различных этапов жизненного цикла ИС язык UML предлагает целую совокупность диаграмм, технология построения которых в StarUML описана в работе [37].

При разработке **концептуальной модели** применяют диаграммы вариантов использования и диаграммы деятельности, модели объектов деятельности, диаграммы последовательностей.

На этапе работы над **логической моделью** ИС описать требования к системе позволяют диаграммы вариантов использования, а при предварительном проектировании используют диаграммы классов, диаграммы состояний, диаграммы последовательностей.

Детальное проектирование при разработке **физической модели** выполняют с применением диаграмм классов, диаграмм развертывания, диаграмм компонентов. Далее будут подробнее рассмотрены перечисленные диаграммы с указанием их назначения в процессе проектирования ИС.

### 2.3.2. Проектирование логической модели ИС и моделей баз данных

Одной из диаграмм, применяющихся на этапе проектирования логической модели ИС, является **диаграмма вариантов использования** (диаграмма прецедентов, use case diagram), предназначенная для построения на концептуальном уровне модели того, как функционирует система в окружающей среде.

Основными элементами для построения модели прецедентов на диаграмме являются:

- **актёр** (actor) – элемент, обозначающий роли пользователя, который взаимодействует с определенной сущностью;
- **прецедент** – элемент, отражающий действия, выполняемые системой (в т.ч., с указанием возможных вариантов), которые приводят к результатам, наблюдаемым актерами.

Между прецедентами в модели могут быть установлены три вида связей:

- **обобщение** (generalization) – указывает общность ролей;
- **включение** (include) – указывает взаимосвязь нескольких вариантов использования, базовый из которых всегда использует функциональное поведение

связанных с ним прецедентов;

– **расширение** (extend) – указывает взаимосвязь базового варианта использования и вариантов использования, являющихся его специальными случаями.

Пример диаграммы вариантов использования представлен на рис. 2.13.

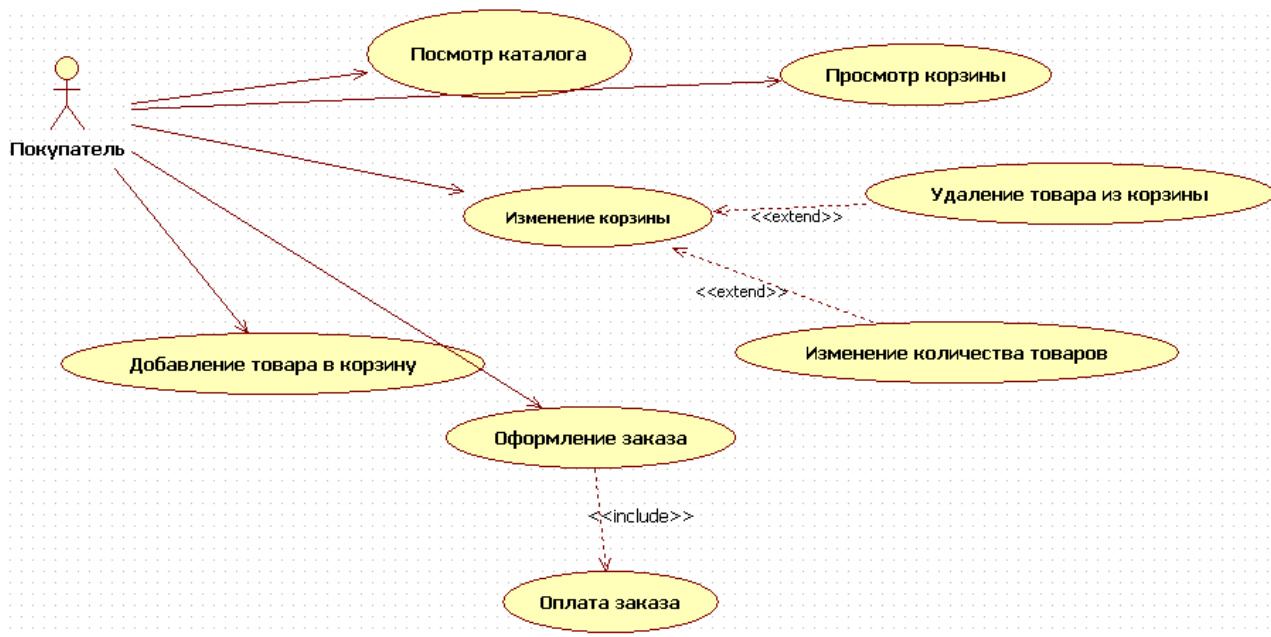


Рис. 2.13. Диаграмма вариантов использования

Более детально описать процессы деятельности позволяют диаграммы деятельности и диаграммы последовательностей.

**Диаграмма деятельности** (activity diagram) – диаграмма, используемая при моделировании бизнес-процессов, на которой представлено разложение на составные части некоторой деятельности, а именно: скоординированного выполнения отдельных действий и вложенных видов деятельности, которые соединяются между собой потоками от выходов одного узла к входам другого, с указанием их исполнителей. Пример диаграммы деятельности представлен на рис. 2.14.

Разработанные на этапе построения моделей прецедентов диаграммы деятельности могут корректироваться вследствие выявления новых подробностей в описании процессов объекта автоматизации на этапах анализа и проектирования.

**Диаграмма последовательности** (sequence diagram) – диаграмма, отражающая упорядоченные по времени проявления взаимодействия объектов.

На диаграмме данного типа слева направо помещаются основные элементы: объекты; вертикальные линии (lifeline), моделирующие течение времени при выполнении действий объектом; стрелки, определяющие действия, выполняемые объектом.

Пример диаграммы последовательности представлен на рис. 2.15.

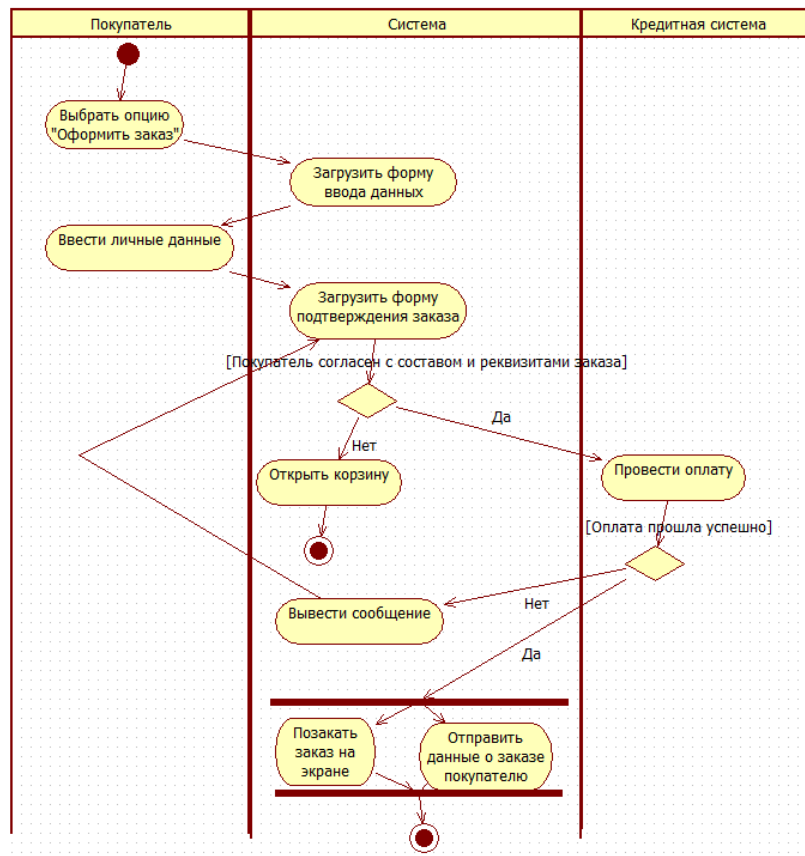


Рис. 2.14. Диаграмма деятельности

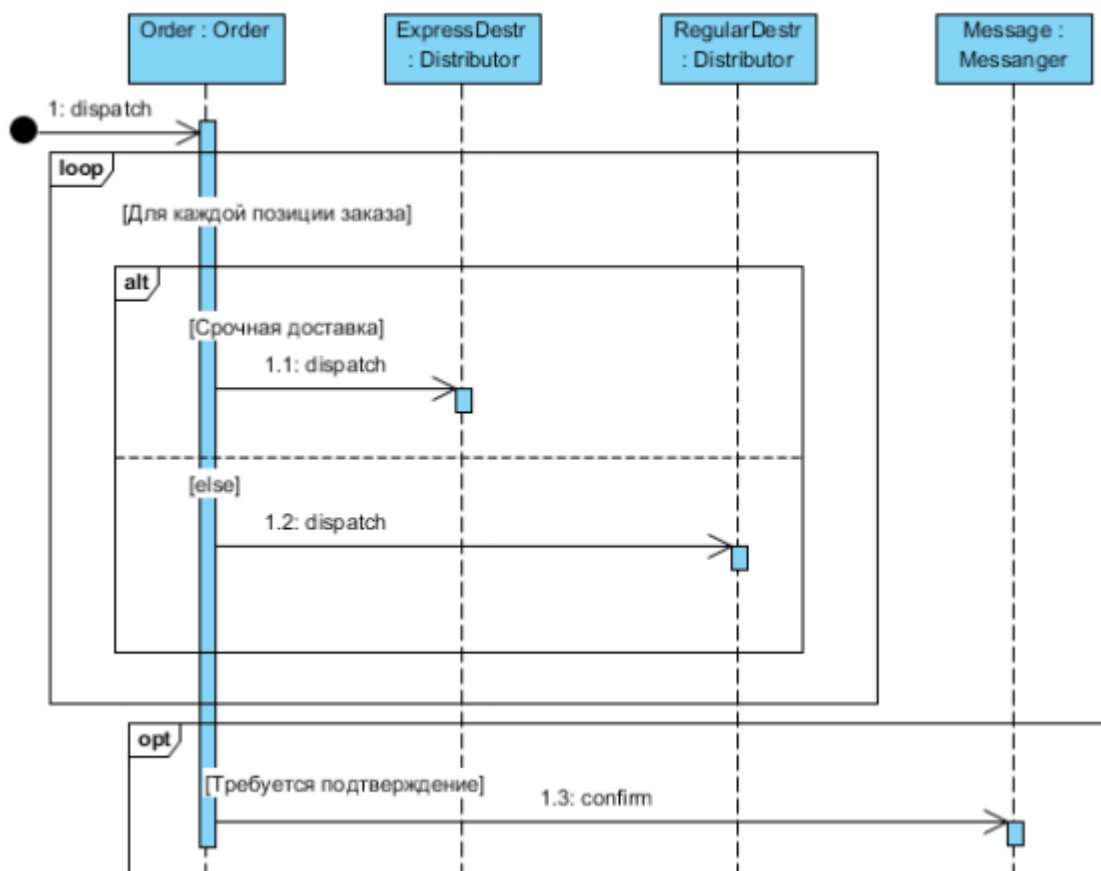


Рис. 2.15. Диаграмма последовательности

В результате построения диаграмм на этом этапе разработаны подробные описания действий специалистов по внедрению ИС, которые необходимы для обеспечения ее функциональности.

На этапе формирования требований разрабатывается модель системных прецедентов, на которой для внутренних и внешних исполнителей указываются их конкретные обязанности с использованием ИС. Модели системных прецедентов разрабатываются на основе моделей, построенных на предыдущем этапе, однако при этом необходимо детально описать прецеденты с определениями используемых данных и указанием последовательности их выполнения, то есть подробно описать реализацию функций проектируемой системы.

На этапе анализа требований и предварительного проектирования системы на основе построенных моделей системных прецедентов строятся диаграммы классов системы. Диаграмма классов, являясь логическим представлением модели, представляет детальную информацию о структуре модели системы с использованием терминологии классов объектно-ориентированного программирования, а именно: о внутреннем устройстве системы (об архитектуре системы). На диаграмме классов могут быть указаны внутренняя структура и типы отношений между отдельными объектами и подсистемами, что приводит к развитию концептуальной модели системы. Класс в языке UML обозначает некоторое множество объектов, обладающих одинаковой структурой и взаимосвязями с объектами других классов. В диаграммах классов системы указываются объекты из модели системных прецедентов с их описанием и указанием взаимосвязей между классами. Синтаксис диаграмм классов является эффективным средством структурирования требований к элементам проектируемой системы, к их данным, интерфейсам, функциональности.

Пример диаграммы классов представлен на рис. 2.16.

На этом этапе проектирования подробно описаны состав и функции системы в соответствии с разработанными моделями, что дает уверенность в соответствии проектируемой системы требованиям заказчика.

На следующем этапе элементы разработанных моделей классов отображаются в элементы моделей базы данных и приложений, а именно:

- классы – в таблицы;
- атрибуты – в столбцы;
- типы – в типы данных СУБД;
- ассоциации – в связи между таблицами (в том числе, создавая дополнительные таблицы связей);
- приложения – в классы с определенными методами и атрибутами, связанными с данными в базе.



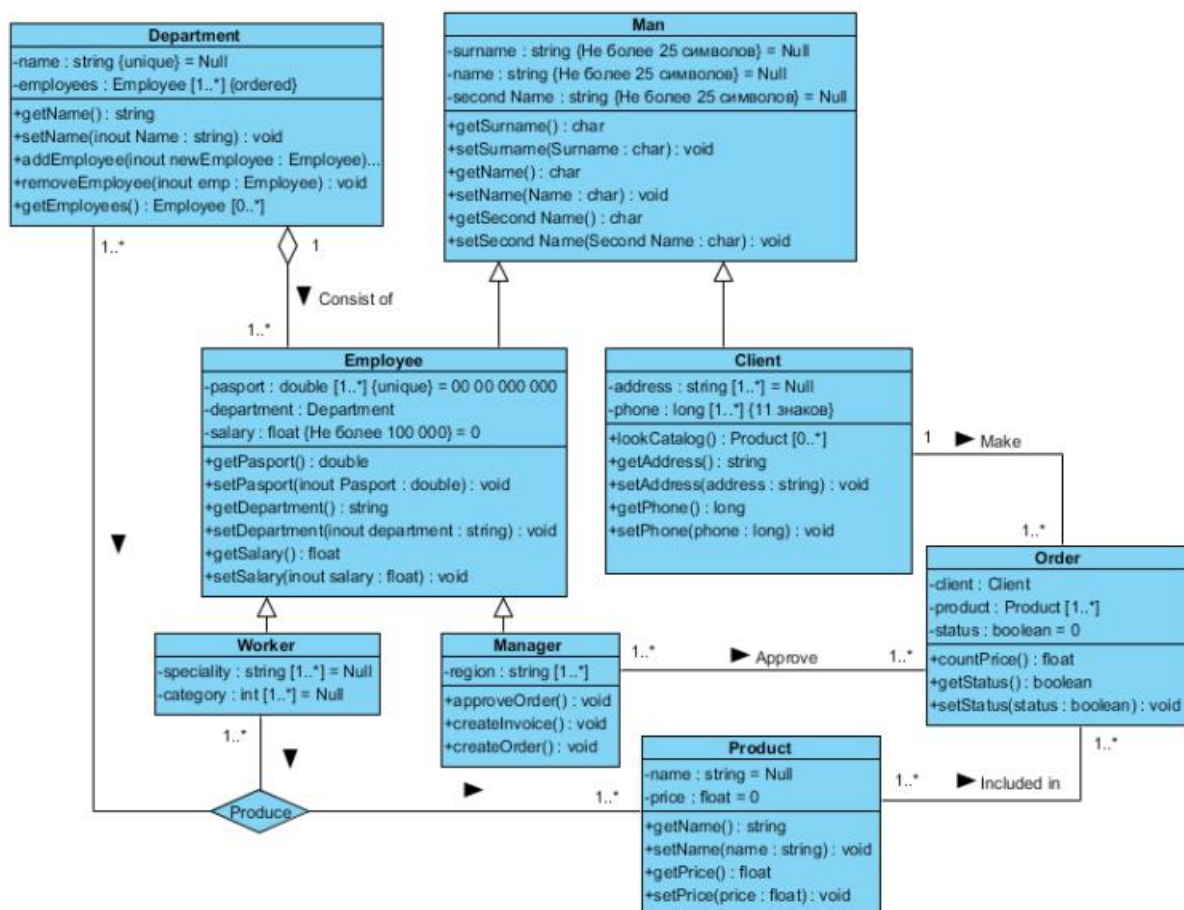


Рис. 2.16. Диаграмма классов

В модели базы данных для каждого простого класса формируется таблица, которая включает столбцы, поставленные в соответствие атрибутам класса [17].

Классы подтипов могут отображаться в таблицы несколькими способами:

- в случае отображения одной таблицы на класс отдельная таблица формируется для каждого класса с установлением последующих связей;
- в случае отображения одной таблицы на суперкласс для суперкласса создается таблица, а затем в таблицы подклассов размещаются столбцы для атрибутов суперкласса;
- в случае отображения одной таблицы на иерархию формируется единая таблица, в которой расположены атрибуты суперкласса и всех подклассов с добавлением дополнительных столбцов для определения исходных таблиц подклассов.

Язык UML для разработки проекта БД предлагает специальный профиль Profile for Database Design, в котором используются основные компоненты диаграмм, такие, как: таблица, столбец, ключи, связи, домен, процедура и т.д. На диаграммах также можно указывать дополнительные характеристики столбцов и таблиц: ограничения, триггеры, типы данных и т.д. В результате этого этапа проект базы данных и приложений системы становится детально описанным.

### 2.3.3. Проектирование физической модели ИС

Следующий этап проектирования системы включает в себя дополнения модели баз данных и приложений диаграммами их размещения на технических средствах.

На данном этапе рассматриваются следующие понятия UML [38]:

- **компонент** – элемент физического представления системы, реализующий определенный набор интерфейсов;
- **зависимость** – связь между двумя элементами, обозначающая ситуацию, при которой изменение одного элемента модели влечет за собой изменение ее другого элемента;
- **процессор и устройство** – узел, выполняющий и не выполняющий обработку данных соответственно;
- **соединение** – связь между процессорами и устройствами.

Наиболее полно особенности физического представления системы на языке UML позволяют диаграммы компонентов и диаграммы развертывания.

**Диаграмма компонентов** (component diagram) отображает иерархию подсистем, структурных компонентов и зависимостей между ними. Физическими компонентами выступают базы данных, исполняемые файлы, приложения, библиотеки, интерфейсы ИС и т.д.

В случае использования диаграммы компонентов для отображения внутренней структуры компонентов, интерфейсы составного компонента делегируются в определенные интерфейсы внутренних компонентов.

Основными целями построения диаграмм компонентов являются:

- определение архитектуры проектируемой системы;
- построение концептуальной и физической моделей баз данных;
- представление структуры исходного и исполняемого кодов системы;
- многократное использование определенных фрагментов программного кода.

Пример диаграммы компонентов представлен на рис. 2.17.

Для описания аппаратной конфигурации ИС применяют **диаграмму развертывания** (deployment diagram). С помощью диаграммы развертывания моделируют физическое распределение различных программных, информационных, аппаратных компонентов системы по комплексу технических средств. Особое внимание на диаграмме развертывания уделяется отображению того, какие используются аппаратные компоненты («узлы» - серверы баз данных и приложений, веб-верверы), какие программные компоненты («артефакты» - базы данных, веб-приложения) работают на каждом из них и как части комплекса соединены друг с другом. Таким образом, в случае проектирования сложной ИС ее необходимо разделить на части и исследовать каждую часть отдельно.

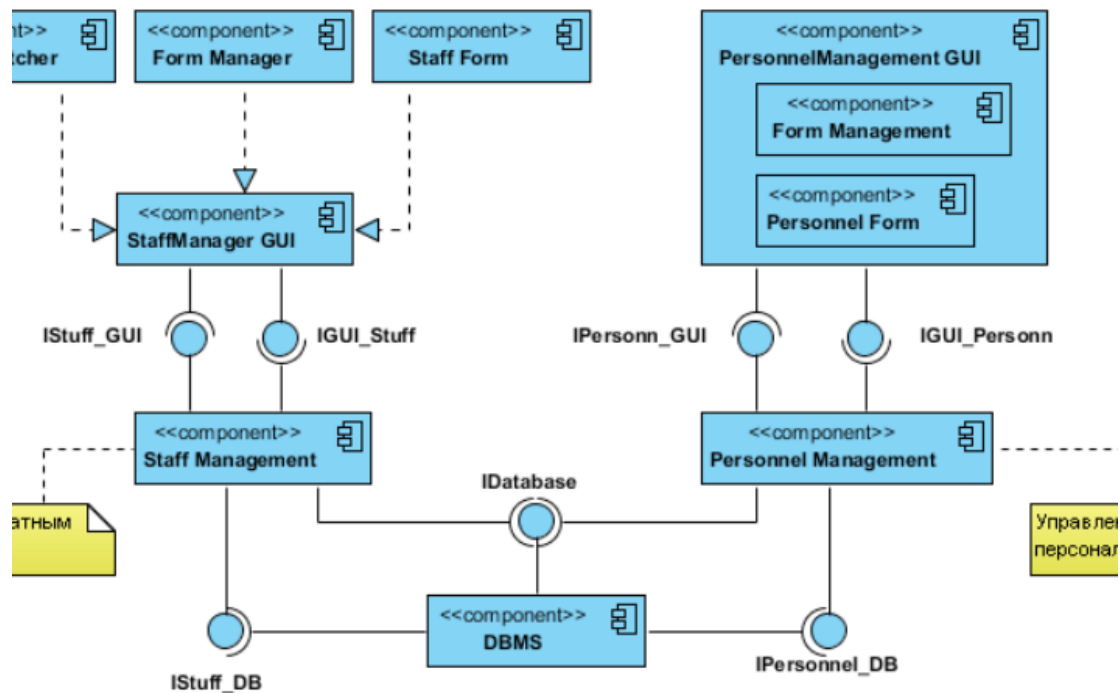


Рис. 2.17. Диаграмма компонентов

Пример диаграммы развертывания представлен на рис. 2.18. Характерной особенностью объектной декомпозиции является выделение объектов, взаимодействующих между собой, выполняющих определенные функции (методы) объекта.

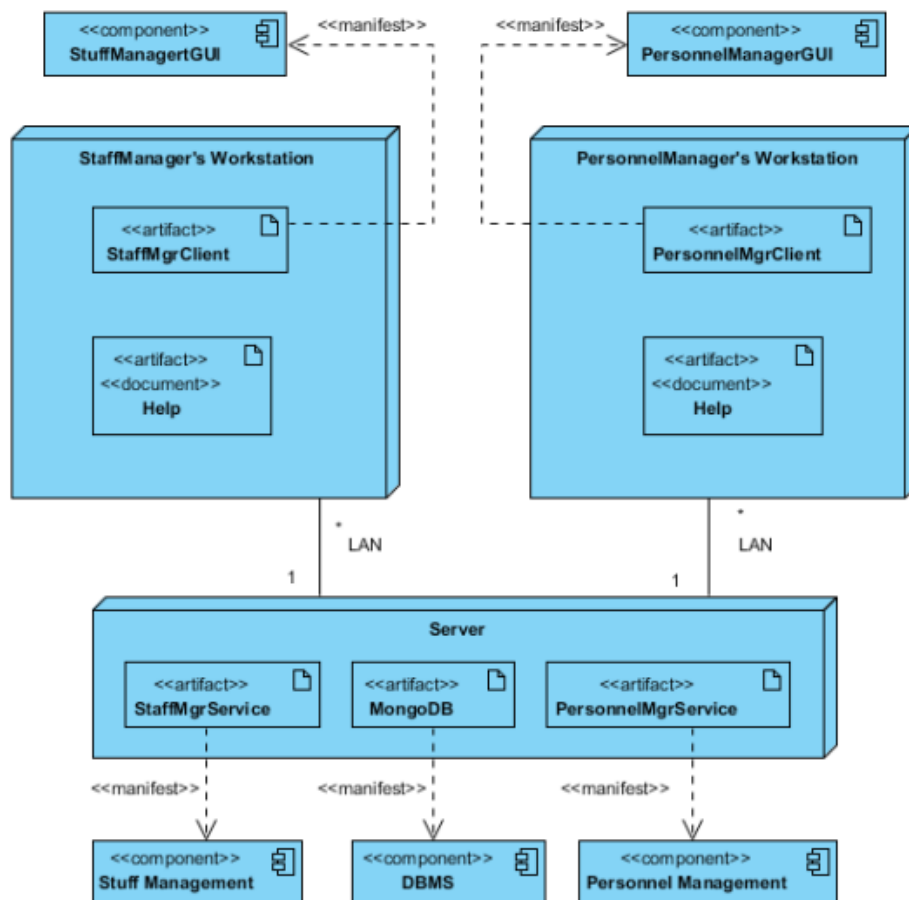


Рис. 2.18. Диаграмма развертывания

### 3. ПРОЕКТИРОВАНИЕ СЕТЕВОЙ КОМПЬЮТЕРНОЙ ИНФРАСТРУКТУРЫ ИНФОРМАЦИОННЫХ СИСТЕМ

#### 3.1. Основы проектирования компьютерных сетей

##### 3.1.1. Этапы проектирования компьютерных сетей

**Техническое обеспечение** – это комплекс технических средств, предназначенных для работы информационной системы, а также соответствующая документация на эти средства и технологические процессы.

**Комплекс технических средств** – это совокупность взаимосвязанных технических средств, предназначенных для автоматизированной обработки данных. В комплексе технических средств выделяются:

- средства сбора и регистрации информации (датчики, измерительные устройства, сканеры и т.д.);
- комплекс средств передачи информации (компьютерные сети, средства телефонной и телеграфной связи, радиосвязи, спутниковой связи и т.д.);
- средства хранения данных (накопители различных типов);
- средства обработки данных (компьютеры);
- средства вывода информации (мониторы, принтеры, плоттеры);
- средства оргтехники.

Основой технического обеспечения современных ИС являются **компьютерные сети**, образующие сетевую компьютерную инфраструктуру ИС.

**Компьютерная сеть** – это совокупность компьютеров, соединенных с помощью каналов связи и средств коммутации в единую систему для обмена сообщениями и доступа пользователей к программным, техническим, информационным и организационным ресурсам сети [45].

Процесс проектирования сетевой инфраструктуры ИС носит *итерационный характер*. Итерации могут включать в себя более чем один уровень проектирования, когда приходится многократно выполнять процедуру *анализа объекта*. Поэтому очевидно стремление уменьшить трудоемкость каждого варианта анализа без ущерба для качества окончательного проекта. В этих условиях целесообразно на начальных стадиях проектирования, когда высокой точности результатов не требуется, использовать наиболее простые и экономичные модели [29].

**Создать проект сети означает** выбрать структуру сети, определить значения всех его параметров и представить результаты в установленной форме. Результаты (проектная документация) могут быть выражены в виде схем, пояснительных записок, программ и других документов на бумаге или на машинных носителях информации [47]. Разработка (или выбор) структуры сети есть проектная процедура **структурного синтеза**, а расчет (или выбор) значений параметров элементов – процедура **параметрического синтеза**.

**Задача структурного синтеза** заключается в определении цели, множества возможных решений и ограничивающих условий. Задача заключается в синтезе (или коррекции) структуры, определении типов серверов (программно-аппаратных средств), распределении функций по серверам таким образом, чтобы достигался экстремум целевой функции при выполнении заданных ограничений.

Следующая после синтеза группа проектных процедур – процедуры анализа. **Цель анализа** – получение информации о характере функционирования и значениях выходных параметров при заданной структуре объекта, сведениях о внешних параметрах и параметрах элементов.

Проектирование компьютерных сетей выполняется в **несколько этапов** [41]:

- 1) предпроектный этап;
- 2) общий структурный синтез сети;
- 3) разработка протоколов и средств административной системы сети;
- 4) проектирование программно-технических средств сети;
- 5) проектирование топологии сети;
- 6) распределение ресурсов между абонентами.

На **первом этапе** производится анализ исходных данных для проектирования, выясняется назначение проектируемой компьютерной сети, режимы ее функционирования, определяются показатели эффективности функционирования, ограничения и формируются допущения.

На **втором этапе** формируется задача синтеза общей структуры компьютерной сети. Определяется число уровней иерархии сети, методы управления сетью, система адресации узлов, состав базовых протоколов сети, функции административной системы сети, структура сети передачи данных, места расположения центров управления сетью, типы используемых каналов связи. Рассчитываются основные показатели эффективности для выбранного варианта структуры сети: технические, временные, экономические.

На **третьем этапе** разрабатываются или выбираются алгоритмы сетевых протоколов и административной системы сети.

На **четвертом этапе** производится реализация нестандартных программно-технических средств сети, в том числе программная реализация сетевых протоколов и административной системы сети.

На **пятом этапе** обследуются объекты, где размещаются АРМы; рассчитывается требуемый поток между узлами сети; уточняются технические требования на сеть; формируется перечень междугородных трактов, которые могут быть использованы для выделенных каналов связи сети передачи данных; разрабатывается *математическая модель задачи синтеза передачи данных* и производится синтез топологии компьютерной сети. При **решении задач синтеза** распределяется требуемый *поток данных*; устанавливаются выделенные каналы связи и их

пропускная способность; определяется требуемая пропускная способность трактов сети связи; распределяются базы данных и крупные функциональные задачи по узлам сети; определяется рациональная длина пакета; формируются маршруты передачи данных; оцениваются основные показатели эффективности спроектированной сети.

На **шестом этапе** производится распределение вычислительных, обменных и информационных ресурсов сети между абонентами.

Методы проектирования компьютерных сетей в основном связаны с синтезом топологии сети, которые хорошо формализуются. Однако практические задачи проектирования топологии компьютерной сети осложняются наличием двух мешающих факторов: *большой размерности сети* и *неопределенности исходных данных*.

Точные методы синтеза топологии для размерности сети порядка нескольких сотен узлов непригодны, и возникает проблема разработки эффективных **приближенных методов структурного синтеза** сетей большой размерности.

Отсутствие достоверных исходных данных при проектировании сетей (например, объема вычислительных работ, интенсивностей потоков сообщений пользователей) порождает вторую проблему – **синтеза компьютерных сетей в условиях неопределенности**. Для разрешения этой проблемы проводят многовариантные расчеты при варьировании параметрами в широком диапазоне с анализом устойчивости получаемых структур.

При проектировании сети используется **декомпозиционный подход**. Путем декомпозиции осуществляется сведение единой сложной задачи к множеству локальных слабосвязанных подзадач.

Проектирование топологии сети охватывает следующие задачи:

- 1) выбор общей структуры сети;
- 2) размещение центров коммутации пакетов (ЦКП);
- 3) синтез топологии подключения абонентских компьютеров к ЦКП и выбор пропускных способностей абонентских линий связи;
- 4) синтез абонентских систем передачи данных;
- 5) синтез топологии сети передачи данных базовых ЦКП;
- 6) распределение требуемого потока в сети базовых ЦКП и выбор пропускных способностей арендуемых каналов связи;
- 7) выбор рациональной длины пакета;
- 8) распределение задач между ЭВМ сети;
- 9) распределение баз данных между ЭВМ сети;
- 10) распределение вычислительных ресурсов сети между абонентами;
- 11) выбор структуры ЛВС;
- 12) расчет характеристик эффективности спроектированной сети;
- 13) расчет требуемой производительности объема памяти ЦКП.

### 3.1.2. Теоретические основы проектирования компьютерных сетей

При проектировании компьютерных сетей, помимо общих подходов к проектированию информационных и вычислительных систем [28, 41, 49, 51], необходимо использовать теоретические основы проектирования сложных систем для обоснования архитектуры, параметров и показателей качества проектируемых объектов. При этом активно используется теория очередей [29], теория надежности [48], теория живучести [35], теория управления потоками [50] и т.д.

Одним из первых этапов проектирования компьютерной сети является *выбор технических средств и системы протоколов* (включая способы коммутации и доставки данных в сети). Последующие этапы проектирования требуют решения совокупности сложных взаимосвязанных задач, к которым относятся:

- оптимизация пропускных способностей каналов связи;
- выбор маршрутов; оптимизация топологической структуры;
- выбор методов управления потоками и определение параметров управления;
- анализ объемов буферной памяти узлов коммутации и маршрутизации;
- выбор стратегии буферизации при перегрузках и т.д.

Формально задача проектирования глобальной сети сводится к отысканию минимума функционала приведенной стоимости

$$C(U, \Omega, Y) \rightarrow \min \quad (3.1)$$

при наличии ограничений на вероятностно-временные и структурные характеристики сети

$$V_i(U, \Omega, Y) \leq V_{i0} \quad (3.2)$$

и требования принадлежности множества вариантов архитектуры сети, удовлетворяющих ограничениям (6.2), к области технически реализуемых решений

$$Q(U, \Omega, Y) \in Q_0 \quad (3.3)$$

Здесь  $U$  – векторная величина, отражающая параметры сетевой нагрузки, включая интенсивности потоков сообщений между каждой парой узлов коммутации сети, распределение длин сообщений, приоритетность потоков сообщений и т.д.;  $\Omega$  – векторная величина, представляющая собой совокупность параметров технических средств, включая производительность узлов коммутации и каналообразующей аппаратуры, надежность технических средств, достоверность передачи информации и т.д.;  $Y$  – векторная величина, отражающая параметры логической структуры сети.

Средством решения общей задачи проектирования является создание комплекса математических моделей и программ проектирования компьютерной сети. Высокое качество проектирования может быть достигнуто, если отдельные методы и модели объединены на основе системного подхода в единую систему

проектирования, охватывающую все или большую часть задач проектирования.

Практическая невозможность постановки и решения в рамках одной математической задачи всего комплекса проблем проектирования сети приводит к необходимости использования процедуры декомпозиции. Такая декомпозиция возможна как на структурном уровне, так и на уровне решения отдельных задач проектирования и позволяет перейти от задачи большой размерности к последовательности задач меньшей размерности.

**Декомпозиция сети на структурном уровне** означает, что проектирование компьютерной сети сводится к независимому проектированию ряда подсетей *при соблюдении условий* совпадения или близости оптимальных решений задачи проектирования сети и соответствующих решений для подсетей.

Примеры условий:

- подсети по области ограничений должны быть независимы;
- целевая функция сети является строго монотонной функцией от целевых функций подсетей.

В качестве критерия выбирают обобщенный экономический критерий. Другие критерии (среднее время задержки, надежность) используются в качестве ограничения при решении задачи проектирования.

**Декомпозиция глобальной компьютерной сети на уровне проектирования базовой и региональной сетей** означает создание многоуровневой иерархии взаимосвязанных моделей, анализ которых позволяет получить решение общей задачи проектирования для каждой из таких сетей и тем самым решение задачи проектирования компьютерной сети в целом.

Таким образом, в основу проектирования компьютерной сети положены следующие общие принципы:

- 1) интерактивности;
- 2) независимости;
- 3) многоуровневого моделирования;
- 4) адаптивности и развития.

Создание крупного проекта неизбежно связано с поэтапным вводом в эксплуатацию и развитием сети. При проектировании нужно учитывать динамику развития сети, возможность проектировать сеть при изменяющихся исходных данных и параметрах с минимальными затратами на изменение моделей и соответствующих программ. Более того, для решения одной и той же задачи проектирования необходим набор математических моделей, позволяющих находить как точное решение для сети небольшой размерности, так и приближенное решение для сетей большой размерности.

Наряду с общими принципами непосредственно при проектировании компьютерных сетей используется ряд частных принципов.



### 3.1.3. Анализ требований к компьютерным сетям

**Главным требованием**, предъявляемым к компьютерным сетям, является выполнение сетью ее основной функции – обеспечение пользователям потенциальной *возможности доступа к разделяемым ресурсам* всех компьютеров, объединенных в сеть. Все остальные требования связаны с качеством выполнения этой основной задачи.

**Основными требованиями, предъявляемыми к сетям, являются:**

- 1) производительность;
- 2) надежность;
- 3) совместимость;
- 4) управляемость;
- 5) защищенность
- 6) расширяемость и масштабируемость;
- 7) прозрачность;
- 8) поддержка разных видов трафика;
- 9) интегрируемость.

**Производительность** – это одно из основных свойств распределенных систем, к которым относятся компьютерные сети. Это свойство обеспечивается возможностью распараллеливания работ между несколькими компьютерами сети. Существует несколько основных характеристик производительности сети:

- время реакции;
- пропускная способность;
- задержка передачи и вариация задержки передачи.

**Надежность системы** определяется надежностью работы всех ее компонентов. Для повышения надежности работы аппаратных компонентов обычно используют дублирование, когда при отказе одного из элементов функционирование сети обеспечат другие.

**Совместимость** – способность сети включать в себя самое разнообразное программное и аппаратное обеспечение (различные ОС, поддерживающие разные стеки коммуникационных протоколов, аппаратные средства и приложения от разных производителей). Сеть, состоящая из разнотипных элементов, называется *неоднородной* или *гетерогенной*, а если гетерогенная сеть работает без проблем, то она является интегрированной. Основной путь построения интегрированных сетей – использование модулей, выполненных в соответствии с открытыми стандартами и спецификациями.

**Управляемость сети** – возможность централизованно контролировать состояние основных элементов сети, выявлять и разрешать проблемы, возникающие при работе сети, выполнять анализ производительности и планировать развитие

сети. При работе компьютерной сети, которая объединяет отдельные компьютеры в единое целое, необходимы не только средства для наблюдения за работой сети, сбора разнообразной информации о функционировании сети, но и средства управления сетью.

**Главная задача управления ИС** сводится к обеспечению адаптации внутреннего состояния системы к постоянно изменяющейся внешней среде.

**Защищенность сети** – способность системы защитить данные от несанкционированного доступа. В распределенной системе это сделать гораздо сложнее, чем в централизованной. В сетях сообщения передаются по линиям связи, часто проходящим через общедоступные помещения, в которых могут быть установлены средства прослушивания линий. Другим уязвимым местом могут быть оставленные без присмотра персональные компьютеры. Кроме того, всегда имеется потенциальная угроза взлома защиты сети от неавторизованных пользователей, если сеть имеет выходы в глобальные сети общего пользования.

**Расширяемость** – возможность сравнительно легкого добавления отдельных элементов сети (пользователей, компьютеров, приложений, служб), наращивания длины сегментов сети и замены существующей аппаратуры более мощной.

Лёгкость расширения системы иногда может обеспечиваться в некоторых весьма ограниченных пределах.

*Например, односегментная локальная сеть Ethernet обладает хорошей расширяемостью, в том смысле, что позволяет легко подключать новые станции. Однако такая сеть имеет ограничение на число станций – их число не должно превышать 30-40. Хотя сеть допускает физическое подключение к сегменту и большего числа станций (до 100), но при этом чаще всего резко снижается производительность сети. Наличие такого ограничения является признаком плохой масштабируемости системы при хорошей расширяемости.*

**Масштабируемость** – возможность наращивать количество узлов и протяженность связей в очень широких пределах, без существенного снижения производительности. Для обеспечения масштабируемости сети приходится применять **дополнительное коммуникационное оборудование** и специальным образом структурировать сеть.

*В одноранговых сетях легко добавить новый компьютер, используя дополнительный кабель и сетевой адаптер, но существуют ограничения на количество подключаемых компьютеров в связи с существенным падением производительности сети. Хорошей масштабируемостью обладают многосегментные сети, построенные с использованием коммутаторов и маршрутизаторов и имеющие иерархическую структуру связей.*

**Прозрачность сети** предполагает скрывание (невидимость) особенностей сети от конечного пользователя. Прозрачность сети достигается в том случае, когда

сеть представляется пользователям не как множество отдельных компьютеров, связанных между собой сложной системой кабелей, а как **единая вычислительная машина**. Пользователь обращается к ресурсам сети как к обычным локальным ресурсам компьютера, на котором он работает.

**Поддержка разных видов трафика** – возможность совмещения в одной сети традиционного компьютерного (файлы, программы) и мультимедийного трафика (голос, видео, изображение). Главной особенностью **мультимедийного трафика** является наличие жестких требований к синхронности передаваемых сообщений. **Компьютерный трафик** характеризуется крайне неравномерной интенсивностью поступления сообщений в сеть при отсутствии жестких требований к синхронности доставки этих сообщений.

**Интегрируемость** означает возможность подключения к вычислительной сети **разнообразного и разнотипного** оборудования, ПО от разных производителей. Если такая неоднородная вычислительная сеть успешно выполняет свои функции, то можно говорить о том, что она обладает хорошей интегрируемостью.

**Основным направлением** развития интегрируемости вычислительных сетей является стандартизация сетей, их элементов и компонентов.

### **3.1.4. Основные модели и характеристики проектируемых компьютерных сетей**

Синтез перспективных СВТ, анализ известных инженерных решений, выбор оптимального варианта проектирования требуют постоянного развития и совершенствования моделей, описывающих все более сложные объекты проектирования [29]. Для сложных систем, таких как компьютерные системы и сети, рассмотрение одной или нескольких не связанных между собой моделей явно недостаточно. Определение эффекта можно достичь лишь при тщательном исследовании основных параметров системы на всех уровнях функционирования.

**Целью моделирования** будем считать получение информации об особенностях организации, управления и физической структуры компьютерных систем и сетей. Выбор моделей для исследования на каждом уровне производится в соответствии со следующими критериями: адекватности структуре (функции) объекта исследования; адекватности поставленной цели исследования; наглядности; доступности для компьютерного анализа.

Всякая ИВС является системой обработки информации – приема, передачи, изменения и выдачи ее в виде результатов. Компьютерная система (сеть) представляется как *комплекс программно-аппаратных средств*. На самом абстрактном уровне моделирования следует выделить три основных компонента этого комплекса: *внешняя среда – ИВС – информация*. Для упрощения всего комплекса моделей примем, что внешняя среда отрицательно воздействует на остальные

компоненты объекта. Определим следующие уровни моделирования заданного комплекса: уровень общего описания (функциональный), уровень архитектуры, уровень управления и диспетчеризации, уровень программной и аппаратной поддержки, уровень элементарных составляющих (табл.3.1).

Таблица 3.1.

**Уровни моделирования и проектирования ИВС**

Уровень	Задачи исследования	Модели	Аппарат исследования
Функциональный	Типы ИВС, типы целей функционирования, определение функц. модулей, принципы связи и управления	Качественные, теоретико-игровые, имитационные, развивающиеся процессы	Теория прогнозирования, теория игр
Архитектуры	Синтез оптимальной структуры ИВС, проектирование средств и механизмов обеспечения надежности, живучести, устойчивости	Вероятностные, логико-динамические, имитационные, структурно-функциональные схемы, сетевые модели	Теория вероятностей, теория игр, теория очередей, теория потоков, теория проектирования, имитационное моделирование
Управления и диспетчеризации	Дисциплины распределения ресурсов, обслуживания потоков задач, организация реконфигурации, адаптации, восстановления	спец. модели отказов, сбоев, ошибок, имитационные модели управления, реконфигурации, адаптации	теория вероятностей, теория расписаний, оптимальное управление, теория принятия решений, теория восстановления
Программной и аппаратной поддержки	Выделение критических функций, определение состава функц. модулей, разработка модульной структуры программных и аппаратных частей	Модели восстановления, резервирования, реконфигурации по всем видам ресурсов	Имитационное моделирование, теория автоматов, теория восстановления, теория расписаний
Элементарных составляющих	Принципы компоновки и размещения, меры по предотвращению физических отказов	Инженерные модели расчета надежности	Теория надежности, теория проектирования

**На функциональном уровне** определяется множество функций, исполняемых системой. Задается периодичность и основные режимы работы системы, определяются принципы связи и управления, обусловленные областью применения. Задается иерархичность структуры ИВС. Определяются наиболее опасные и наиболее вероятные воздействия внешней среды на ИВС и на информацию.

**На уровне архитектуры** конкретизируются определенные ранее основные функции. Отбираются структура минимальной аппаратной и программной частей системы, средства связи и управления, конкретное влияние внешней среды на ресурсы системы и их учет при выборе аппаратного решения.

**На уровне управления и диспетчеризации** определяется иерархия управления всеми ресурсами системы от самых высших до самых нижних уровней, ста-

вится задача обеспечения синхронизации и передачи информации между отдельными подсистемами, организации взаимодействия подсистем, определяется способ распределения ресурсов, в том числе при отказах.

**На уровне программной и аппаратной поддержки** главная задача – обеспечить некоторое оптимальное соотношение между программной и аппаратной частями системы, добившись максимизации критериев производительности, живучести и гибкости системы. Существенным моментом моделирования на этом уровне представляется вопрос о взаимозаменяемости аппаратных и программных модулей.

**На уровне элементарных составляющих** обеспечивается выполнение всех функций более высоких уровней.

Для анализа компьютерной сети широко используются математические методы и модели **теории массового обслуживания** [25]. Упрощенные модели массового обслуживания позволяют находить явный вид целевой функции, в качестве которой используется характеристика сети, такая как время задержки сообщений. Моделирование жизненного цикла компьютерной сети проводится по итеративной модели — это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов работы. Компьютерная сеть при этом подходе в каждой стадии проходит повторяющийся цикл: планирование – реализация – проверка – оценка.

В автоматизированных проектных процедурах вместо проектируемого еще не существующего объекта оперируют моделью, которая отражает некоторые интересующие исследователя свойства объекта.

**Математические функциональные модели** в общем случае представляют собой алгоритмы вычисления вектора выходных параметров при заданных векторах параметров элементов и внешних параметров.

Закон функционирования сети может быть представлен в следующем виде:

$$H(t) = f_c(S, F, Y, X, t),$$

где  $f_c$  – функция, функционал, логические условия, алгоритм, методика, таблица или словесное описание, определяющее правило (закон) преобразования входных величин (параметров) в выходные величины (характеристики);

$H(t)$  – вектор характеристик, зависящий от текущего момента времени  $t$  ( $t \geq 0$ ):  $H = \{V, T, N, C, Z\}$ .

**Параметры – первичные данные сети:**

- $S$  – структурные,
- $F$  – функциональные,
- $Y$  – нагрузочные,
- $X$  – внешней среды.

**Характеристики – вторичные данные сети:**

- V – мощностные,
- T – временные,
- N – надёжностные,
- C – экономические,
- Z – прочие.

Понятие **характеристики функционирования сети** включает вторичные свойства компьютерной сети, которые определяются в процессе решения задач анализа, как функция параметров. **Параметры компьютерной сети** описывают первичные свойства сети и являются исходными данными при решении задач анализа.

Характеристики компьютерных сетей – это совокупность показателей эффективности (качества) сети. Характеристики компьютерных сетей можно разделить на **качественные и количественные**.

Количественные характеристики можно разделить на две группы:

- **глобальные**, определяющие наиболее важные свойства сети как целостного объекта;
- **локальные**, определяющие свойства отдельных устройств или частей сети и позволяющие получить более детальное представление об эффективности сети.

К **глобальным** относятся характеристики производительности, оперативности, надёжности, стоимостные, прочие (энергопотребления, массогабаритные).

В качестве **локальных характеристик** компьютерных сетей могут использоваться в зависимости от целей исследования самые разнообразные показатели эффективности.

Локальные характеристики описывают эффективность функционирования:

- узлов и каналов связи;
- отдельных сегментов сети;
- узлов обработки данных: вычислительной системы и её подсистем.

Локальные характеристики могут быть разбиты на две группы:

1) **временные**; 2) **безразмерные**.

К **временным характеристикам** относятся:

- время доставки пакетов при передаче между соседними узлами сети;
- время ожидания передачи данных в узлах сети или освобождения ресурсов ВС;
- время пребывания данных в различных узлах, устройствах или подсистемах.

К **безразмерным характеристикам** относятся:

- число пакетов, находящихся в буферной памяти узлов (маршрутизаторов, коммутаторов);
- коэффициенты загрузки узлов, каналов связи и устройств ВС и т.д.

## 3.2. Проектирование локальных и глобальных сетей

### 3.2.1. Общие вопросы проектирования локальных и глобальных сетей

После анализа и оценки возникающих потребностей Заказчика по созданию компьютерной сети необходимо учесть ряд факторов, влияющих на проект локальной или глобальной сети.

Основными факторами, влияющими на структуру локальных и глобальных сетей, являются [46]:

- 1) ожидаемый сетевой трафик;
- 2) требования по избыточности;
- 3) перемещения пользователей;
- 4) перспективное развитие;
- 5) требования безопасности;
- 6) подключение к глобальным сетям;
- 7) стоимость.

При развертывании сети необходимо хорошо представлять себе *объем трафика*, предполагаемого в сети. В новых сетях нужно учитывать количество пользователей и тип серверов или хостов, которые будут работать в сети.

Другим определяющим фактором является *необходимость в резервных сетевых путях* для передачи данных. Иногда наличие ЛВС важно для пользователей, однако в силу характера их деятельности они могут некоторое время обходиться без сети, если, например, часть сети выйдет из строя из-за неисправности коммутатора. В других ситуациях требуются резервные сетевые маршруты для перераспределения трафика – чтобы пользователи никогда не замечали неисправностей оборудования. Наличие *избыточности* – это требование тех организаций, которые могут потерять значительные убытки за каждую минуту простоя части сети.

В некоторых организациях пользователям нужно обеспечить *физическое передвижение по зданию*, имея при себе портативный компьютер. Это также следует учитывать для правильного выбора топологии беспроводной сети, позволяющей таким пользователям получать полный доступ ко всем сетевым ресурсам.

Все перспективные сети должны предусматривать *возможность роста*, для чего в них нужно заложить проектные решения. При соблюдении этого условия средства, вложенные в сеть, окупаются в течение многих лет.

Хотя некоторые средства безопасности нужны для большинства сетей, не для всего сетевого трафика нужен одинаковый *уровень защиты*.

Подключение к глобальным сетям также является важным фактором, учитываемым при проектировании локальной сети. Но для разных локальных сетей необходимы *различные возможности* подключения к глобальным сетям.

При осуществлении любого проекта ЛВС и ГВС необходимо учитывать расходы. В большинстве организаций расходы на развертывание сети ограничены

(определяются некоторым бюджетом или суммой, выделенной на проект).

В проекте сети необходимо учитывать стоимость следующих составляющих:

- коммуникационный кабель;
- сетевые устройства;
- дополнительные компьютеры, необходимые в сети;
- программные и аппаратные средства анализа и управления сетью;
- монтажные работы;
- обучение;
- консультации поставщиков;
- плата за услуги глобальной сети или за выделенные линии.

### 3.2.2. Планирование сетевой архитектуры

Рассмотрим главные моменты **планирования сетевой архитектуры** локальной сети, понятия **физической и логической топологии сети**.

Архитектура сети включает описание различных типов топологий:

- физическая топология;
- логическая топология;
- топология управления обменом;
- информационная топология.

**Физическая топология** относится к физической структуре сети и учитывает географическую схему расположения компьютеров и прокладки кабелей.

**Логическая топология** сети характеризует способ прохождения пакетов данных по сети, а также метод организации связи в сети, обеспечивающий одновременную работу «на передачу» только одной сетевой станции. Учитывает структуру связей, характер распространения сигналов по сети.

**Топология управления обменом** учитывает принцип и последовательность передачи права на захват сети между отдельными компьютерами.

**Информационная топология** учитывает направление потоков информации, передаваемой по сети.

Например, сеть с **физической и логической топологией *шина*** может в качестве **метода управления** использовать эстафетную передачу права захвата сети (быть в этом смысле ***кольцом***) и одновременно передавать всю информацию через выделенный компьютер (быть в этом смысле ***звездой***). Или сеть с **логической топологией *шина*** может иметь физическую топологию ***звезда*** (пассивная) или ***дерево*** (пассивное).

Выбор **физической топологии** сети зависит от нескольких факторов:

- 1) структуры помещений;
- 2) способов диагностики неисправностей;



- 3) стоимости инсталляции;
- 4) типа используемого кабеля.

Рассмотрим подробнее данные факторы.

1. При установке нескольких компьютеров в **одну комнату** появляется больше возможных вариантов организации сети, чем в случае, когда множество компьютеров распределяется по различным комнатам здания.

2. Наличие **методов и средств диагностирования неисправностей** зависит в какой-то мере от используемой физической топологии.

3. **Стоимость инсталляции.** Не все физические топологии эквивалентны друг другу по стоимости. Некоторая доля в стоимости, несомненно, определяется планом помещений. Разводка сети, расположенной в обширной области, более трудоёмка, и в стоимости будут отражаться эти дополнительные усилия. Однако часть стоимости определяется **сложностью** выбранной вами топологии, и, что еще более важно, тем, насколько сложно эту топологию привести в соответствие с пространством помещений.

4. Выбор физической топологии в значительной степени определяется **типом кабеля** и наоборот.

Сеть с любой топологией (физической, логической, управления обменом) может считаться **звездой в смысле информационной топологии**, если она построена на основе одного сервера и нескольких клиентов, общающихся только с этим сервером.

Точно так же любая сеть может быть названа **шиной в информационном смысле**, если она построена из компьютеров, являющихся одновременно как серверами, так и клиентами. Такая сеть будет мало чувствительна к отказам отдельных компьютеров.

Наиболее известные физические топологии:

- 1) физическая шинная топология;
- 2) физическая звездообразная топология;
- 3) распределенная физическая звездообразная топология;
- 4) физическая кольцевая топология.

Рассмотрим различные варианты **физической топологии** сети.

1. Для простых сетей, расположенных в пределах небольшой территории, **физическая шинная топология** может оказаться наилучшим решением. В топологии шины кабель идет от компьютера к компьютеру, связывая их в цепочке. Все компьютеры в сети связаны одним общим кабелем, как правило, коаксиальным (рис.3.1).

2. В сети, построенной по **звездообразной топологии**, каждый сервер и рабочая станция подключаются к центральному коммутатору (рис.3.2), который обеспечивает связь между ними. Центральной частью сети является коммутатор.

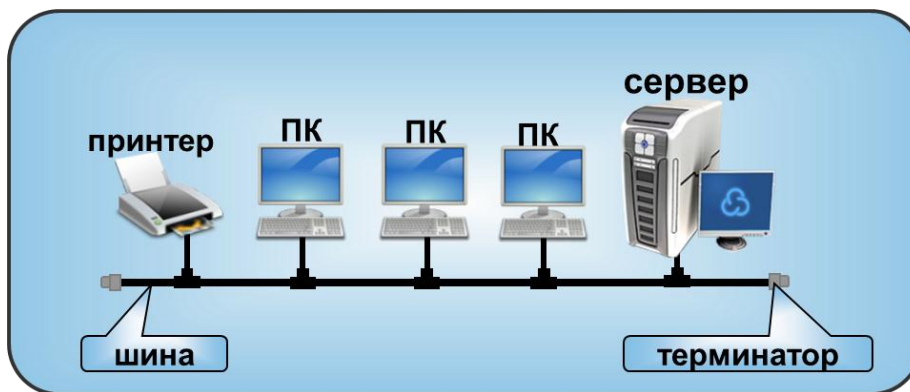


Рис. 3.1. Топология «шина»

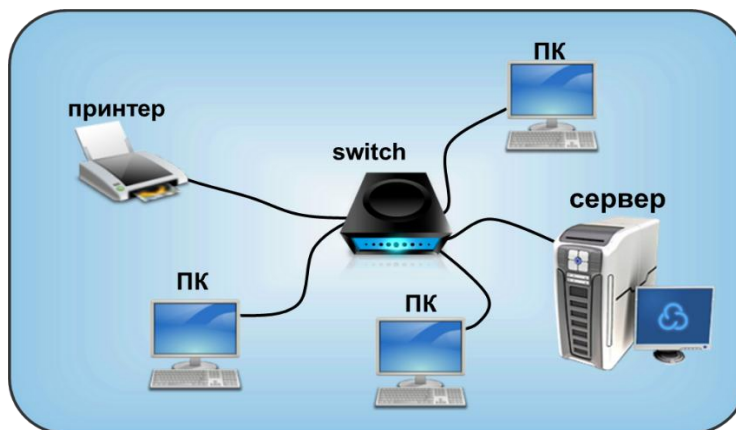


Рис. 3.2. Топология «звезда»

#### Преимущества:

- коммутатор реализуют центральный узел для всех сетевых кабелей, обеспечивая связь между портами,
- легко создать кабельную систему, не тревожась о расположении компьютеров в сети.
- каждое соединение работает независимо (обрыв кабеля не влияет на работу других);
- легко диагностировать неисправности.

#### Недостатки:

- используется много кабеля;
- длина отрезка кабеля от каждой рабочей станции или сервера до коммутатора ограничена (для «витой пары» – 90-100 м);
- при выходе коммутатора из строя сеть пропадает.

3. В топологии **распределенной звезды** коммутаторы сети последовательно подключены друг к другу, так что все они могут обмениваться информацией (рис.3.3). Такая организация сети имеет некоторые недостатки, свойственные сети, построенной по шинной топологии: разрыв кабеля между двумя коммутаторами изолирует части сети по обеим сторонам разрыва. Однако этот недостаток компенсируется тем, что при отсутствии шины коммутаторы были бы изолированы друг от друга в любом случае.

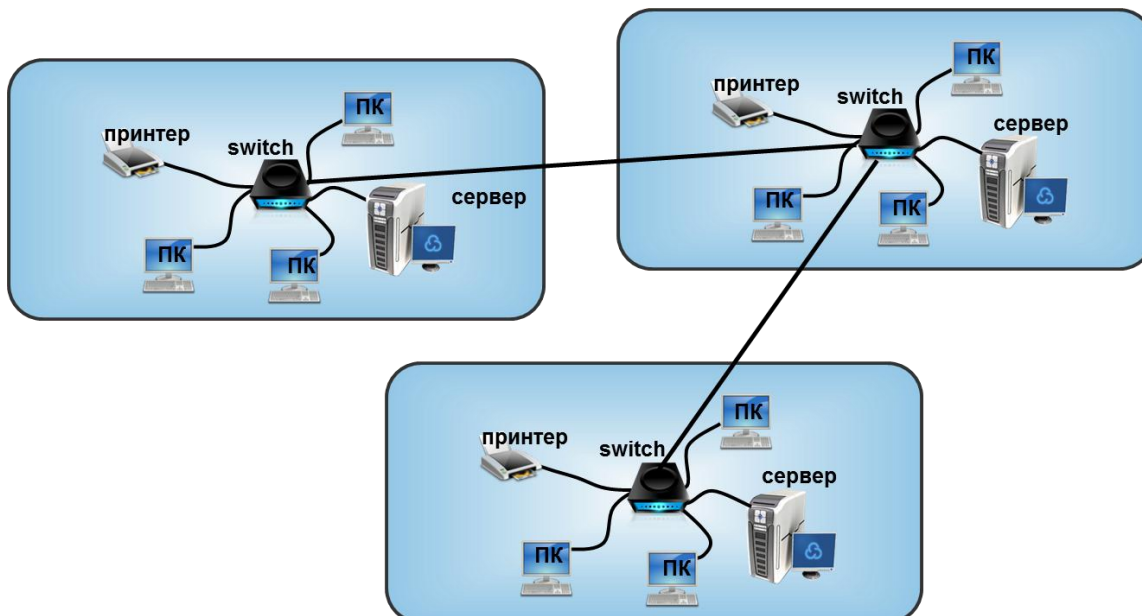


Рис. 3.3. Топология «распределенная звезда» («снежинка»)

4. В **физической кольцевой топологии** все персональные компьютеры сети для обеспечения целостности сети соединены в кольцо, выполненное в виде пары кабелей, проложенных между каждым узлом. Такая система вполне работоспособна, но её стоимость и трудоёмкость прокладки кабельной системы весьма велики, поскольку и такой сети затраты на кабель удваиваются (рис.3.4).

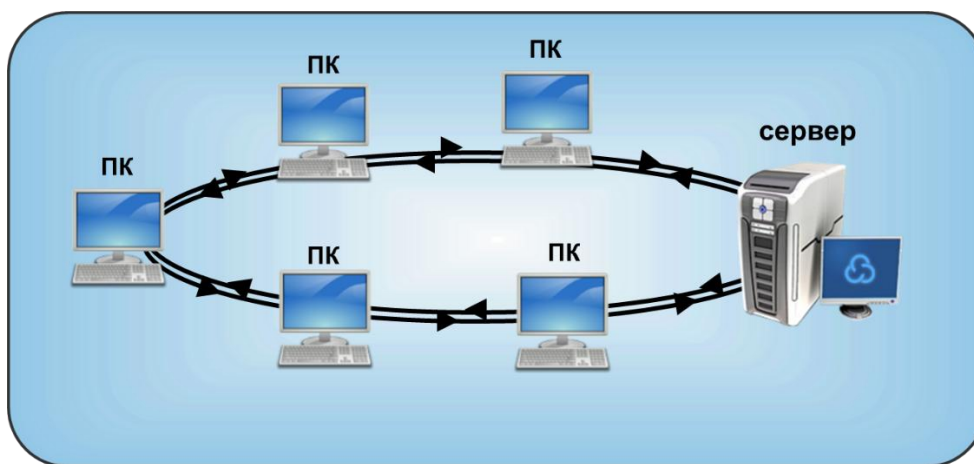


Рис. 3.4. Топология «кольцо»

**Логическая (или электрическая) топология** описывает способ, в соответствии с которым устройства сети передают информацию от одного узла к следующему, но нельзя путать его с теми линиями, которые нарисованы на структурной схеме сети. **Физическая топология не имеет прямого отношения к логической.** Сеть может иметь физическую звездообразную топологию и логическую кольцевую или физическую звездообразную и логическую шинную (рис.3.5) и т.д.

**Логическая шинная топология** является системой широковещательной передачи: то, что «скажет» одна станция, «слышат» все остальные. Когда у какого-либо узла сети оказываются данные для другого узла, то первый узел производит

"оповещение" всей сети. Все остальные узлы «слушают» сеть и проверяют, предназначены эти данные для них или нет. Если предназначены, то они «оставляют их себе», если нет — игнорируют.



Рис. 3.5. Физическая и логическая топологии

Сеть, построенная по **кольцевой топологии**, функционирует иначе. В таких сетях, как Token Ring и FDDI (Fiber Distributed Data Interface – распределенный интерфейс передачи данных по оптоволоконным каналам), каждая станция должна повторять то, что она услышит от предыдущей, работая при переносе данных подобно "пожарной цепочке". Когда порция данных возвращается их отправителю, передача прекращается. Весь файл не может быть передан целиком в виде одного пакета, поэтому он будет передаваться порциями.

Помимо классических топологий, получили развитие так называемые комбинированные топологии (рис.3.6).

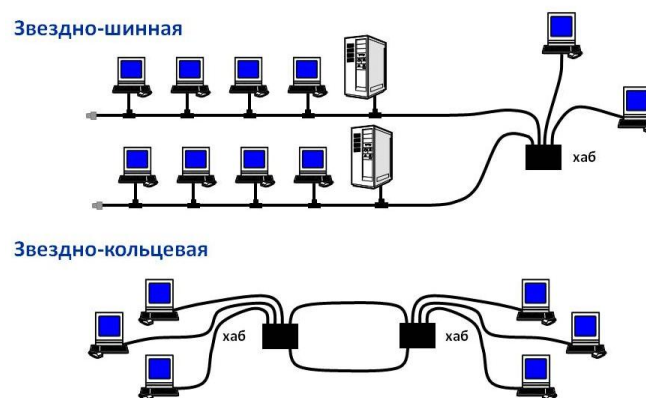


Рис. 3.6. Комбинированная топология

### 3.2.3. Принципы проектирования локальных сетей

В современных локальных сетях применяются маршрутизаторы и коммутаторы. Эти устройства позволяют использовать принципы построения структурированных кабельных систем и сетей. *Маршрутизаторы* обеспечивают сегментацию сетей и управление трафиком, а *коммутаторы* позволяют организовывать

отдельные области коллизий и повысить скорость пересылки сетевого трафика.

Во многих организациях трафик между подразделениями, как правило, ниже, чем внутри них. Рассмотрим локальную сеть организации, являющуюся защищенным сегментом корпоративной сети, где маршрутизатор используется в качестве брандмауэра. В смежном сегменте требования безопасности ниже или могут вообще отсутствовать. Тогда маршрутизатор может уменьшить общий сетевой трафик, ограничивая его тем сегментом, для которого он предназначен, при этом он также действует как брандмауэр, защищая одну или несколько сетей. Маршрутизаторы также используются для трансляции двух различных протоколов между подразделениями.

Коммутаторы изначально разрабатывались в качестве многопортовых мостов, но сегодня некоторые типы коммутаторов имеют функции маршрутизации Уровня 3, а некоторые даже Уровня 4, поскольку могут проверять очередность полученных пакетов и даже определять тип приложения, отправившего пакет, при помощи идентификатора порта. В большинстве коммутаторов для обеспечения быстрых алгоритмов обработки применяются аппаратная логика или специализированные интегральные схемы. Коммутаторы не располагают такой гибкостью программирования и настройки, как у маршрутизаторов, но их легче устанавливать и администрировать, а, следовательно, сетевому администратору требуется меньше времени на обучение. Кроме того, в расчете на стоимость порта, коммутаторы дешевле маршрутизаторов.

При создании или обновлении локальной сети выполняется поэтапная реализация плана сети:

- замена имеющиеся концентраторов на коммутаторы;
- замена устаревших коммутаторов, несовместимых с протоколом SNMP, на новые модели, в которых эта совместимость присутствует;
- подключение быстродействующих рабочих станций к коммутаторам;
- подключение серверов к высокоскоростным коммутируемым портам;
- подключение сгруппированных коммутаторов или коммутаторов рабочих групп к быстродействующим коммутаторам с использованием высокоскоростных каналов;
- подключение основных сегментов подразделения или быстродействующих коммутаторов к маршрутизаторам с использованием по мере надобности высокоскоростные каналы.

Важным вопросом при проектировании локальных сетей является ***размещение хостов и серверов сети***.

Хосты и серверы можно располагать в сети централизованно или в разных точках сети. Группа серверов (хостов) обычно располагается в машинном зале с контролируемой средой, т.е. этот зал имеет также специальное оборудование для

фильтрации колебаний силового напряжения и поддержания температуры в заданном диапазоне. Кроме того, в зале устанавливаются ИБП и системы архивации, а сам зал закрывается от постороннего доступа.

Планирование поддержания постоянной температуры в помещениях с серверами обеспечит надежную запись информации на магнитные диски и стабильную работу всего компьютерного оборудования.

Создание группы серверов (хостов) позволяет сэкономить средства, поскольку некоторое оборудование может обслуживать целое помещение, и его не нужно покупать отдельно для каждого хоста и сервера.

Централизованное размещение серверов позволяет сэкономить средства на управлении и ресурсах. Размещение серверов в соответствии со структурой подразделений организации позволяет при эксплуатации ресурсов учитывать специфику конкретного подразделения.

Каналы, связывающие серверы и сетевое оборудование, должны быть высокоскоростными, и их следует изолировать от тех сегментов, в которых располагаются рабочие станции. Наличие скоростных каналов обеспечит полосу пропускания, достаточную для всех пользователей, обращающихся к серверам. Изолируя серверы от других сегментов, можно также обеспечить избыточность, и соответственно, живучесть сети. Высококритичные хосты и серверы могут напрямую подключаться к отдельным дуплексным интерфейсам коммутаторов.

В высокоскоростной канал к серверу эффективен только тогда, когда и сетевой адаптер, и шина, и процессор этого сервера могут работать с достаточно высокой скоростью. В противном случае сервер окажется узким местом в сети.

Избыточность можно также создать, установив на серверы две и более сетевых адаптеров, благодаря чему при отказе одного адаптера сервер сможет взаимодействовать с сетью через другой адаптер.

Создание избыточности в сети осуществляется также за счет добавления резервного коммутатора между хостами (серверами) и теми компьютерами, которые к ним обращаются. При выходе главного коммутатора из строя резервный коммутатор подключается либо сетевым администратором, либо автоматически.

При распределенном по сети размещении хостов и серверов их также необходимо непосредственно подключать к коммутаторам с помощью высокоскоростных каналов. Преимуществами такого подхода являются:

- менее концентрированный трафик, чем при создании группы серверов;
- гибкость сетевой структуры в случае отказа одного из серверов;
- возможность установки резервных хостов на разных площадках (на случай отказа одного из хостов или его недоступности из-за сетевых проблем).

Недостатком рассредоточенного размещения хостов и серверов является потребность в дополнительном оборудовании в каждой точке установки.

В заключение вопроса затронем вопросы **проектирования беспроводных локальных сетей**. В беспроводных локальных сетях используются две основные топологии: одноранговая (peer-to-peer) и многоячеечная (multiple-cell). Одноранговая архитектура присуща небольшим сетям с числом пользователей, не превышающим 20–25. При использовании устройств стандарта 802.11a все взаимодействующие друг с другом станции должны находиться в радиусе 18 м, в случае применения устройств стандарта 802.11b это расстояние увеличивается до 90 м.

В одноранговой топологии сети стандарта 802.11b фактически имеется только одна ячейка, внутри которой вещают все станции. При этом отсутствует точка доступа, соединенная с кабельной локальной сетью.

В многоячеечной структуре используются *точки доступа* (например, беспроводные мосты, подключенные к кабельной локальной сети). Вокруг каждой точки доступа образуется ячейка. Если имеются четыре точки доступа, то существуют четыре ячейки. Некоторые станции (например, настольные системы) могут располагаться в ячейке неподвижно. Другие станции (такие, как портативные компьютеры) могут перемещаться от одной ячейки к другой.

Ячейки сконфигурированы так, что можно обеспечить роуминг с использованием какого-нибудь соответствующего протокола (например, Inter-Access Point Protocol, IAPP). Широковещательная область вокруг каждой ячейки имеет радиус 18 м для устройств стандарта 802.11a и 90 м для устройств стандарта 802.11b. Чтобы не разрешать роуминг, необходимо настроить по-разному каждую точку доступа и связанные с ней рабочие станции. При этом некоторая станция сможет взаимодействовать только с определенной точкой доступа.

### **3.2.4. Принципы проектирования глобальных сетей**

Обычно проектирование ГВС включает в себя этап подключения локальной сети или некоторого узла к глобальной сети. Для подключения локального узла к глобальной сети используются следующие устройства:

- маршрутизаторы;
- серверы доступа;
- модемы;
- специализированные адаптеры;
- мультиплексоры;
- беспроводные и спутниковые устройства;
- коммутаторы доступа к глобальным АТМ-сетям.

Распространенным и простым способом подключения локального узла к ГВС является использование модема, терминального адаптера, адаптера DSL или X.25, установленного на сервере. Более широкий спектр возможностей подключения дает установка сервера доступа с несколькими перечисленными устройствами.



Для оптимального управления сетевым трафиком и для реализации функции межсетевого экранирования сервер или сервер доступа необходимо подключать *непосредственно к отдельному интерфейсу маршрутизатора*, т.е. *нельзя использовать те же интерфейсы*, с которыми связаны коммутаторы и сегменты рабочих групп (рис.3.7). При такой организации сети сервер или сервер доступа оказывается ближе к сетевой магистрали, в результате чего уменьшается количество ретрансляций несущего сигнала через некоторое сетевое устройство.

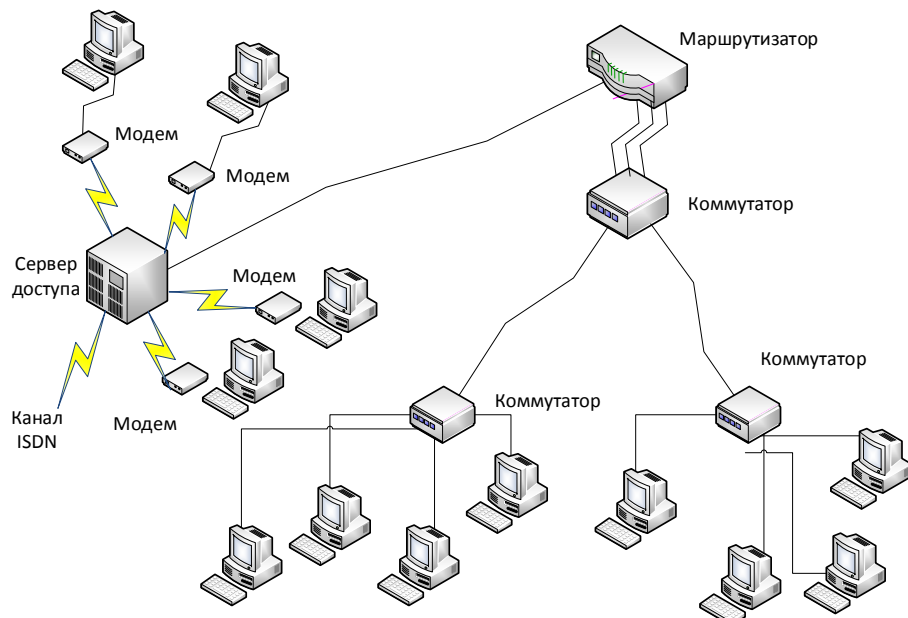


Рис. 3.7. Пример конфигурирования топологии локальной сети для осуществления глобальных коммуникаций

Другим способом подключения локальной сети к глобальной является применение маршрутизатора или устройств обслуживания канала / обработки данных (CSU/DSU). Маршрутизатор позволяет контролировать трафик глобальной сети и выполняет функции брандмауэра для входящего и исходящего трафика.

Структура затрат на глобальную сеть. В отличие от расходов на локальную сеть, которые могут значительно изменяться в зависимости от стоимости обновлений и новых программ, расходы на услуги глобальной сети остаются относительно стабильными, поскольку определяются ежемесячными выплатами. Однако, фактические затраты на развертывание и эксплуатацию глобальной сети определить сложно, поскольку имеются как прямые (ежемесячная плата за услугу), так и косвенные (обучение и поддержка пользователей) расходы.

При анализе расходов на ГВС необходимо учитывать следующие факторы:

- ежемесячная плата за услуги;
- стоимость оборудования для подключения локальной сети к глобальной;
- стоимость обучения и поддержки пользователей;
- стоимость обучения сетевого персонала;
- расходы на поддержку сети и устранение неисправностей;



- потери рабочего времени при разрыве соединения;
- затраты на периодическую модернизацию оборудования.

Крупными поставщиками услуг ГВС обычно предлагается *соглашение об уровне сервиса* (SLA, service level agreement), представляющее собой некоторое соглашение между поставщиком и клиентом, гарантирующее минимальный уровень качества обслуживания. В SLA оговариваются следующие моменты:

- гарантированное время готовности (например, доступность в течение 95% всего времени);
- максимальная задержка на линии;
- гарантированное или среднее время восстановления неисправной линии
- гарантированная пропускная способность;
- величина издержек для возмещения клиенту поставщиком в случае нарушения соглашения об уровне сервиса.

### **3.3. Порядок разработки и оформления результатов проектирования сетевой инфраструктуры**

#### **3.3.1. Общие требования стандартов по проектированию сетевой инфраструктуры**

Для проектирования локальных сетей (СКС) существует ряд международных и национальных стандартов. Наиболее известными являются:

- международный стандарт ISO/IEC IS 11801-2002 «Information Technology. Generic cabling for customer premises»
- европейский стандарт CENELEC EN 50173 «Information Technology. Generic cabling systems».
- американский стандарт TIA/EIA-568B «Commercial Building Telecommunications Wiring Standard».

Основными отечественными стандартами по проектированию компьютерных сетей являются:

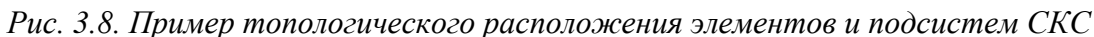
- ГОСТ 53246-2008 «Информационные технологии. Системы кабельные структурированные. Проектирование основных узлов системы. Общие требования».
- ГОСТ 53245-2008 «Информационные технологии. Системы кабельные структурированные. Монтаж основных узлов системы. Методы испытания».

**Структурированная кабельная система** (СКС) в соответствии с этими стандартами состоит из следующих функциональных элементов:

- главного кросса (МС);
- кабеля магистральной подсистемы первого уровня;
- промежуточного кросса (IC);
- кабеля магистральной подсистемы второго уровня;

- Перечисленные выше функциональные элементы объединяются в группы, формирующие подсистемы. Схематичные модели различных функциональных элементов и их взаимодействия приведены на рис. 3.8.

- магистральной кабельной подсистемы первого уровня;
- магистральной кабельной подсистемы второго уровня;
- горизонтальной кабельной подсистемы.



106

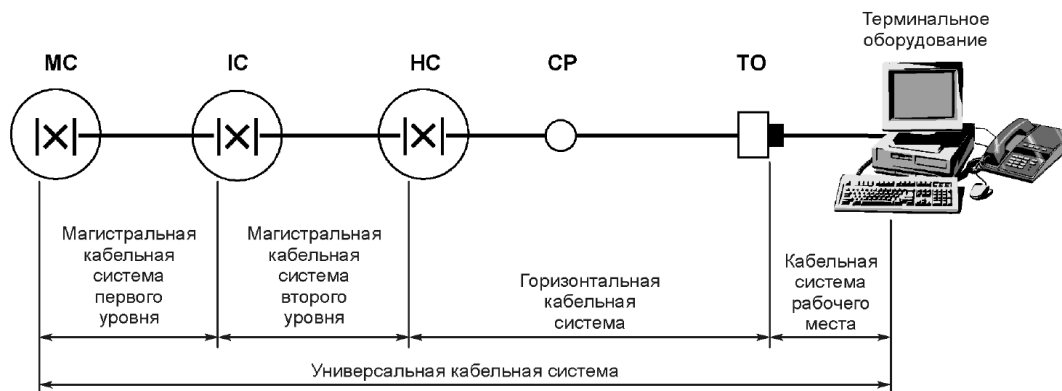


Рис. 3.9. Подсистемы СКС

При проектировании компьютерной сети в соответствии с перечисленными стандартами необходимо обратить внимание на следующие аспекты:

- проектирование горизонтальной подсистемы локальной сети;
- проектирование магистральной подсистемы локальной сети;
- планирование телекоммуникационных пространств и помещений.

Рекомендуется следующий **общий порядок разработки и оформления** проекта локальной сети.

### 1. Анализ требований Заказчика к проектируемой сети

До начала разработки проектной документации Заказчику необходимо совместно с Разработчиком сформулировать оптимальные требования и обсудить все важные аспекты создаваемых систем. Необходимо выявить и проанализировать требования и потребности Заказчика к системе передачи данных, установить, какие функции и задачи должна выполнять компьютерная сеть, выбрать топологию построения сети и сетевое оборудование, рассчитать стоимость построения сети. С целью создания долговечной системы на этапе проектирования СКС в подсистемы закладывают резерв по пропускной способности и количеству линий. Цена такого подхода, безусловно, выше, однако избыточность системы обеспечивает снижение эксплуатационных издержек. Следует учитывать, что в пределах срока службы СКС может смениться три-четыре поколения компьютеров и сетевых устройств.

### 2. Разработка технического задания.

Техническое задание включает требования Заказчика к числу центральных узлов и рабочих мест, их расположению, категории или классу системы. В большинстве случаев требуется комплексное решение – компьютерная и телефонная сеть плюс электропитание должны быть у каждого пользователя.

### 3. Выбор оборудования и его технико-экономическое обоснование.

На данном этапе проектирования производится подбор необходимого оборудования, для обеспечения максимально эффективного и надежного функционирования СКС с учетом оптимального соотношения цена – качество, а также разрабатывается техническое обоснование предложенного решения.

#### **4. Схематическое построение структурированной кабельной системы**

На этом этапе осуществляется фактический аудит существующих помещений Заказчика, с целью определения мест прокладки основных кабельных трасс, мест расположения центральных узлов, расположения серверов и рабочих мест сотрудников организации. Итогом этого этапа является схема построения СКС, служащая основой для разработки технорабочего проекта.

#### **5. Разработка технорабочего проекта компьютерной сети**

Технорабочий проект – это детально проработанный документ, раскрывающий все аспекты построения системы. Проект включает:

- 1) *пояснительную записку*, в которой изложены принципы и особенности проектируемой сети, поэтажные планы с точным пространственным расположением каждого элемента системы на архитектурных чертежах здания;
- 2) *структурные схемы СКС*, отражающие количественные параметры подсистем СКС: особенности, конфигурацию, число рабочих мест, число кабелей в горизонтальной и магистральной подсистемах, тип монтажных конструктивов и панелей;
- 3) *таблицы соединений* с перечнем всех элементов инфраструктуры, их назначение, привязку к помещениям, портам, кабельным трассам;
- 4) *схемы кабельных проводок*, расположения элементов телекоммуникационной инфраструктуры;
- 5) *схемы размещения панелей* в телекоммуникационных шкафах / стойках;
- 6) *схемы подключений кабелей* на панелях / кроссах;
- 7) *схемы организации рабочих мест*;
- 8) *спецификацию материалов*, включает точный перечень требуемых для реализации проекта материалов.

Итогом этапа технорабочего проектирования является создание функционально полного комплекта документации, предоставляющего исчерпывающую информацию для проведения монтажных работ.

#### **6. Разработка сметной документации и согласование её с клиентом.**

Заказчику предоставляются на согласование сметная документация, а также этапы и сроки проведения монтажных и пуско-наладочных работ.

#### **3.3.2. Проектирование горизонтальной подсистемы локальной сети**

Горизонтальная кабельная подсистема (ГКП) является частью СКС и соединяет телекоммуникационную розетку на рабочем месте с горизонтальным кроссом, расположенным в телекоммуникационной.

*В горизонтальную кабельную подсистему* входят:

- фиксированные кабельные сегменты;
- телекоммуникационные розетки на рабочих местах;

- коммутационное оборудование в горизонтальном кроссе, коммутационные кабели (шнуры);
- кроссировочные перемычки в телекоммуникационной;
- многопользовательские розетки (MuTOA) и консолидационные точки (CP) как дополнительный элемент.

При проектировании ГКП рекомендуется учитывать возможность работы в ней телекоммуникационных приложений следующих основных видов:

- телекоммуникационные системы передачи речи;
- коммутационное оборудование зданий;
- цифровые системы связи;
- локальные вычислительные сети;
- видеосистемы;
- сигнальные системы зданий (системы автоматизации зданий, системы безопасности, противопожарные системы и т.п.).

Горизонтальная кабельная подсистема должна планироваться с целью снижения расходов на ее обслуживание и внесение изменений, а также с учетом возможного расширения парка активного оборудования и появления новых сервисов. Для горизонтальной кабельной подсистемы определена физическая топология типа «звезда» (рис. 3.10). При необходимости реализации других сетевых топологий, таких как «шина», «кольцо» или «дерево», могут быть эффективно использованы кросс-соединения в горизонтальном кроссе.

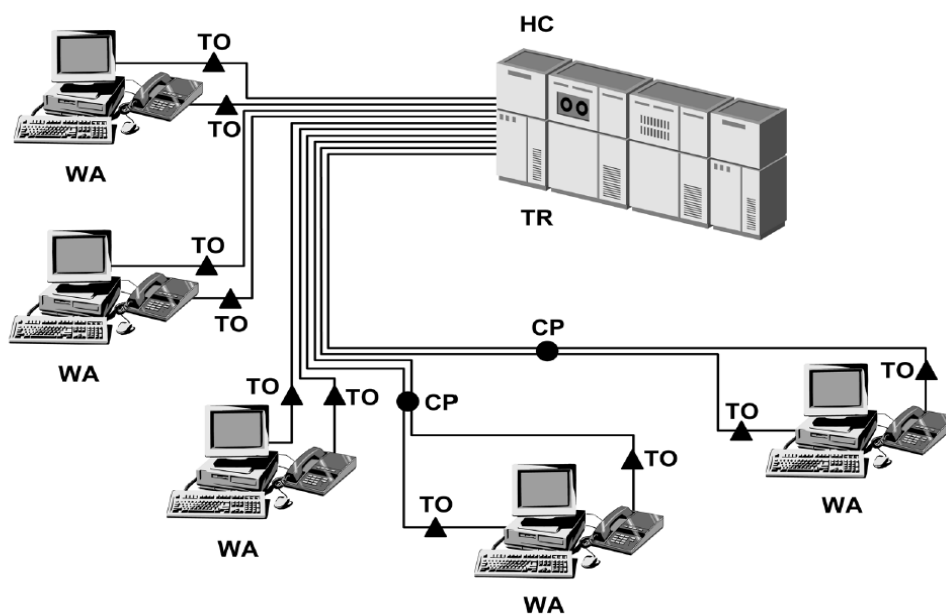


Рис. 3.10. Топология типа «звезда» горизонтальной кабельной подсистемы

Все телекоммуникационные розетки на рабочих местах должны быть соединены с горизонтальным кроссом в телекоммуникационной с помощью кабеля. Рекомендуется, чтобы телекоммуникационная была расположена на одном

этаже с обслуживаемыми ею рабочими местами. Рабочие места должны обслуживаться горизонтальным кроссом, расположенным в телекоммуникационной на том же или на смежном с ними этаже.

В ГКП на основе витой пары проводников (UTP/FTP/ScTP/SFTP) *в модели постоянной линии* (рис.3.11) допускается наличие не более трех точек коммутации (трех коннекторов):

- 1 – коннектор второй единицы коммутационного оборудования в горизонтальном кроссе (НС);
- 2 – коннектор консолидационной точки (СР);
- 3 – коннектор телекоммуникационной или многопользовательской розетки.

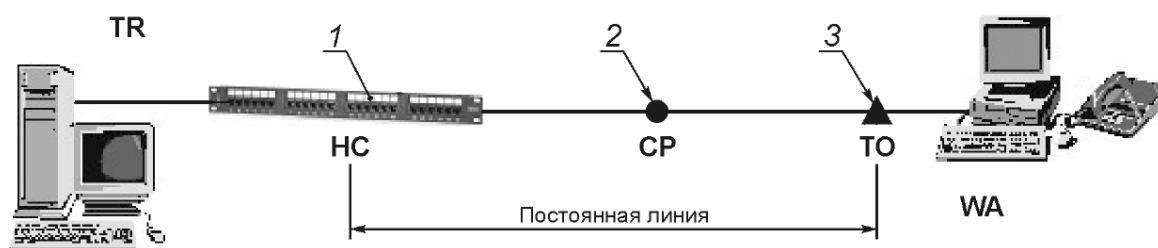


Рис.3.11. Модель постоянной линии ГКП с тремя точками коммутации

В ГКП на основе витой пары проводников (UTP/FTP/ScTP/SFTP) *в модели канала* (рис.3.12) допускается наличие не более четырех точек коммутации (четырёх коннекторов):

- 1 – коннектор телекоммуникационной или многопользовательской розетки (ТО или МуТОА);
- 2 – коннектор консолидационной точки (СР);
- 3 – коннектор первой единицы коммутационного оборудования в горизонтальном кроссе (НС);
- 4 – коннектор второй единицы коммутационного оборудования в горизонтальном кроссе (НС).

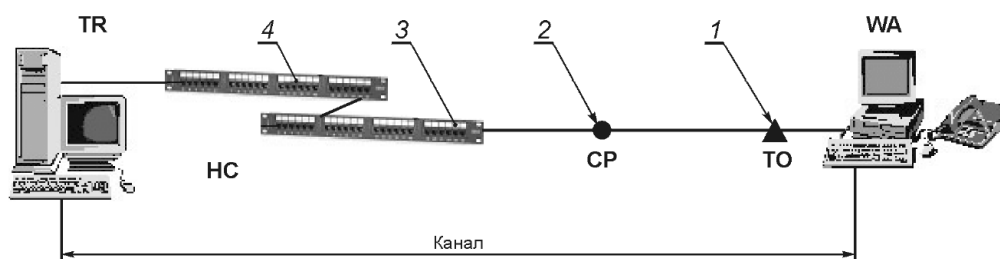


Рис.3.12. Модель канала ГКП с четырьмя точками коммутации

В горизонтальном кроссе используются два метода подключения активного оборудования к горизонтальной кабельной подсистеме и один метод для пассивной коммутации между собой горизонтальной и магистральной подсистем:

– **кросс-соединение** (метод коммутации, в котором для подключения активного оборудования к горизонтальной кабельной подсистеме или пассивной коммутации кабельных сегментов горизонтальной и магистральной подсистем используются две единицы коммутационного оборудования, соединяемые коммутационными шнурами).

– **межсоединение** (метод коммутации, в котором для подключения активного оборудования к горизонтальной кабельной подсистеме используется одна единица коммутационного оборудования, соединенная непосредственно с кабелем горизонтальной подсистемы).

В горизонтальном кроссе для подключения активного оборудования с многопортовыми коннекторами к горизонтальной кабельной подсистеме и для пассивной коммутации между собой кабельных сегментов горизонтальной и магистральной подсистем должен применяться *метод кросс-соединения*.

В горизонтальном кроссе запрещено применение метода межсоединения для пассивной коммутации между собой кабельных сегментов горизонтальной и магистральной подсистем, за исключением случаев использования топологии СОА (Centralized Optical Architecture – Централизованная Оптическая Архитектура). При пассивной коммутации между собой кабельных сегментов горизонтальной и магистральной подсистем методом межсоединения возникают неразрешимые проблемы при необходимости изменения конфигурации подключения сегментов к различным коммутационным полям.

### **3.3.3. Проектирование магистральной подсистемы локальной сети**

Магистральная кабельная подсистема имеет топологию типа «звезда» с двумя уровнями подчинения (рис.3.13). Каждый промежуточный кросс соединен с главным кроссом МС внешними магистральными сегментами. Все линии сходятся к единому центру МС, образуя тип «звезда», — первый уровень подчинения. В свою очередь, каждый горизонтальный кросс соединен с промежуточным кроссом внутренними магистральными сегментами, также образуя тип «звезда» с единым центром в МС, — второй уровень подчинения.

Кабельная подсистема СКС, задача которой – соединение главного распределительного пункта (ГРП) с промежуточными распределительными пунктами (ПРП) и этажными распределительными пунктами (ЭРП), называется **магистральной подсистемой 1-го уровня**.

В магистральную подсистему первого уровня входят магистральные кабели, проложенные от ГРП к распределительным пунктам, а также распределительные устройства, которые используются для заделки этих кабелей. Также в состав магистральной подсистемы 1-го уровня входят все коммутационные шнуры и перемычки, используемые в ГРП.

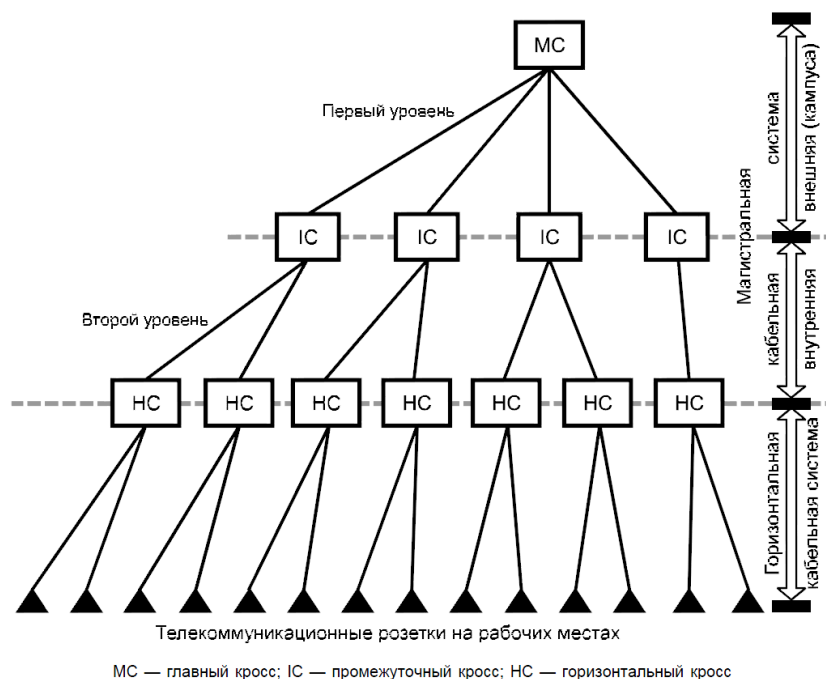


Рис. 3.13. Топология магистральной кабельной подсистемы

**Примечание:** Магистраль первого уровня (по ГОСТ Р 53246-2008) в международном стандарте ISO/IEC 11801 называется магистралью комплекса, а американском стандарте TIA/EIA-568 – магистралью между зданиями.

**В промежуточном распределительном пункте происходит разделение СКС на магистральные подсистемы 1-го уровня и 2-го уровня.**

В состав магистральной подсистемы 2-го уровня входят *кабели*, соединяющие промежуточные и этажные распределительные пункты, *распределительные устройства*, используемые для их заделки, *коммутационные шнуры* и *перемычки* ПРП, то есть их длина должна быть учтена в общей длине канала.

Стандарты отводят существенную роль регламентированию допустимой длины магистралей каждого уровня. При проектировании сети необходимо учитывать не только требования стандартов, но и рекомендации производителей телекоммуникационного оборудования. В первую очередь данный показатель зависит от избранной среды передачи данных.

Для витой пары, используемой для прокладки локальной сети, *предельно допустимая дистанция* составляет 100 метров для любой подсистемы. Если при проектировании сети возникает необходимость в прокладке более длинных магистралей, то придется обратить внимание на оптоволоконное оборудование, которое устойчиво работает на существенно больших расстояниях.

Стандарт ISO/IEC 1801 позволяет включать в подсистемы внешних магистралей (то есть магистралей, соединяющих отдельные здания) кабели длиной до 1500 метров. При этом длина магистрального кабеля от кроссовой внешней магистрали до этажной кроссовой не должна превышать 2000 метров, из которых 1500 метров относятся к внешней магистрали, а 500 метров – к внутренней. Коммутационные



и оконечные шнуры должны в этом случае иметь стандартную длину.

Вышеуказанные требования относятся к случаям, когда используется *многомодовый оптоволоконный кабель*. Использование *одномодового кабеля* позволяет увеличить длину магистральных кабелей до 3000 метров, 2500 метров из которых относятся к внешней магистрали и 500 метров – к внутренней. В американском стандарте TIA/EIA-568-B прописана длина одномодового кабеля 2700 метров и многомодового – 1700 метров.

В российском ГОСТ Р 53246-2008 для *многомодового оптоволоконного кабеля* установлена максимальная длина между главным кроссом (МС) и горизонтальным кроссом (НС) в 2000 метров. При этом для магистрали второго уровня предельная длина определена в 300 метров. Для одномодового кабеля предельно допустимая длина ограничена пятью километрами, из которых 300 метров приходится на магистраль между промежуточным кроссом (ИС) и НС.

Как видно, различия между этими стандартами весьма значительные, при этом не исключена ситуация, в которой производитель телекоммуникационного оборудования ориентируется на международный стандарт, в то время как заказчик – на ГОСТ. Можно предположить, что инсталлятору в подобной ситуации придется отдавать предпочтение наименьшему значению спорного параметра.

В ходе проектирования сети необходимо учитывать, что при измерении длины магистралей используется модель канала. То есть аппаратные и коммутационные шнуры входят в максимально допустимую длину магистрали. Кроме того, в магистраль первого уровня входят и кабели, соединяющие городской ввод с главным кроссом. В целом вне зависимости от используемой системы стандартов при проектировании сети длина магистрали внешнего уровня не должна превышать 5 километров для одномодовых систем и 2 километров – для многомодовых.

**Если расстояния, на которые нужно обеспечить связь, превышают указанные значения, стандарты рекомендуют воспользоваться телекоммуникационными каналами операторов связи, предоставляющих доступ к WAN.**

Передача данных между промежуточными и этажными распределительными пунктами внутри одного здания обеспечивается за счет системы внутренних магистралей (магистралей второго уровня). Чаще всего технические помещения располагаются на разных этажах здания. Поэтому подсистема внутренних магистралей здания нередко называется *вертикальной*.

### **3.3.4. Планирование телекоммуникационных пространств и помещений в проекте компьютерной сети**

#### **Проектирование подсистемы рабочего места**

В помещениях должно быть достаточное количество телекоммуникационных розеток, а для подключения оборудования должно хватать аппаратных шнуров

стандартной длины. На этом этапе разрабатываются и утверждаются планы расположения розеток системы, силовых и информационных, определяются тип и количество адаптеров, аппаратных шнуров, переходников и иного телекоммуникационного оборудования. Места установки силовых и информационных розеток наносятся на поэтажный план здания или заносятся в таблицу, которая будет в дальнейшем использоваться в качестве основного документа при проектировании *горизонтальной подсистемы*.

Рекомендуется устанавливать силовые и телекоммуникационные розетки на рабочем месте рядом, на расстоянии не более метра на одной высоте.

Выбирая места установки розеток при проектировании сети, следует руководствоваться, в том числе, нормами СНиП 2.09.04-87, согласно которым на одно рабочее место должно приходиться не менее 4 квадратных метров площади. В ГОСТ Р 53246-2008 среднее значение площади рабочего места определено в 10 квадратных метров. Рабочие места должны быть равномерно распределены по площади помещения. При выборе мест для розеток необходимо учесть возможность прокладки кабеля и монтажа розеток в данном месте.

Если в перспективе планируется расширение сети, то рекомендуется еще на этапе проектирования сети заложить некоторую избыточность в количество кабелей горизонтальной подсистемы, проложенных к рабочим местам.

*Хотя при проектировании сети допускается использование розеток с разной разводкой или оптоволоконных розеток разного типа, это не рекомендуется существующими стандартами.*

Соединительные шнуры в ходе проектирования сети учитываются в зависимости от количества оборудования, подключаемого к сети после ее ввода в эксплуатацию. Резерв сетевых соединительных шнуров для последующего расширения системы может составлять от 10 и более процентов от числа существующих.

*Как правило, для помещений небольших размеров используются соединительные шнуры длиной 2-3 метра, большие размеры помещения могут потребовать использование кабелей длиной до 5 метров. Использование соединительных шнуров длиной более 5 метров действующими стандартами, в частности, пунктом 5.1.1 российского национального стандарта, **запрещено**.*

*Не рекомендуется использовать самодельные или самостоятельно модернизированные (например, удлиненные) соединительные шнуры – это может отрицательно сказаться на производительности и стабильности системы. При проектировании сети лучше предусмотреть использование аппаратных кабелей, терминированных в заводских условиях. При необходимости подключения к розеткам кабельной системы нестандартного активного оборудования все переходники и адаптеры для него должны быть заранее предусмотрены и указаны в спецификации рабочего места.*

## Проектирование телекоммуникационных помещений

Проектирование телекоммуникационных помещений осуществляется на основании следующих исходных данных:

- **особенности объекта**, для которого осуществляется проектирование сети, в том числе архитектурные особенности, количество этажей, которые затрагивает прокладка сети, планировка помещений и их предназначение.

- **требования стандартов и нормативов**, в том числе строительных норм и правил, правил пожарной безопасности, ГОСТов по СКС и иных, затрагивающих телекоммуникационные помещения.

- **требования Заказчика.**

**Телекоммуникационные помещения** предназначены для размещения устройств и оборудования, которое обеспечивает работоспособность сети. Размещать в этом помещении другие виды оборудования, а тем более использовать его в качестве склада категорически не рекомендуется.

Стандарты выделяют два основных типа телекоммуникационных помещений: это **помещения для кроссов** и иного телекоммуникационного оборудования и **аппаратные**. Следует отметить, что отечественный ГОСТ Р 53246-2008 выделяет два вида телекоммуникационных помещений – собственно *телекоммуникационную*, которая предназначена для размещения главного и промежуточных кроссов, и *аппаратную*. В литературе нередко встречается иное название телекоммуникационной – кроссовая.

**Главное предназначение телекоммуникационной** – размещение распределительных пунктов и устройств. Также телекоммуникационная может быть использована для размещения активного телекоммуникационного оборудования. При проектировании сети телекоммуникационное помещение рекомендуется размещать в центре обслуживаемой сети. Минимальное количество телекоммуникационных помещений на этаже здания – одно. Допускается обслуживание телекоммуникационным помещением соседних этажей, но при условии, что стационарная кабельная линия будет иметь длину не более 90 метров (на основе витой пары). Для оптоволоконных допустимы намного большие расстояния.

Площадь телекоммуникационного помещения зависит от размещаемого в нем оборудования и размера обслуживаемой рабочей зоны. Минимальная площадь телекоммуникационного помещения должна составлять 6,6 квадратных метра, минимальная высота – 2,44 метра.

По стандарту ANSI/TIA/EIA-569A, размер телекоммуникационного помещения находится в следующей зависимости от площади обслуживаемой рабочей зоны: для площади рабочей зоны до 500 квадратных метров размер кроссового помещения должен составлять 3 x 2,2 метра, для площади 500-800 метров - 3 x 2,8 метра, для площади 800-1000 метров – 3 x 3,4 метра. При проектировании

сети стоит учитывать, что рекомендации верны при условии, что на 10 квадратных метров рабочей зоны устанавливается три телекоммуникационных модуля. Эти же требования предъявляет к телекоммуникационным помещениям и ГОСТ Р 53246-2008.

**Телекоммуникационное помещение**, в котором, помимо распределительных устройств, размещается большое количество активного телекоммуникационного оборудования, называется **аппаратной**.

В аппаратной устанавливаются распределительные устройства и пункты. Аппаратную следует размещать в непосредственной близости от магистральных кабелей. По возможности аппаратную размещают как можно ближе к ГРП или в одном помещении. При размещении аппаратной важно предусмотреть возможность дальнейшего увеличения размера ее площади. Поэтому при проектировании сети не стоит размещать аппаратную у лестничных пролетов, шахт лифтов, вентиляционных камер и т.д.

Площадь аппаратной зависит от количества установленного в ней оборудования и размера рабочей площади, которую будет обслуживать данная аппаратная. Должен быть обеспечен свободный доступ к оборудованию для проведения технического обслуживания. Также нужно выделить место под установку различных дополнительных устройств. Минимальная площадь, которую необходимо выделить для аппаратной – 14 квадратных метров.

*Зависимость площади аппаратной от обслуживаемой рабочей области, которую рекомендуется соблюдать при проектировании сети, составляет 0,09 метра на каждые 10 квадратных метров обслуживаемого помещения.*

Если, несмотря на большую площадь, количество телекоммуникационных розеток в здании невелико, площадь аппаратной вычисляют, исходя из количества рабочих зон. Если количество рабочих зон менее 100, то площадь аппаратной минимальна, то есть составляет 14 метров квадратных. Для количества рабочих зон от 101 до 400 на аппаратную выделяют 37 квадратных метров. При количестве рабочих зон от 401 до 800 площадь аппаратной должна составлять 74 квадратных метра, а при количестве рабочих зон от 801 до 1200 площадь аппаратной составит 111 квадратных метров.

При грамотном проектировании сети в аппаратной обязательно устанавливаются **системы вентиляции и кондиционирования**, а также аварийного освещения. Кроме того, аппаратная должна быть оборудована пожарно-охранной сигнализацией и средствами пожаротушения.

### **Проектирование кабельного ввода в здание**

Для **ввода в здание** внешних кабелей, магистрали СКС первого уровня и кабелей операторов связи, их заделки, а также размещения распределительных

устройств, используются помещения и пространства, которые получили название помещения кабельного ввода в здание. При проектировании сети решение о том, каким образом будет реализована точка ввода, принимается с учетом архитектурных особенностей здания, габаритов и типа оборудования, которое планируется задействовать, а также требований заказчика. Так же, как и в случае с иными телекоммуникационными помещениями, стандарты категорически не рекомендуют при проектировании сети размещение в помещении кабельного ввода устройств и оборудования, не относящихся к его прямому функциональному назначению.

*Отдельные помещения кабельного ввода в здания целесообразно выделять в том случае, если общая площадь здания превышает 2000 квадратных метров. В помещении кабельного ввода может быть размещен ГРП или большое количество активного оборудования. В этом случае в ходе проектирования сети к помещению кабельного ввода предъявляются такие же требования, как и к аппаратной и телекоммуникационной. Это относится как к площади помещения, так и к системам поддержания микроклимата, пожарной безопасности и резервного электроснабжения.*

Помещение кабельного ввода должно быть расположено таким образом, чтобы расстояния до наружной точки ввода и до помещения, в котором установлен главный заземляющий зажим или главная шина заземления, были минимальными. Таким образом, сокращается длина заземляющего проводника, который соединяет телекоммуникационную шину заземления с главной. Помещение кабельного ввода должно иметь хорошую систему вентиляции и не подвергаться воздействию влаги. Стандарт требует, чтобы при проектировании сети была учтена вероятность затопления помещения, и при наличии таковой предусмотреть дренаж.

При проектировании сети рекомендуется для помещения кабельного ввода выделять не менее 6,6 квадратных метров площади, высота потолков в котором не менее 2,44 метров. Если планируется установка в данном помещении активного оборудования, то требуется соответственно увеличение его площади.

Место ввода внешних телекоммуникационных кабелей в здание называется **точкой ввода**. Внешние кабели могут входить в здание через крышу, стены, перекрытия или фундамент. Если это возможно, при проектировании сети рекомендуется ограничиться одной точкой ввода для всего здания. В то же время в ГОСТ Р 53246-2008 упомянута возможность проектирования сети с двумя точками ввода, расположенными на удалении друг от друга (например, для обеспечения требований к бесперебойному предоставлению телекоммуникационного сервиса или для разделения каналов связи в соответствии с требованиями безопасности).

Кабели могут быть подведены к зданию воздушным или подземным способом. Во втором случае возможна как прокладка кабеля в специально отрытой траншее, так и при помощи существующей инфраструктуры, в частности канализации. Выбирая способ подвода кабелей при проектировании сети, рекомендуется учитывать такие аспекты, как перспективы расширения сети, вероятность прокладки новых кабелей в здание.

*Размещать точку ввода следует в непосредственной близости от помещения, в котором осуществляется заделка внешнего кабеля. При создании точки ввода обычно используются надежно закрепленные трубы: две основные диаметром не менее 100 мм и 3 дополнительные диаметром не менее 50 мм.*

В зданиях, где рабочая площадь не превышает 500 квадратных метров и выделение отдельного телекоммуникационного помещения не представляется возможным, используют альтернативные телекоммуникационные помещения, такие как выделенное пространство небольшой площади или телекоммуникационный шкаф. Оно заменяет собой помещение кабельного ввода, телекоммуникационную и аппаратную.

Телекоммуникационный шкаф устанавливается в рабочих или технических помещениях, в коридорах или в специально отгороженном пространстве. Требования к телекоммуникационному шкафу, которые в обязательном порядке учитываются в ходе проектирования сети: наличие запирающейся двери и хорошая влаго- и пылезащищенность. Для рабочих областей площадью до 100 метров возможно использование настенного телекоммуникационного шкафа.

Используемые в качестве альтернативных телекоммуникационных помещений выгородки или ниши должны удовлетворять ряду условий. Длина ниши, которая при проектировании сети отводится для телекоммуникационной, должна составлять не менее 2,6 метров, а ее глубина – не менее 0,6 метров. Длина выгородки должна составлять от 1,3 метров, глубина – также от 1,3 метров.

Помещения кабельного ввода должны запираться для предотвращения несанкционированного доступа к оборудованию. При использовании помещения несколькими организациями, ГОСТ Р 53246-2008 рекомендует, чтобы доступ к нему контролировал владелец здания.

## ЗАКЛЮЧЕНИЕ

Одним из важнейших направлений совершенствования АСУ Вооруженных Сил РФ является создание Единого информационного пространства (ЕИП) на основе интеграции информационных сервисов на всех уровнях управления Вооруженных Сил. Основным предназначением ЕИП ВС РФ является обеспечение информационного взаимодействия органов военного управления и удовлетворение в реальном масштабе времени их информационных потребностей. Для этого ИВС ЕИП должны обеспечивать пользователям потенциальную возможность доступа к разделяемым ресурсам всех компьютеров, объединенных в сеть ЕИП.

АСУ ВС РФ должна строиться по единому замыслу на единых системотехнических решениях, как открытая система, обеспечивающая возможность оперативного изменения ее конфигурации, исходя из решаемых задач, использования унифицированных технических средств, программного и информационного обеспечения. Практика создания АС показывает, что невозможно решить задачи создания перспективной АСУ ВС РФ с применением устаревших методов и технологий, не учитывающих требования современных руководящих документов и стандартов в области методологии и технологии проектирования АИС.

Поэтому важное значение имеет подготовка высококвалифицированных военных кадров в области создания современных автоматизированных систем, поскольку от них во многом зависят вопросы научного сопровождения разработки перспективных ВВСТ, основанных на информационных технологиях.

Использование методов моделирования и проектирования систем, освоенных курсантами в рамках данной дисциплины, является неотъемлемой частью подготовки выпускниками выпускной квалификационной работы (ВКР).

Выпускная квалификационная работа решает задачи:

- 1) подтверждения усвоения теоретических и практических знаний, полученных курсантом за весь период обучения;
- 2) углубления полученного и накопления практического опыта в области проектирования и разработки АС и их программного обеспечения;
- 3) выявления степени подготовленности выпускника по специальности к применению полученных знаний при решении конкретных задач и к самостоятельному решению инженерных задач.

Для подготовки к выполнению ВКР курсанту за время военно-научной работы и эксплуатационной практики на пятом курсе рекомендуется провести:

- анализ и систематизацию научно-технической информации по теме ВКР;
- изучение и анализ аналогов объекта проектирования (исследования) с целью модернизации или создания усовершенствованного или нового аппаратного, программного, информационного обеспечения или технологии;

- предварительное моделирование разрабатываемых систем или технологий.
- В процессе выполнения ВКР выпускник должен продемонстрировать навыки:
- самостоятельного решения инженерных задач;
- проведения исследований в области разработки теоретических вопросов аппаратного, программного и/или информационного обеспечения;
- разработки и реализации аппаратного, программного и/или информационного обеспечения в конкретной предметной области.

Выпускник должен обладать навыками проведения сопоставительного анализа на основе литературных и патентных источников, владеть методами комбинаторного и математического моделирования.

По содержательной направленности ВКР по специальности могут быть:

- научно-исследовательскими,
- проектными,
- технологическими,
- программными.

**Тематика научно-исследовательских ВКР** (дипломных работ) подразумевает проведение научно-практических исследований, выполняющихся по профилю специальности и являющихся продолжением научных исследований курсанта при работе в течение ряда лет на кафедре. Данный вид ВКР может отражать следующие направления в научно-практических исследованиях:

- 1) работы с углубленным изучением теоретических вопросов алгоритмизации и небольшим по объему ПО, раскрывающим тему проекта;
- 2) работы по построению математических моделей различного прикладного характера и имеющих в результате проектирования практически применимое аппаратное и программное обеспечение;
- 3) работы, содержащие математическое обоснование поставленной задачи и являющиеся новыми решениями специальных научных и практических задач;
- 4) работы, являющиеся расширением известных аппаратно-программных изделий, но в которых дается обоснование необходимости такого расширения и математически обосновывается выбранный алгоритм усовершенствования.

К тематике **проектных ВКР** относятся дипломные проекты, основу которых составляет проектирование АИС или управляющих систем, баз данных, АРМ различного назначения, например, следующих направлений:

- моделирование, проектирование и оптимизация АИС конкретного назначения, в том числе предназначенных для целей обучения;
- разработка систем и средств управления передачей данных в ЛВС;
- моделирование, анализ и оптимизация систем повышенной надежности;
- проектирование подсистем и задач АИС;
- разработка систем контроля работоспособности технических систем;



- проектирование и модернизация ПО микропроцессорных структур;
- разработка ПО в концепции CASE- технологии проектирования;
- разработка автоматизированных систем обучения.

Тематика **технологических ВКР** подразумевает отражение специфики технологии разработки комплекса аппаратно-программных средств или его частей, разработку информационных технологий. В технологических проектах основное внимание уделяется обоснованию и выбору методики исследований и технологий, описанию используемой технологии, предлагается технологическое решение поставленной задачи, а разработка аппаратного или программного обеспечения играет второстепенную роль. Разработка также может быть ориентирована на создание различного рода ИС, обучающих систем и «электронных» учебных пособий. Данные ВКР могут отражать следующие направления:

- разработка структуры видов обеспечения АИС;
- разработка технического, программного или информационного обеспечения АИС в выбранной CASE среде;
- разработка технологии взаимодействия оператора с техническими устройствами в системах «человек – машина»;
- разработка методов и технических средств оценки функционального состояния и рабочих параметров систем;
- обоснование специфических требований к отдельным подсистемам АИС, корректировка состава технических средств;
- определение производительности отдельных подсистем и системы в целом.

Тематика **программных ВКР** подразумевает разработку и написание практически применимых программных изделий, прикладного и/или системного программного обеспечения по специальности обучаемых. Данные дипломные проекты могут отражать следующие направления:

- разработка ОС или ее части для новых ЭВМ или новых конфигураций;
- разработка драйверов ОС для новых периферийных устройств ЭВМ;
- разработка программных систем для интеллектуальных терминалов;
- разработка ПО микропроцессорных систем;
- разработка программных средств канального, сетевого или транспортного уровня локальной сети ЭВМ.

Основное требование к тематике ВКР заключается в актуальности выбранной темы и ее практической целесообразности, а также в предоставлении возможности выпускнику продемонстрировать свою подготовку как инженера. В этом ему существенную поддержку окажут практические знания и навыки, полученные при изучении дисциплины «Моделирование и проектирование систем».

## СПИСОК ЛИТЕРАТУРЫ

1. Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ. Об информации, информационных технологиях и о защите информации.
2. ГОСТ 24.202-80 . Требования к содержанию документа «Технико-экономическое обоснование создания АСУ.
3. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
4. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированных систем.
5. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.
6. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадия создания.
7. ГОСТ 34.603-92. Виды испытаний автоматизированных систем.
8. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.
9. ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.
10. ГОСТ РВ 51987-2002. Информационная технология. Комплекс стандартов на автоматизированные системы. Требования и показатели качества функционирования информационных систем (ИС). Общие положения.
11. ГОСТ Р ИСО 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.
12. ГОСТ 29099-91. Сети вычислительные локальные. Термины и определения. – М.:ИПК Издательство стандартов, 1991.
13. ГОСТ Р 53622-2009. Информационные технологии. Информационно-вычислительные системы. Стадии и этапы жизненного цикла, виды и комплектность документов. М.:Стандартинформ, 2011. – 8 с.
14. ГОСТ 16325-88. Машины вычислительные электронные цифровые общего назначения. Общие технические требования. – М.:ИПК Издательство стандартов, 1988. – 22 с.
15. ГОСТ 28195-89. Оценка качества программных средств. Общие положения. – М.:ИПК Издательство стандартов, 1989.
16. ГОСТ 28806-90. Качество программных средств. Термины и определения. – М.:ИПК Издательство стандартов, 1990.
17. ГОСТ 15971-90. Системы обработки информации. Термины и определения. – М.:ИПК Издательство стандартов, 1990. – 12 с.
18. ГОСТ 19781-90. Обеспечение систем обработки информации программное. Термины и определения.– М.:Стандартинформ, 2010. – 14 с.
19. ГОСТ 53246-2008 «Информационные технологии. Системы кабельные

структурированные. Проектирование основных узлов системы. Общие требования». – М.: Стандартиформ, 2009. – 72 с.

20. ГОСТ 53245-2008 «Информационные технологии. Системы кабельные структурированные. Монтаж основных узлов системы. Методы испытания». – М.: Стандартиформ, 2009. – 35 с.

21. ISO/IEC/IEEE 24765:2010. Системы и программотехника. Словарь.

22. ISO/IEC IS 11801-2002 «Information Technology. Generic cabling for customer premises»

23. CENELEC EN 50173 «Information Technology. Generic cabling systems».

24. TIA/EIA-568B «Commercial Building Telecommunications Wiring Standard».

25. Алиев Т. И. Основы моделирования дискретных систем: учебное пособие – СПб.: СПбГУ ИТМО, 2009. – 363 с.

26. Анисимов В.В. Проектирование информационных систем. [Электронный ресурс: <https://sites.google.com/site/anisimovkhv/learning/pris>].

27. Буч Г., Якобсон А., Рамбо Дж., UML. Классика CS. Издание второе, - СПб.: Питер, 2006. - 736 с.

28. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – М.: Финансы и статистика, 2002. – 352 с.

29. Вишневский В.М. Теоретические основы проектирования компьютерных сетей. – М.: Техносфера, 2003. – 512 с.

30. Волкова В.Н., Денисов А.А. Теория систем и системный анализ. – М.: Юрайт, 2010. - 679с.

31. Галямина И.Г., Управление процессами, - СПб.: Питер, 2013. - 304 с.

32. Грекул В. И., Денищенко Г. Н., Коровкина Н. Л.— Проектирование информационных систем: учебное пособие / 2-е изд., испр. — М.: Интернет-Университет информационных технологий (ИНТУИТ.РУ): БИНОМ. Лаборатория знаний, 2010 .С 299.

33. Дейт К. Дж., Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом “Вильямс”, 2005. — 1328 с.: ил. — Парал. тит. англ.

34. Диго С.М., Базы данных. Проектирование и создание: Учебно-методический комплекс. – М.: Изд. центр ЕАОИ. 2008. – 171 с.

35. Додонов А.Г., Кузнецова М.Г., Горбачик Е.С. Введение в теорию живучести вычислительных систем. – Киев: Наук. думка, 1990. –184 с.

36. Дубейковский В.И., Эффективное моделирование с СА ERwin Process Modeler (BPwin; AllFusion Process Modeler), - М.: Диалог-МИФИ, 2009, - 384 с.

37. Каюмова А.В. Визуальное моделирование систем в StarUML: Учебное пособие/ А.В. Каюмова. Казань. – Казанский федеральный университет, 2013. – 104с.

38. Коцюба И.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. – СПб: Университет ИТМО, 2015. – 206 с.

39. Колосов В.Г., Мелехин В.Ф. Проектирование узлов и систем автоматики и вычислительной техники: Учебное пособие для вузов. – Л.: Энергоатомиздат. Ленинг.отд-ние, 1983. – 256 с.

40. Ларман К., Применение UML 2.0 и шаблонов проектирования. Введение в

объектно-ориентированный анализ, проектирование и итеративную разработку, - М.: Вильямс, 2013. - 736 с.

41. Максименков А.В., Селезнев М.Л. Основы проектирования информационно-вычислительных систем и сетей ЭВМ. – М.: Радио и связь, 1991. – 320 с.

42. Соколов, Н.Е. Методология и технология проектирования информационных систем / Н.Е. Соколов, В.А. Кокунов – СПб. – Изд-во «Скифия-Принт». – 2014.

43. Норенков И.П. Основы автоматизированного проектирования: Учебник для вузов. – М.: Изд-во МГТУ, 2000. – 360с.

44. Новиков Ф.А., Иванов Д.Ю., Моделирование на UML [Электронный ресурс]: Интернет книга. - Электронные данные. - 2013. - Режим доступа: <http://book.uml3.ru>.

45. Олифер В. Г., Олифер Н. А., Компьютерные сети. Принципы, технологии, протоколы. – 4-е изд. – СПб.: Питер, 2011. – 943 с.

46. Палмер М., Синклер Р.Б. Проектирование и внедрение компьютерных сетей. Учебный курс. – СПб.: БХВ-Петербург, 2004. – 752 с.

47. Платунова С.М. Методы проектирования фрагментов компьютерной сети – СПб: НИУ ИТМО, 2012. – 51 с.

48. Половко А.М., Гуров С.В. Основы теории надежности. 2-е изд. – СПб.: БХВ-Петербург, 2006. – 704 с.

49. Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф. Проектирование экономических информационных систем: Учебник. – М.: Финансы и статистика, 2001. – 512 с.

50. Столлингс В. Современные компьютерные сети, 2-е издание Энциклопедия. – СПб.: Питер, 2003. – 782 с.

51. Хетагуров Я.А. Проектирование автоматизированных систем обработки информации и управления (АСОИУ): учебник. – М. : БИНОМ. Лаборатория знаний, 2015. – 240 с.

52. Хетагуров Я.А., Древс Ю.Г. Проектирование информационно-вычислительных комплексов: Учеб. для вузов. – М.: Высшая школа, 1987. – 280 с.

53. Хорошев А.Н. Основы системного проектирования технических объектов. - М.: Изд-во МГТУ, 2011. – 372с.

54. StarUML. The Open Source UML/MDA Platform. [Электронный ресурс.: <http://staruml.sourceforge.net/en/documentations.php>] (дата обращения 12.01.2016).