

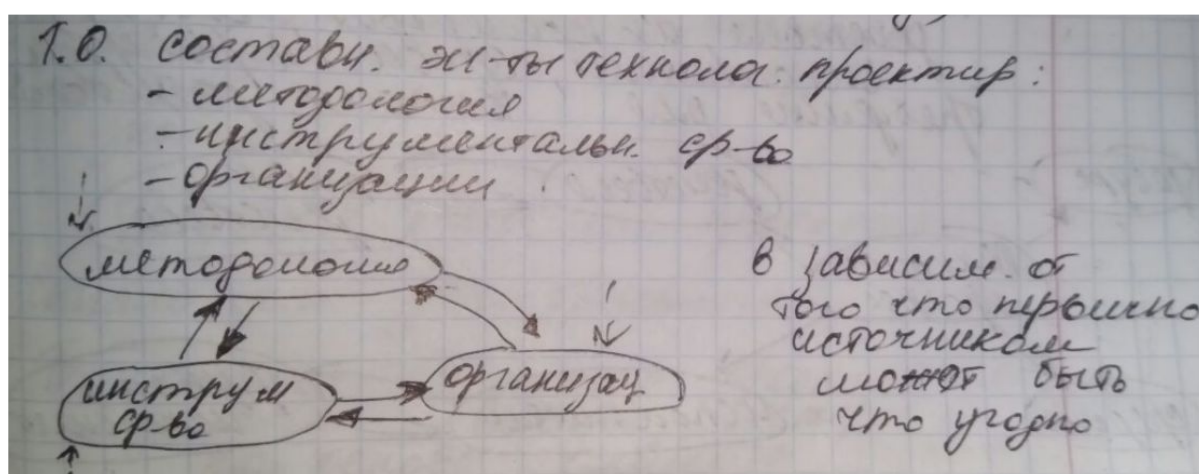
1. Понятие технологии проектирования ИС, понятие информационной модели.	3
2. Подходы к проектированию и построению ИС	3
3. Понятие метода проектирования ИС. Классификация методов проектирования ИС	4
4. НЕ ЗАКОНЧЕНО Систем автоматизированного проектирования (САПР) технических средств и ИС: сходства, отличия, особенности, примеры.	5
5. Требования к технологиям проектирования ИС	8
6. [????] Формализация описания технологии проектирования ИС	8
7. Классификация технологий проектирования ИС. Выбор технологии проектирования ИС	9
8. [НЕТУ] Классификация средств проектирования ИС	9
9. Основные стадии жизненного цикла проектирования ИС	9
10. [НЕТУ] Модели жизненного цикла ИС: основные и модифицированные	11
11. [НЕТУ] Соответствие информационных процессов, информационных технологий и средств их проектирования. Принципы проектирования базовых информационных технологий.	11
12. Классификация стандартов на проектирование и разработку информационных систем.	11
13. [НЕТУ] Международные стандарты ISO/IEC 12207:1995-08-01 (ГОСТ Р ИСО/МЭК 12207), стандарт ISO/IEC 15288:2002 (ГОСТ Р ИСО/МЭК 15288-2005), комплекс стандартов ГОСТ 34.	13
14. Фирменные стандарты (Oracle CDM, MSF и др.)	13
15. Методология моделирования функциональной структуры объектов – SADT. Методология моделирования данных – ERD (Entity-Relationship Diagrams) (case-метод Баркера).	14
16. Методология моделирования работы в реальном времени – STD.	15
17. Модель быстрой разработки приложений RAD (Rapid Application Development)	17
18. [НЕТУ] Методология функционального моделирования процессов – IDEF0. Характеристика диаграмм. Типы взаимосвязей между блоками.	17
19. [НЕТУ] Кроссплатформенная система моделирования и анализа бизнес-процессов Ramus Educational: возможности, особенности применения при проектировании ИС.	17
20. [НЕТУ] Методология функционального моделирования процессов – IDEF0. Последовательность создания функциональных моделей.	18

21. [Дописать DEF1X] Технология анализа взаимосвязей между информационными потоками – IDEF1 (IDEF1X).	18
22. [НЕТУ] Средства СА ERwin Data Modeler Community Edition: возможности, особенности применения.	19
23. Методология описания (документирования) и моделирования процессов – IDEF3.	19
24. [НЕТУ] Диаграммы PFDD (Process Flow Description Diagrams), OSTN (Object State Transition Network).	20
25. Методология рационального унифицированного процесса RUP (Rational Unified Process).	20
26. [НЕТУ] Диаграмма Последовательности (Sequence Diagram) и диаграмма Ганта: сходства, отличия, назначение.	26
27. Технологии, основанные на моделировании, анализе бизнес-процессов BPMN.	26
28. Инструмент для моделирования бизнес-процессов ARIS Express: типы диаграмм, возможности.	28
29. [НЕТУ] Реинжиниринг бизнес-процессов в ИС (реорганизации бизнес-процессов – BPMN) при проектировании ИС.	29
30. [НЕТУ] Технологии объектно-ориентированного подхода (анализа) при проектировании ИС.	29
31. [НЕТУ] Технология объектно-ориентированного подхода с использованием паттернов.	29
32. [НЕТУ] Средства имитационного моделирования ИС на основе технологии модельно-ориентированного подхода.	29
33. [НЕТУ] Основные подходы в имитационном моделировании ИС и соответствующие технологии.	29
34. [НЕТУ] Средства модельно-ориентированного подхода (анализа): MathLab Sumulink, VinSim, AnyLogic.	29
35. [НЕТУ] Инструмент для разработки и исследования имитационных моделей AnyLogic: панели, окна, редакторы.	29
36. Гибкие методологии проектирования (agile-методы).	30
37. [НЕТУ] Понятие CASE-технологии проектирования ИС. Основные принципы CASE - технологии.	31
38. [НЕТУ] Классификация CASE-средств, стратегия их выбора.	31
39. [НЕТУ] Технологии планирования проектов по стандарту PMBOK Guide – Руководству к своду знаний по управлению проектами.	31

40. Метод PERT (Program Evaluation and Review Technique) при проектировании ИС. 31
41. [НЕТУ] Средства планирования проектов: OpenProj, Microsoft Project. 32
42. [НЕТУ] Проблемы и перспективы проектирования ИС: мультизадачность ИС, гетерогенность системы и среды, неопределенность задач и требований, Runtime, On-line проектирование. 32

## 1. Понятие технологии проектирования ИС, понятие информационной модели.

**Технология проектирования информационных систем** – совокупность технологических операций в их последовательности и взаимосвязи, приводящая к разработке проекта системы, при этом определяется совокупность методов (методология) и средств, направленных не только непосредственно на проектирование ИС, но и на организацию, управление, внедрение и модернизацию проекта ИС, то есть обеспечение всего жизненного цикла системы.



Расшифровка текста: методология, инструментальные средства, организация проектирования

**Информационная модель** – Модель объекта, в которой представлены информационные аспекты моделируемого объекта. Является основой разработки моделей ИС.

## 2. Подходы к проектированию и построению ИС

Есть 2 основных подхода: Локальный и Системный.

Сущность **локального подхода** к проектированию состоит в последовательном наращивании задач, решаемых в системе. В этом случае проектирование

информационной системы состоит из решения задач, ориентированных на удовлетворение потребностей конкретных подразделений или требований, связанных с реализацией конкретных условий. При этом данные организовывают в отдельные логически структурированные файлы.

Этот подход имеет серьезные недостатки:

- избыточность информации;
- противоречивость;
- недостаточная скорость обработки;
- отсутствие гибкости;
- низкая стандартизация программного обеспечения.

**Системный подход**, будучи общей методологической базой проектирования информационных систем, основан на концепции интеграции данных, которые описывают все сферы деятельности объекта информатизации. Этот подход предусматривает рассмотрение всех элементов и составляющих процесса проектирования в их взаимосвязи, взаимозависимости и взаимном влиянии в интересах оптимального достижения как отдельных, так и общих целей создания информационной системы.

На современном этапе можно выделить четыре основных подхода к проектированию:

- **структурный подход** (функционально-ориентированное проектирование), который использует структурные методы для построения функциональной, информационной и других моделей информационной системы;
- **блочно-иерархический подход** к проектированию использует идеи декомпозиции сложных описаний объектов и соответственно средств их создания на иерархические уровни и аспекты, вводит понятие стиля проектирования (восходящее, нисходящее, смешанное), устанавливает связь между параметрами соседних иерархических уровней;
- **объектный подход** (объектно-ориентированное проектирование) предлагает набор объектных моделей для описания предметной области;
- **модельный подход** (модельно-ориентированное проектирование) основан на настройке и доработке типовой конфигурации информационной системы в среде специализированных инструментальных систем.

### 3. Понятие метода проектирования ИС. Классификация методов проектирования ИС

Классификация методов проектирования ИС:

1) По подходу к автоматизации объектов:

- сверху вниз - разработка начинается с определения целей решения проблемы, после чего идет последовательная детализация. При нисходящем проектировании задача анализируется с целью определения возможности

разбиения ее на ряд подзадач. Затем каждая из полученных подзадач также анализируется для возможного разбиения на подзадачи;

- снизу вверх - разработка подсистем (процедур, функций), в то время когда проработка общей схемы не закончилась, то есть разработка ведется от отдельных задач ко всей системе;
- принципы «дуализма» и многокомпонентности – подход к проектированию ИС заключается в сбалансированном сочетании двух предыдущих.

2) По степени автоматизации:

- вручную (на бумаге);
- автоматизированной (на бумаге + CASE-средства).

3) По степени использования типовых решений:

- типового проектирования, предполагающего конфигурацию ИС из готовых типовых проектных решений;
- оригинального (индивидуального) проектирования, когда проектные решения разрабатываются «с нуля» в соответствии с требованиями к ИС.

4) По степени адаптивности проектных решений:

- реконструкция - адаптация проектных решений выполняется путем переработки соответствующих компонентов (перепрограммирования программных модулей);
- параметризации - проектные решения настраиваются (перегенерируются) в соответствии с изменяемыми параметрами;
- реструктуризация - регенерация используемого набора проектных решений в соответствии с изменениями модели предметной области.

#### **4. НЕ ЗАКОНЧЕНО Систем автоматизированного проектирования (САПР) технических средств и ИС: сходства, отличия, особенности, примеры.**

**Организационное обеспечение автоматизированного проектирования** - совокупность документов устанавливающих состав проектной организации, межорганизационные связи, функции, формы представления результатов проектирования и порядок их согласования

**САПр** - комплекс средств автоматизации проектирования взаимосвязанных с необходимыми подразделениями проектирования и специалистами (проектировщиками)

Классификация САПр  
ГОСТ 23501.108-85

Фасетный метод классификации XXXX-XXXX

1. Тип объекта (1-8)
2. Вид
3. Ложность (1-5)
4. Уровень автоматизации (1-3)
5. Комплексность автоматизации (1-3)
6. Характер документов (1-5) обычно ставится 4 или 5
7. Количество документов (1-3)

8. Количество уровней в архитектуре технического обслуживания (1-3)

**Тип объектов**

1. Машиностроение
2. Приборостроение
3. Технологические процессы машиностроения
4. Строительство
5. Технологические процессы в строительстве
6. Программные изделия
7. Организационные системы
8. Прочие

**Вид объекта - нет ничего, у кого что-то есть напишите**

**Сложность**

1. Простые объекты (< 100 элементов)
2. Средней сложности (100-1000)
3. Сложные (1000-10000)
4. Высокой сложности(10000-1000000)
5. Сверхсложные(>1000000)

**Уровень автоматизации**

1. Низко автоматизированные <25%
2. Средне автоматизированные 25-50%
3. Высоко автоматизированные >50%

**Комплексность автоматизации**

1. Одноэтапно
2. Многоэтапно
3. Комплексно

**Количество выпускаемых документов**

1. САПр малой производительности <10000
2. САПр средней производительности 10000-1000000
3. САПр большой производительности >1000000

**Характер документов**

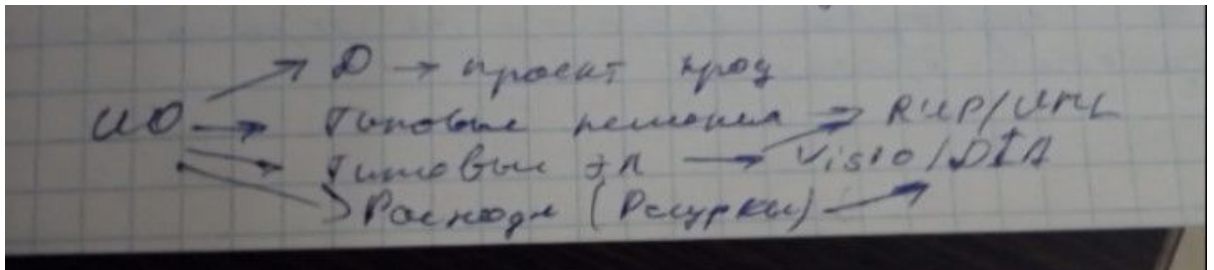
4. Комбинированные документы
5. Прочие

**Число уровней в архитектуре**

1. Одноуровневая
2. Двухуровневая
3. Трехуровневая

**Информационное обеспечение САПр**

Совокупность документов, описывающих стандартные процедуры, типовые решения, комплектующие элементы и другие ресурсы



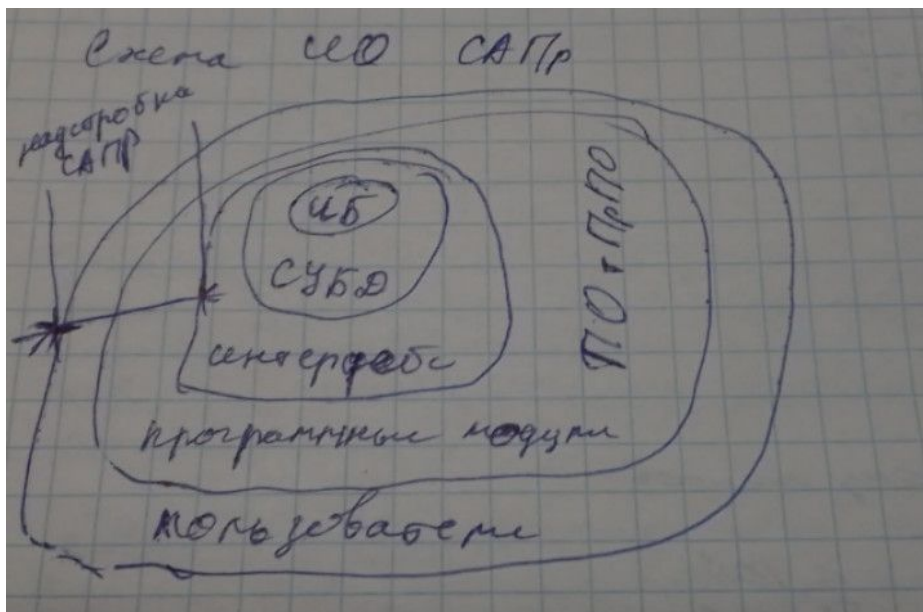
D -> процесс проектирования  
 Типовые решения -> RUP/UML  
 Типовые элементы -> Visio/DIA  
 Расходное (Ресурсы)

#### Основные требования к ИО САПр

- Наличие необходимой информации как для ручного так и для автоматизированного проектирования
- Возможности хранения и поиска информации, предоставление результатов процессов проектирования
- Объем хранимой информации и соответствующая его структура
- Быстродействие системы информационного обеспечения
- Возможность быстрого внесения изменений

При создании ИО САПр основная проблема заключается в преобразовании информации или консолидации

Типовыми группами данных являются классификаторы и таблицы соответствий для них, а также расчетно-проектная информация (класс оперативной информации)



Основной проблемой построения ИО САПр является взаимодействие системы с проектными модулями

#### Проблемы

- Согласование и сопряжение ИО САПр с модулями по форматам записей
- Согласование по программным средствам
- Согласование языков программирования

- Согласование кодов и обозначений данных

Сложность разработки БД для САПр обуславливается тем, что формирование ее структуры возможна только после разработки алгоритмов проектирования

## 5. Требования к технологиям проектирования ИС

Технология проектирования предполагает возможность выбора различных методов и средств на основе вариантного анализа.

- 1) Обеспечение создания ИС, отвечающей целям и задачам (заказчика)
- 2) Гарантированное создание системы с заданным качеством в заданное время в рамках заданных ресурсов
- 3) Поддержка сопровождения, модификации и наращивания системы
- 4) Обеспечение преемственности разработки
- 5) Обеспечение роста производительности труда проектировщика
- 6) Обеспечение надежности программного продукта и простота ведения проектной документации

Проблемы, порождаемые применением технологий проектирования на различных этапах ЖЦ(жизненного цикла)

- 1) Формирование требований к ИС  
Проблема на 1: противоречивость требований
- 2) Разработка концепции
- 3) Разработка ТЗ

Проблемы на 2 и 3: отсутствие аналогов, целесообразность привлечения специалистов, необходимость системного подхода.

- 4) Эскизный проект
- 5) Технический проект

Проблемы на 4 и 5: целесообразность унификации этапов, стандартизация описания, противоречивость этапов

- 6) Рабочая документация

Проблема на 6: стандарты, проблема коммуникации между специалистами разных уровней с целью описания отдельных структур

- 7) Ввод в эксплуатацию(действие)
- 8) Сопровождение

Проблемы на 7 и 8: целесообразность модифицирования, сокращение количества натурных испытаний

## 6. [????] Формализация описания технологии проектирования ИС

Технология проектирования в общем виде может быть формализована в следующем виде на основании теории множеств:

$$TD = \langle D \langle M^D, S^D \rangle; OD \langle M^{OD}, S^{OD} \rangle; OR \langle M^{OR}, S^{OR} \rangle \rangle$$

Где D - проектирование, OD-организация, OR- модернизация и реинжиниринг

M - возможно??? методы,

S - ??? средства



Нужна расшифровка буковок в формуле

## **7. Классификация технологий проектирования ИС. Выбор технологии проектирования ИС**

Технология проектирования информационных систем – совокупность технологических операций в их последовательности и взаимосвязи, приводящая к разработке проекта системы, при этом определяется совокупность методов (методология) и средств, направленных не только непосредственно на проектирование ИС, но и на организацию, управление, внедрение и модернизацию проекта ИС, то есть обеспечение всего жизненного цикла системы.

Среди технологий проектирования ИС выделяют два основных класса:

- 1) каноническая технология;
- 2) индустриальная технология.

Технология канонического проектирования предполагает использование инструментальных средств универсальной компьютерной поддержки и предназначена для создания индивидуальных (оригинальных) проектов локальных ИС. При этом адаптация проектных решений возможна лишь путем перепрограммирования соответствующих программных модулей.

Организация канонического проектирования ИС ориентирована на использование главным образом каскадной модели жизненного цикла ИС. Стадии и этапы работы описаны в ГОСТ 34.601-90 [18].

Технологии индустриального проектирования используют инструментальные средства специальной компьютерной поддержки для разработки проектов сложных интегрированных (корпоративных) ИС.

Индустриальная технология проектирования, в свою очередь, разбивается на два подкласса:

- типовое (параметрически-ориентированное или модельно-ориентированное) проектирование.
- автоматизированное проектирование (использование CASE-технологий).

## **8. [НЕТУ] Классификация средств проектирования ИС**

## **9. Основные стадии жизненного цикла проектирования ИС**

Понятие жизненного цикла (ЖЦ) является одним из базовых понятий методологии проектирования информационных систем.

Модель ЖЦ, как правило, представляется в виде диаграммы ЖЦ, на которой изображаются пути перехода из некоторого начального состояния в конечное

состояние и события, инициирующие изменения состояния. Модель ЖЦ может быть также представлена в виде текстового описания.

Структура ЖЦ по стандарту ГОСТ Р ИСО/МЭК 12207 базируется на трех группах процессов (каждый процесс включает ряд действий. Каждое действие включает ряд задач.):

- основные процессы ЖЦ ПО (приобретение, поставка, разработка, эксплуатация, сопровождение);
- вспомогательные процессы, обеспечивающих выполнение основных процессов (документирование, управление конфигурацией, обеспечение качества, верификация, аттестация, оценка, аудит, решение проблем);
- организационные процессы (управление проектами, создание инфраструктуры проекта, определение, оценка и улучшение самого ЖЦ, обучение).

Структура ЖЦ по стандарту ГОСТ Р ИСО/МЭК 15288-2005 базируется на пяти группах процессов:

- договорные процессы: приобретение (внутренние решения или решения внешнего поставщика); поставка (внутренние решения или решения внешнего поставщика);
- процессы предприятия: управление окружающей средой предприятия; инвестиционное управление; управление ЖЦ ИС; управление ресурсами; управление качеством;
- проектные процессы: планирование проекта; оценка проекта; контроль проекта; управление рисками; управление конфигурацией; управление информационными потоками; принятие решений;
- технические процессы, включающие: определение требований; анализ требований; разработка архитектуры; внедрение; интеграция; верификация; переход; аттестация; эксплуатация; сопровождение; утилизация;
- специальные процессы: определение и установка взаимосвязей исходя из задач и целей.

Структура ЖЦ по стандарту ГОСТ 34.601-90 предусматривает следующие стадии и этапы создания автоматизированной системы (АС) (8 групп процессов):

- **формирование требований к АС:** обследование объекта и обоснование необходимости создания АС; формирование требований пользователя к АС; оформление отчета о выполнении работ и заявки на разработку АС;
- **разработка концепции АС:** изучение объекта; проведение необходимых научно-исследовательских работ; разработка вариантов концепции АС и выбор варианта концепции АС, удовлетворяющего требованиям пользователей; оформление отчета о проделанной работе;
- **техническое задание:** разработка и утверждение технического задания на создание АС;
- **эскизный проект:** разработка предварительных проектных решений по системе и ее частям; разработка документации на АС и ее части;

- **технический проект:** разработка проектных решений по системе и ее частям; разработка документации на АС и ее части; разработка и оформление документации на поставку комплектующих изделий; разработка заданий на проектирование в смежных частях проекта;
- **рабочая документация:** разработка рабочей документации на АС и ее части; разработка и адаптация программ;
- **ввод в действие:** подготовка объекта автоматизации; подготовка персонала; комплектация АС 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;
- **сопровождение АС:** выполнение работ в соответствии с гарантийными обязательствами; послегарантийное обслуживание; эскизный, технический проекты и рабочая документация – это последовательное построение все более точных проектных решений.

Допускается исключать стадию «Эскизный проект» и отдельные этапы работ на всех стадиях, объединять стадии «Технический проект» и «Рабочая документация» в «Технорабочий проект», параллельно выполнять различные этапы и работы, включать дополнительные.

**Замечание:** стандарт ГОСТ 34.601-90 не вполне подходит для проведения разработок в настоящее время: многие процессы отражены недостаточно, а некоторые положения устарели.

## **10. [НЕТУ] Модели жизненного цикла ИС: основные и модифицированные**

## **11. [НЕТУ] Соответствие информационных процессов, информационных технологий и средств их проектирования. Принципы проектирования базовых информационных технологий.**

## **12. Классификация стандартов на проектирование и разработку информационных систем.**

Реальное применение любой технологии проектирования, разработки и сопровождения ИС в конкретной организации и конкретном проекте происходит при соблюдении всеми участниками проекта правил и соглашений, касающихся:

- проектирования;
- оформления проектной документации;
- пользовательского интерфейса.



Рисунок – Стандарты проектирования информационных систем

**Стандарт проектирования** должен устанавливать:

- набор необходимых моделей (диаграмм) на каждой стадии проектирования и степень их детализации;
- правила фиксации проектных решений на диаграммах, в том числе: правила именования объектов (включая соглашения по терминологии), набор атрибутов для всех объектов и правила их заполнения на каждой стадии, правила оформления диаграмм, включая требования к форме и размерам объектов, и т.д.;
- требования к конфигурации рабочих мест разработчиков, включая настройки операционной системы, настройки CASE-средств, общие настройки проекта и т.д.;
- механизм обеспечения совместной работы над проектом, в том числе: правила интеграции подсистем проекта, правила поддержания проекта в одинаковом для всех разработчиков состоянии (регламент обмена проектной информацией, механизм фиксации общих объектов и т.д.), правила проверки проектных решений на непротиворечивость и т.д.

**Стандарт оформления проектной документации** должен устанавливать:

- комплектность, состав и структуру документации на каждой стадии проектирования;
- требования к ее оформлению (включая требования к содержанию разделов, подразделов, пунктов, таблиц и т.д.);

- правила подготовки, рассмотрения, согласования и утверждения документации с указанием предельных сроков для каждой стадии;
- требования к настройке издательской системы, используемой в качестве встроенного средства подготовки документации;
- требования к настройке CASE-средств для обеспечения подготовки документации в соответствии с установленными требованиями.

Стандарты в области информационных технологий можно классифицировать:

- в зависимости от организации, утверждающей стандарты;
- по уровню утверждающей организации;
- по предмету стандартизации;
- по используемым методическим источникам.

В зависимости от *организации, утверждающей стандарты*:

- официальные международные, официальные национальные или национальные ведомственные стандарты (например, стандарты ISO1, IEC2, ГОСТ);
- стандарты международных консорциумов и комитетов по стандартизации (например, стандарты OMG – Object Management Group (Группа по управлению объектами));
- стандарты «де-факто» – официально никем не утвержденные, но фактически действующие (например, стандартом «де-факто» долгое время были язык взаимодействия с реляционными базами данных SQL4);
- фирменные стандарты (например, Microsoft ODBC5, методика Oracle CDM).

**Объектами стандартизации** при проектировании информационных систем являются:

- процесс (что делать?): ISO12207, ISO15288, IEEE1220;
- практика (как следует делать?): ISO15504;
- источник (какие данные использовать): ISO10303;
- описание: IDEF, UML, IEEE 1471.

Стандартизация информационных технологий и систем повышает их прибыльность за счет снижения затрат на создание и особенно модификацию.

**13. [НЕТУ] Международные стандарты ISO/IEC 12207:1995-08-01 (ГОСТ Р ИСО/МЭК 12207), стандарт ISO/IEC 15288:2002 (ГОСТ Р ИСО/МЭК 15288-2005), комплекс стандартов ГОСТ 34.**

#### 14. Фирменные стандарты (Oracle CDM, MSF и др.)

Методология Oracle (Oracle Method) – комплекс методов, охватывающий большинство процессов ЖЦ ПО [40]. В состав комплекса входят:

- CDM (Custom Development Method) – разработка прикладного ПО;
- PJM (Project Management Method) – управление проектом;
- AIM (Application Implementation Method) – внедрение прикладного ПО;
- BPR (Business Process Reengineering) – реинжиниринг бизнес-процессов;
- OCM (Organizational Change Management) – управление изменениями, и др.

Custom Development Method (методика Oracle) по разработке прикладных информационных систем – технологический материал, детализированный до уровня заготовок проектных документов, рассчитанных на использование в проектах с применением Oracle.

В соответствии с этими факторами в CDM выделяются два основных подхода к разработке:

- классический подход (Classic);
- подход быстрой разработки (Fast Track).

**Классический подход (Classic)** применяется для наиболее сложных и масштабных проектов, он предусматривает последовательный и детерминированный порядок выполнения задач. Для таких проектов характерно большое количество реализуемых бизнес-правил, распределенная архитектура, критичность приложения. Применение классического подхода также рекомендуется при нехватке опыта у разработчиков, неподготовленности пользователей, нечетко определенной задаче. Продолжительность таких проектов от 8 до 36 месяцев.

**Подход быстрой разработки (Fast Track)** в отличие от каскадного классического, является итерационным и основан на методе DSDM (Dynamic Systems Development Method). В этом подходе четыре этапа – стратегия, моделирование требований, проектирование и генерация системы и внедрение в эксплуатацию. Подход используется для реализации небольших и средних проектов с несложной архитектурой системы, гибкими сроками и четкой постановкой задач. Продолжительность проекта от 4 до 16 месяцев.

#### 15. Методология моделирования функциональной структуры объектов – SADT. Методология моделирования данных – ERD (Entity-Relationship Diagrams) (case-метод Баркера).

Методология SADT (Structured Analysis and Design Technique – Технология структурного анализа и проектирования), разработана Дугласом Т. Россом в 1969-73 годах. В основе методологии SADT лежат два основных принципа:

- SA-блоки на основе которых создается иерархическая многоуровневая модульная система, каждый уровень которой представляет собой законченную систему (блок), поддерживаемую и контролируемую системой (блоком), находящейся над ней.

- Декомпозиция. Процесс декомпозиции проводится до достижения нужного уровня подробности описания. Диаграмма ограничивается 3-6 блоками для того, чтобы детализация осуществлялась постепенно, рис.1.4.



Рисунок 1.4 – Функциональный блок и интерфейсные дуги SADT-диаграммы

В программе интегрированной компьютеризации производства (ICAM) Министерства обороны США была признана полезность SADT, что привело в 1993 году к стандартизации и публикации ее части, называемой IDEF0 в качестве федерального стандарта в США, а в 2000 году – в качестве руководящего документа по стандартизации в Российской Федерации

Наиболее распространенной методологией моделирования данных являются диаграммы «сущность-связь» – ERD (Entity-Relationship Diagrams). С их помощью определяются важные для предметной области объекты (сущности), их свойства (атрибуты) и отношения друг с другом (связи). ERD непосредственно используются для проектирования реляционных баз данных.

Нотация ERD была впервые введена П.Ченом (Chen) и получила дальнейшее развитие в работах Баркера.

ERD-диаграмма, рис.1.9, позволяет рассмотреть систему целиком и выяснить требования, необходимые для ее разработки, касающиеся хранения информации.

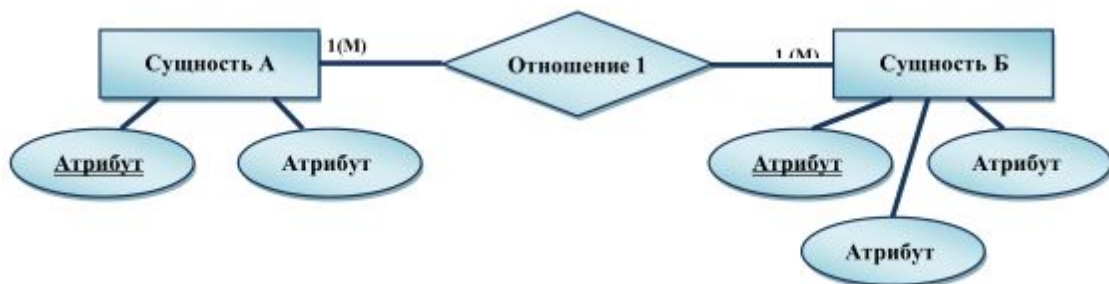


Рисунок 1.9 – Пример ER-диаграммы в нотации П.Чена

Существуют и другие нотации для представления ER-диаграмм: нотация Мартина, нотация IDEF1X (см.п.1.8.6), нотация Баркера.

## 16. Методология моделирования работы в реальном времени – STD.

Методология STD (State Transition Diagram) предназначена для моделирования аспектов функционирования системы, зависящих от времени или реакции на события (так называемая работа в реальном времени)], рис.1.10.



Рисунок 1.10 – Пример STD узлов

**Состояние** – рассматривается как устойчивое значение некоторого свойства в течение определенного времени. Начальное состояние – узел STD, являющийся стартовой точкой для начального системного перехода.

**Переход** определяет перемещение моделируемой системы из одного состояния в другое. При этом имя перехода идентифицирует событие, являющееся причиной перехода и управляющее им.

**Управляющий поток** – это «трубопровод», через который проходит управляющая информация.

Имеются следующие типы управляющих потоков:

- Т-поток (trigger flow) – поток управления процессом, который может вызвать выполнение процесса. При этом процесс включается одной короткой операцией;
- А-поток (activator flow) – поток управления процессом, который может изменять выполнение отдельного процесса. Используется для обеспечения непрерывности выполнения процесса до тех пор, пока поток «включен», с «выключением» потока выполнение процесса завершается;
- Е/D-поток (enable/disable flow) – поток управления процессом, который может переключать выполнение отдельного процесса. Течение по Е-линии вызывает выполнение процесса, которое продолжается до тех пор, пока не возбуждается течение по D-линии. Можно использовать 3 типа таких потоков: Е-поток, D-поток, Е/D-поток.

**Условие** представляет собой событие (или события), вызывающее переход и идентифицируемое именем перехода. Если в условии участвует входной управляющий поток управляющего процесса-предка, то имя потока должно быть заключено в кавычки, например. Кроме условия с переходом может связываться действие или ряд действий, выполняющихся, когда переход имеет место.

**Действие** – это операция, которая может иметь место при выполнении перехода. Если действие необходимо для выбора выходного управляющего потока, то имя этого потока должно заключаться в кавычки.

На STD **состояния** представляются узлами, а переходы – дугами. Условия (по-другому называемые стимулирующими событиями) идентифицируются именем перехода и возбуждают выполнение перехода. **Действия** или отклики на события привязываются к переходам и записываются под соответствующим усло-



вием.

## **17. Модель быстрой разработки приложений RAD (Rapid Application Development)**

Технология проектирования ИС на основе методологии использования средств быстрой разработки приложений, получила широкое распространение и приобрела название методологии быстрой разработки приложений – RAD (Rapid Application Development).

RAD – это комплекс специальных инструментальных средств быстрой разработки прикладных информационных систем, позволяющих оперировать с определенным набором графических объектов, функционально отображающих отдельные информационные компоненты приложений.

Основные принципы методологии RAD можно свести к следующему:

- используется итерационная (спиральная) модель разработки;
- полное завершение работ на каждом из этапов жизненного цикла не обязательно;
- в процессе разработки информационной системы необходимо тесное взаимодействие с заказчиком и будущими пользователями;
- необходимо применение CASE-средств и средств быстрой разработки приложений;
- необходимо применение средств управления конфигурацией, облегчающих внесение изменений в проект и сопровождение готовой системы;
- необходимо использование прототипов, позволяющее полнее выяснить и реализовать потребности конечного пользователя;
- тестирование и развитие проекта осуществляются одновременно с разработкой;
- разработка ведется немногочисленной и хорошо управляемой командой профессионалов;
- необходимы грамотное руководство разработкой системы, четкое планирование и контроль выполнения работ.

## **18. [НЕТУ] Методология функционального моделирования процессов – IDEF0. Характеристика диаграмм. Типы взаимосвязей между блоками.**

## **19. [НЕТУ] Кроссплатформенная система моделирования и анализа бизнес-процессов Ramus Educational: возможности, особенности применения при проектировании ИС.**

**20. [НЕТУ] Методология функционального моделирования процессов – IDEF0. Последовательность создания функциональных моделей.**

**21. [Дописать DEF1X] Технология анализа взаимосвязей между информационными потоками – IDEF1 (IDEF1X).**

Методология IDEF1 разработанная Т.Рэмеем, основана на подходе П.Чена и позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

Методология IDEF1 позволяет на основе наглядных графических представлений моделировать информационные взаимосвязи и различия между:

- реальными объектами;
- физическими и абстрактными зависимостями, существующими среди реальных объектов;
- информацией о реальных объектах;
- структурой данных, используемой для приобретения, накопления и управления информацией.

Центральным понятием IDEF1 является понятие «сущность», рис.1.11.

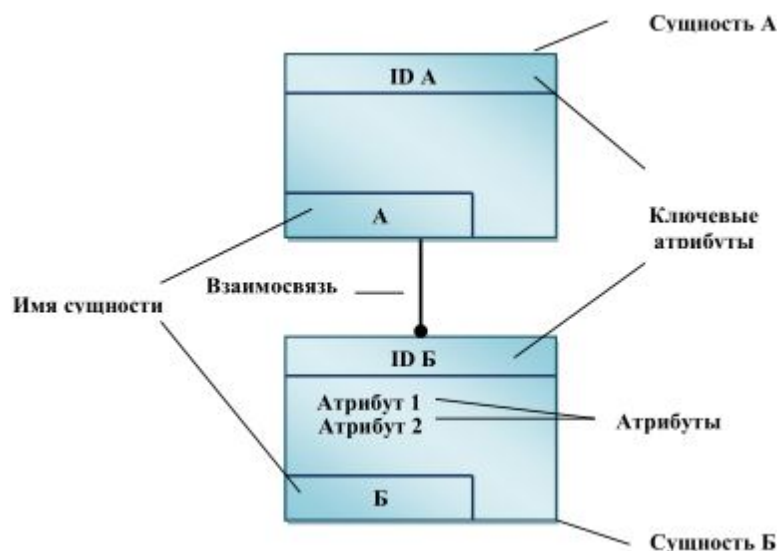


Рисунок 1.11 – Стандарт IDEF1

IDEF1X – методология, предназначена для построения концептуальной схемы логической структуры реляционной базы данных, которая была бы независимой от программной платформы её конечной реализации.

## 22. [HETU] Средства СА ERwin Data Modeler Community Edition: возможности, особенности применения.

## 23. Методология описания (документирования) и моделирования процессов – IDEF3.

Нотация IDEF3, вторая важнейшая нотация (после IDEF0), предназначена для описания потоков работ (Work Flow Modeling).

С помощью диаграмм IDEF3 можно анализировать сценарии из реальной жизни, например, как закрывать магазин в экстренных случаях или какие действия должны выполнить менеджер и продавец при закрытии.

Существуют два типа диаграмм в стандарте IDEF3, представляющие описание одного и того же сценария процесса в разных ракурсах:

- диаграммы Описания Последовательности Этапов Процессы – PFDD (Process Flow Description Diagrams).
- диаграммы Состояния Объекта и его Трансформаций в Процессы – OSTN (Object State Transition Network).

Диаграмма PFDD, рис.1.12 является графическим отображением сценария процесса.

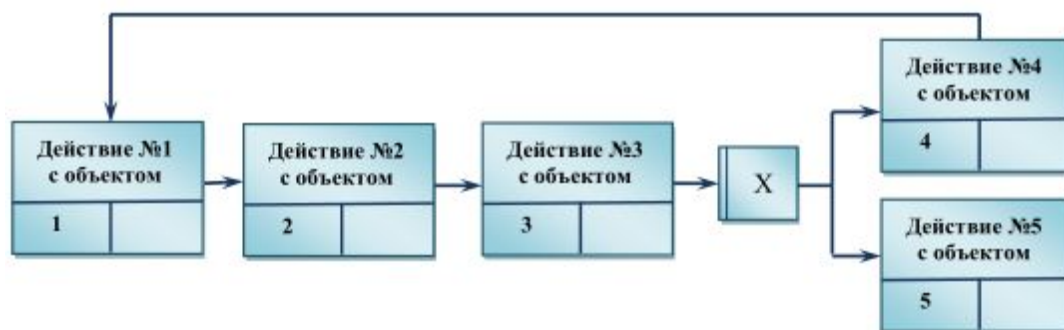


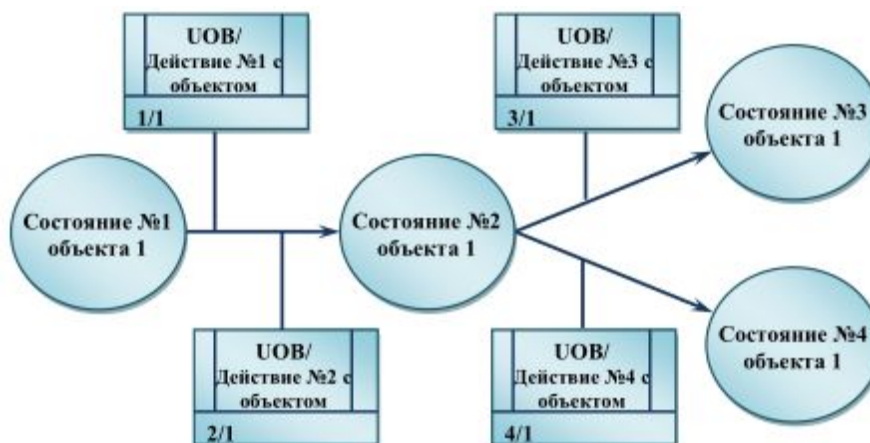
Рисунок 1.12 – Пример PFDD диаграммы

Прямоугольники на диаграмме PFDD называются функциональными элементами или элементами поведения – UOB (Unit of Behavior,) и обозначают событие, стадию процесса или принятие решения.

Если диаграммы PFDD технологический процесс «С точки зрения наблюдателя», то другой класс диаграмм IDEF3 OSTN позволяет рассматривать тот же самый процесс «С точки зрения объекта». На рис.1.13 представлено пример изображения OSTN диаграммы.

Состояния объекта и Изменение состояния являются ключевыми понятиями OSTN диаграммы. Состояния объекта отображаются окружностями, а их изменения направленными линиями. Каждая линия имеет ссылку на соответствующий функциональный блок UOB, в результате которого произошло отображаемое ею изменение состояния объекта.

Нотацию IDEF3 целесообразно применять в случае относительно простых процессов на нижнем уровне декомпозиции, т.е. процессов уровня рабочих мест. В этом случае схема процесса может служить основой для создания документов, регламентирующих работу исполнителей.



## 24. [НЕТУ] Диаграммы PFDD (Process Flow Description Diagrams), OSTN (Object State Transition Network).

## 25. Методология рационального унифицированного процесса RUP (Rational Unified Process).

Еще одной ведущей методологией, в которой инструментально поддерживаются все этапы жизненного цикла разработки ПО, является методология Rational Unified Process (RUP). В основе методологии RUP, как и многих других программных методологий, объединяющих инженерные методы создания ПО, лежит «пошаговый подход». Он определяет этапы жизненного цикла, контрольные точки, правила работ для каждого этапа и, тем самым, упорядочивает проектирование и разработку ПО. Для каждого этапа жизненного цикла методология задает: состав и последовательность работ, а также правила их выполнения; распределение полномочий среди участников проекта (роли); состав и шаблоны формируемых промежуточных и итоговых документов; порядок контроля и проверки качества. Модели позволяют рассмотреть будущую систему, ее объекты и их взаимодействие еще до вкладывания значительных средств в разработку, позволяют увидеть ее глазами будущих пользователей снаружи и разработчиков изнутри еще до создания первой строки исходного кода. Большинство моделей представляются UML диаграммами [39].

Определение требований. Унифицированный процесс – это процесс, управляемый прецедентами, которые отражают сценарии взаимодействия пользователей. Фактически, это взгляд пользователей на программную систему снаружи. Таким образом, одним из важнейших этапов разработки, согласно RUP, будет этап определения требований, который заключается в сборе всех возможных пожеланий к работе системы, которые только могут прийти в голову пользователям и аналитикам. Для облегчения этого процесса аналитики используют Диаграммы Прецедентов

(Вариантов использования) (Use Case Diagram), рис.1.19. На диаграмме отображаются варианты использования (1) и действующие лица (2), между которыми устанавливаются следующие основные типы отношений: ассоциация между действующим лицом и вариантом использования (3); обобщение между действующими лицами (4); обобщение между вариантами использования (5); зависимости (различных типов) между вариантами использования (6).

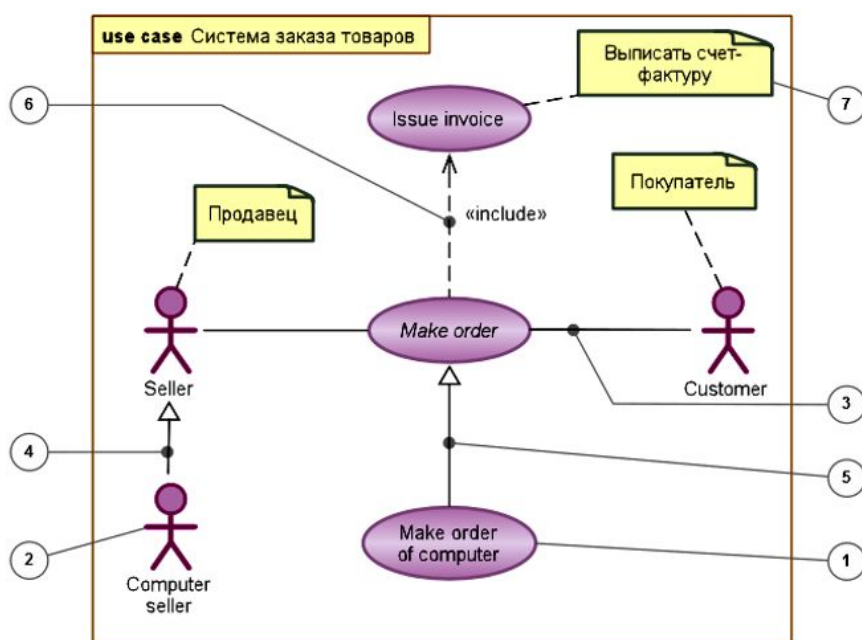


Рисунок 1.19 – Диаграммы Прецедентов (Вариантов использования) (Use Case Diagram)

Для детализации конкретного прецедента используется Диаграмма Активности (Activity Diagram), пример которой дан на рис.1.20. На диаграмме применяют один основной тип сущностей – действие (1), и один тип отношений – переходы (2) (передачи управления и данных). Также используются такие конструкции как развилки, слияния, соединения, ветвления (3), которые похожи на сущности, но таковыми на самом деле не являются, а представляют собой графический способ изображения некоторых частных случаев многоместных отношений.

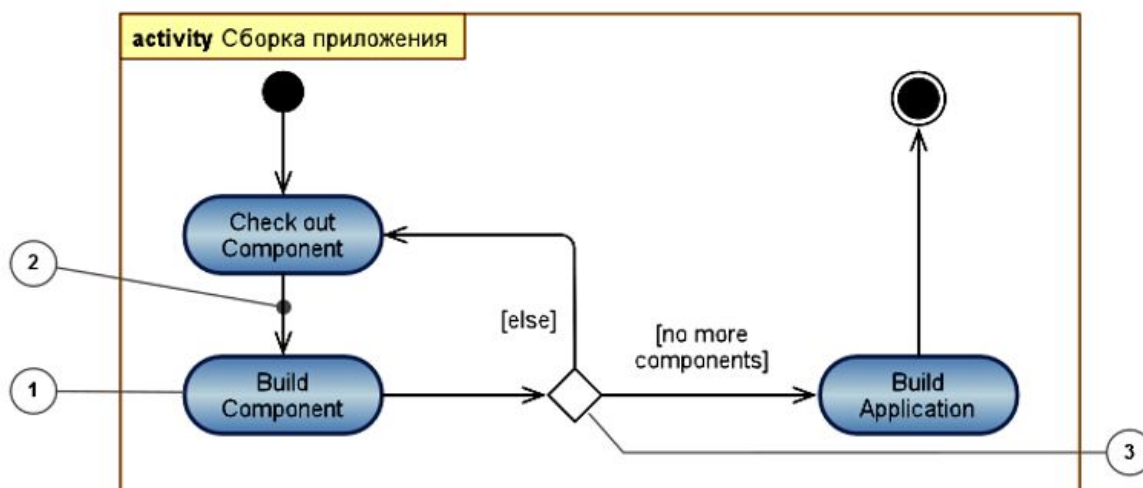


Рисунок 1.20 – Диаграмма Активности (Activity Diagram)

Для создания модели предметной области используется обычная Диаграмма Классов (Class Diagram), рис.1.21, на которой отображаются: основной тип сущностей: классы (1), между которыми устанавливаются следующие основные типы отношений: ассоциация между классами (2); обобщение между классами (3); зависимости (различных типов) между классами 4 и между классами и интерфейсами.).

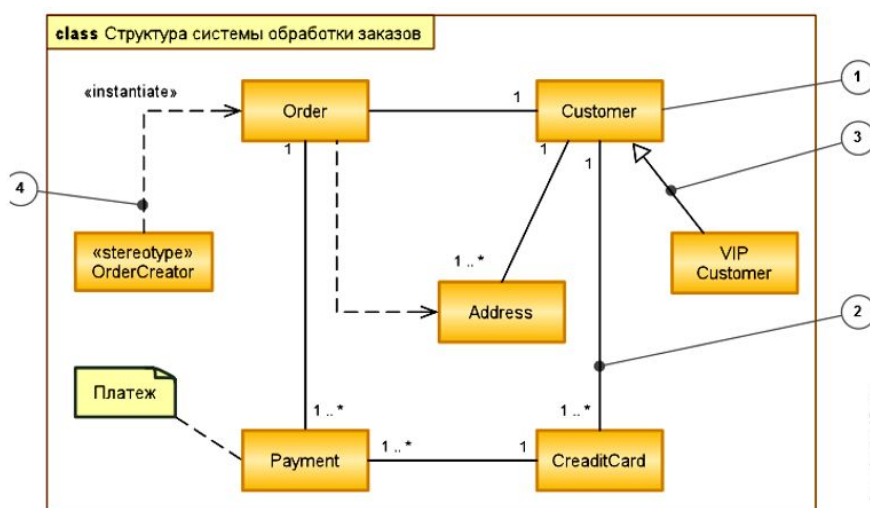


Рисунок 1.21 – Диаграмма Классов (Class Diagram)

Анализ. После определения требований и контекста, в котором будет работать система, наступает черед анализа полученных данных. Аналитическая модель – это взгляд на систему изнутри, в отличие от модели прецедентов, которая показывает, как система будет выглядеть снаружи. Для отображения модели анализа при помощи UML используется Диаграмма Классов со стереотипами (образцами поведения) «граничный класс», «сущность», «управление», а для детализации используются Диаграммы Сотрудничества (Collaboration), рис.1.22.

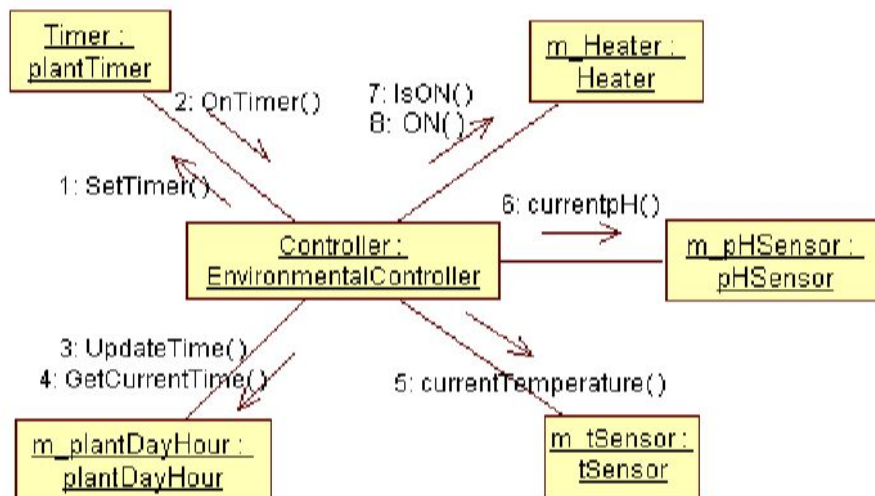


Рисунок 1.22 – Диаграммы Сотрудничества (Collaboration)

Если акцентировать внимание на порядке взаимодействия, то другим его представлением будет Диаграмма Последовательности (Sequence Diagram), рис.1.23.

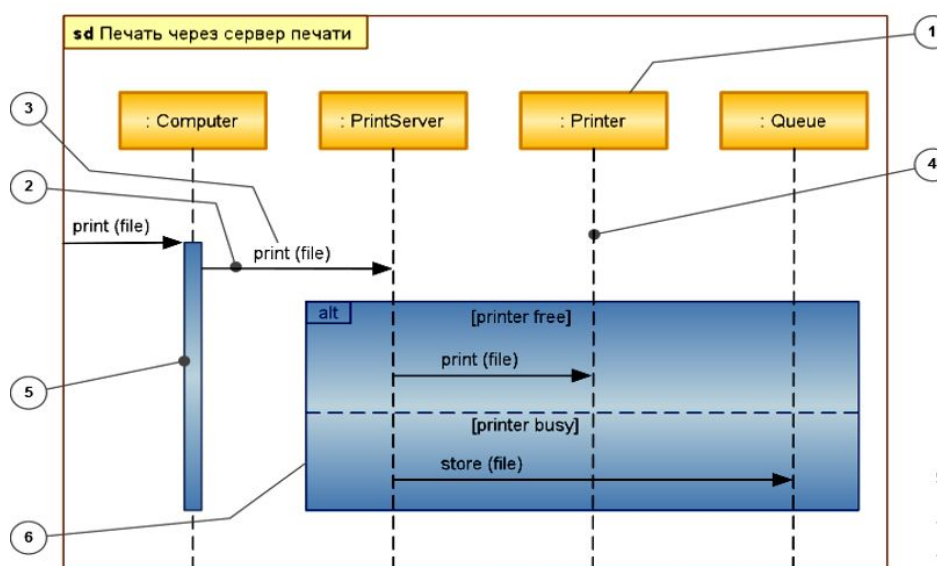


Рисунок 1.23 – Диаграмма Последовательности (Sequence Diagram)

Эта диаграмма позволяет взглянуть на обмен сообщениями во времени, наглядно отобразить последовательность процесса. На диаграмме применяют один основной тип сущностей – экземпляры взаимодействующих классификаторов (1) (в основном классов, компонентов и действующих лиц), и один тип отношений – связи (2), по которым происходит обмен сообщениями (3). Проектирование в ходе которого на основании моделей, созданных ранее, создается модель проектирования. Для создания модели проектирования используются целый набор UML диаграмм: Диаграммы Активности, рис.1.20, Диаграммы Классов, рис.1.21, Диаграммы Коммуникации, рис.1.24, Диаграммы Взаимодействия, рис.1.25.



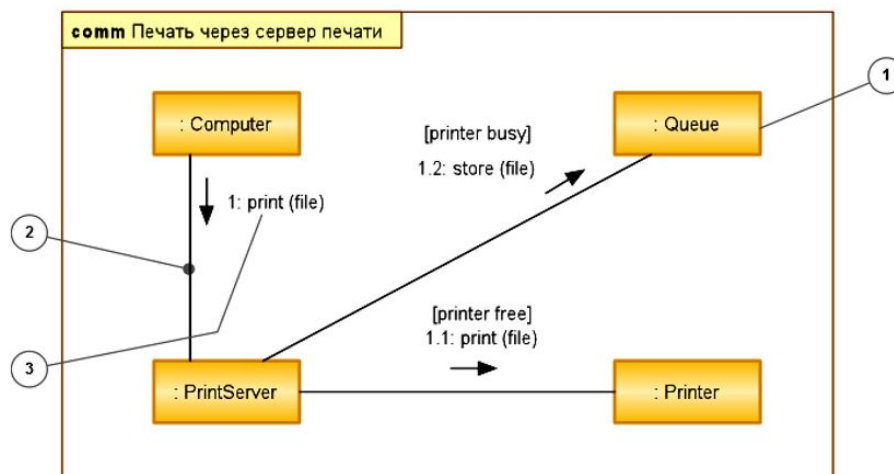


Рисунок 1.24 – Диаграммы Коммуникации (Communication Diagram)

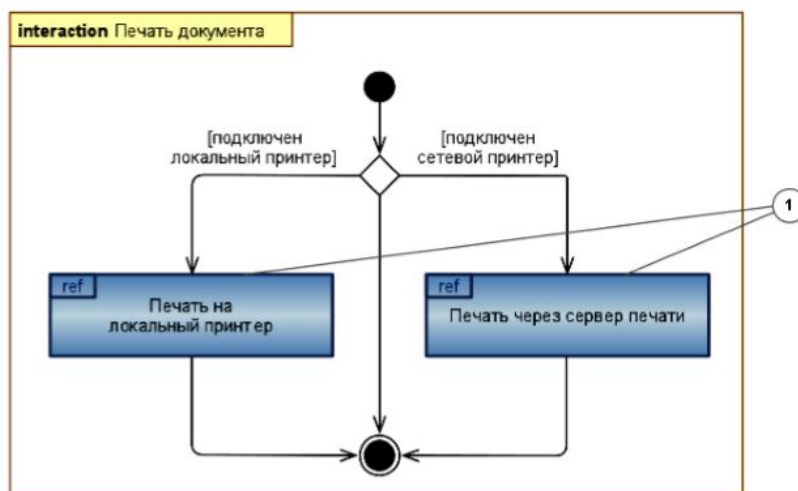


Рисунок 1.25 – Диаграммы Взаимодействия (Interaction diagrams)

Дополнительно в этом рабочем процессе может создаваться модель развертывания, которая реализуется на основе Диаграммы Развертывания (Deployment Diagram), рис.1.26. Это самый простой тип диаграмм, предназначенный для моделирования распределения устройств в сети. На диаграмме присутствует два типа сущностей: артефакт (1), который является реализацией компонента (2) и узел (3) (может быть как классификатор, описывающий тип узла, так и конкретный экземпляр), а также отношение ассоциации между узлами (4), показывающее, что узлы физически связаны во время выполнения. Если одна сущность является частью другой, применяется либо отношение зависимости «deploy» (5), либо фигура одной сущности помещается внутрь фигуры другой сущности (6).



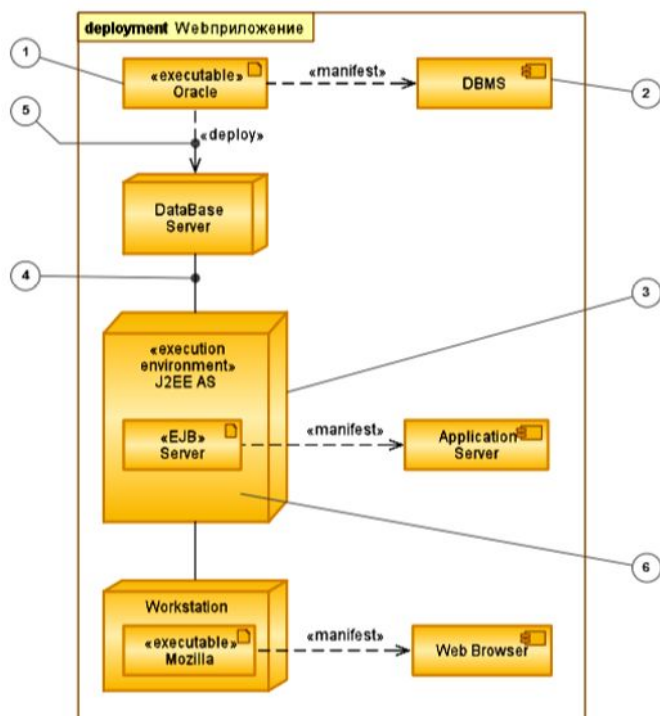


Рисунок 1.26 – Диаграммы Развертывания (Deployment Diagram)

Реализация. Основная задача процесса реализации – создание системы в виде компонентов – исходных текстов программ, сценариев, двоичных файлов, исполняемых модулей и т.д. Данная модель описывает способ организации этих компонентов в соответствии с механизмами структурирования и разбиения на модули, принятыми в выбранной среде программирования и представляется Диаграммой Компонентов (Component Diagram), рис.1.27.

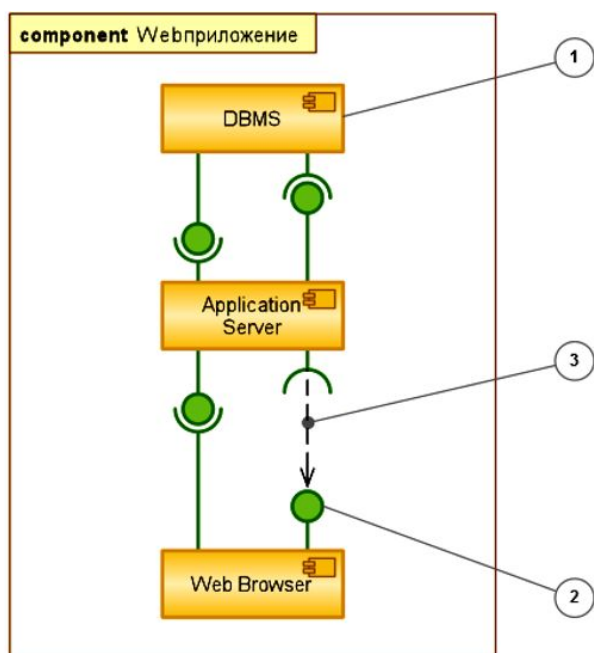


Рисунок 1.27 – Диаграммы Компонентов (Component Diagram)

Основной тип сущностей на диаграмме компонентов – это сами компоненты (1), а также интерфейсы (2), посредством которых указывается взаимосвязь между компонентами. На диаграмме компонентов применяются следующие отношения: реализации между компонентами и интерфейсами (компонент реализует интерфейс); зависимости между компонентами и интерфейсами (компонент использует интерфейс) (3). Тестирование. В процессе тестирования проверяются результаты реализации. Для данного процесса создается модель тестирования, которая состоит из тестовых примеров, процедур тестирования, тестовых компонентов, однако не имеет отображения на UML диаграммы. RUP довольно обширен, и содержит рекомендации по ведению различных программных проектов, от создания программ группой разработчиков в несколько человек, до распределенных программных проектов, объединяющих тысячи человек на разных континентах.

## **26. [НЕТУ] Диаграмма Последовательности (Sequence Diagram) и диаграмма Ганта: сходства, отличия, назначение.**

## **27. Технологии, основанные на моделировании, анализе бизнес-процессов BPMN.**

Модель и нотация бизнес-процессов (BPMN, Business Process Model and Notation) – методология моделирования, анализа и реорганизации бизнес-процессов. Разработана Business Process Management Initiative (BPMI), с 2005 г. поддерживается и развивается Object Management Group (OMG). В отличие от других методологий бизнес-моделирования, имеющих статус «фирменного» (EPC) или «национального» (IDEF0) стандарта, BPMN получила «международный» статус – Международная организация по стандартизации опубликовала стандарт «ISO/IEC 19510:2013. Information technology – Object Management Group. Business Process Model and Notation».

Диаграмма процесса в нотации BPMN представляет собой алгоритм выполнения процесса. В нотации BPMN выделяют пять основных категорий элементов:

- элементы потока (Flow Objects) (события, процессы и шлюзы);
- данные (объекты данных и базы данных) (товарно-материальные ценности (ТМЦ) или информация);
- соединяющие элементы (Connecting Objects) (потoki управления, потоки сообщений и ассоциации);
- зоны ответственности (Swimlanes) (пулы и дорожки);
- артефакты (сноски) (Artifacts).

**Элементы потока** являются важнейшими графическими элементами, определяющими ход бизнес-процесса. Элементы потока, в свою очередь, делятся на: события (Events); действия (Activities); шлюзы (Gateways).

**Артефакты** используются для добавления дополнительной информации о Процессе. Текущий перечень Артефактов включает в себя следующие элементы:

объект данных (Data object); группа (Group); аннотация (Annotation).

Разновидности диаграмм (типы процессов) BPMN:

диаграмма процессов (Process Diagram):

- частный (внутренний) бизнес-процесс (Private (internal) Business Process), рис.1.14;
- публичный (открытый) процесс (Public Process), рис.1.15;
- диаграмма хореографии (Choreography Diagram), рис.1.16;
- диаграмма взаимодействия (Collaboration Diagram):
- процессов (Process), рис.1.17;
- посредством обмена сообщениями (A view of Conversations), рис.1.18.



Рисунок 1.14 – Примерный вид BPMN диаграммы частного (внутреннего) бизнес-процесса

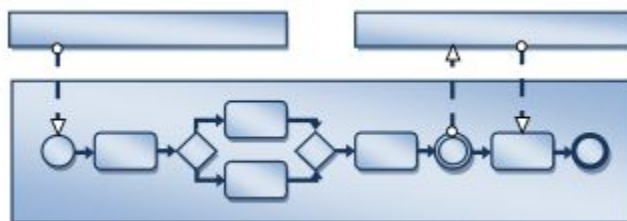


Рисунок 1.15 – Примерный вид BPMN диаграммы публичного (открытого) бизнес-процесса



Рисунок 1.16 – Примерный вид BPMN диаграммы хореографии

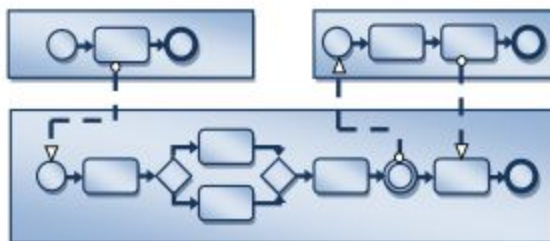


Рисунок 1.17 – Примерный вид BPMN диаграммы взаимодействия процессов

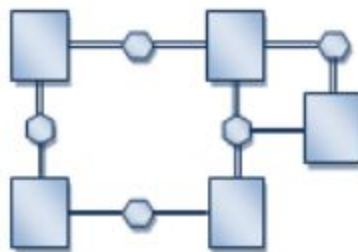


Рисунок 1.18 – Примерный вид BPMN диаграммы посредством обмена сообщениями

## **28. Инструмент для моделирования бизнес-процессов ARIS Express: типы диаграмм, возможности.**

Концепция Архитектуры Интегрированных Информационных Систем - ARIS (ARchitecture of Integrated Information Systems) разработана профессором А.В. Шеером (Scheer).

Эта концепция имеет два основных преимущества:

- позволяет выбрать методы и интегрировать их, опираясь на основные
- особенности моделируемого объекта;
- служит базой для управления сложными проектами, поскольку благодаря структурным элементам содержит встроенные модели процедур для разработки интегрированных информационных систем.

Архитектура ARIS явилась основой ARIS Toolset – инструментальной среды, разработанной компанией IDS Scheer AG.

В методологии ARIS для описания различных подсистем организации используется более ста типов моделей, отражающих различные аспекты деятельности и реализующих различные методы моделирования (в том числе событийная цепочка процесса EPC (Event driven Process Chain):

- модель «сущность-связь» ERM (Entity Relationship Model);
- модели методики объектно-ориентированного моделирования OMT (Object Modeling Technique);
- модели BSC (Balanced Scorecard) – система сбалансированных показателей;
- модели UML и многие другие.

Все многообразие типов моделей ARIS подразделяется на пять видов описания в соответствии с основными подсистемами предприятия:

- организационной;
- функциональной,
- подсистемой данных;
- подсистемой процессов;
- подсистемой продуктов/услуг.

Остальные подсистемы могут моделироваться с использованием типов объектов, входящих в перечисленные виды описания.

**29. [НЕТУ] Реинжиниринг бизнес-процессов в ИС (реорганизации бизнес-процессов – BPMN) при проектировании ИС.**

**30. [НЕТУ] Технологии объектно-ориентированного подхода (анализа) при проектировании ИС.**

**31. [НЕТУ] Технология объектно-ориентированного подхода с использованием паттернов.**

**32. [НЕТУ] Средства имитационного моделирования ИС на основе технологии модельно-ориентированного подхода.**

**33. [НЕТУ] Основные подходы в имитационном моделировании ИС и соответствующие технологии.**

**34. [НЕТУ] Средства модельно-ориентированного подхода (анализа): MathLab Sumulink, VinSim, AnyLogic.**

**35. [НЕТУ] Инструмент для разработки и исследования имитационных моделей AnyLogic: панели, окна, редакторы.**

## 36. Гибкие методологии проектирования (agile-методы).

Agile – это семейство методологий разработки программного обеспечения, для которых характерны:

- итеративный процесс разработки;
- динамичное формирование требований и их реализация;
- тесная взаимосвязь, активное взаимодействие, как между абсолютно всеми участниками команды разработчиков, так и между командой и заказчиком.

Agile Manifesto содержит 4 основные идеи и 12 принципов.

### **Основные идеи:**

*Люди и взаимодействие важнее процессов и инструментов*

*Работающий продукт важнее исчерпывающей документации*

*Сотрудничество с заказчиком важнее согласования условий контракта*

*Готовность к изменениям важнее следования первоначальному плану*

### **Основные принципы:**

- наивысшим приоритетом для нас является удовлетворение потребностей заказчика, благодаря регулярной и ранней поставке ценного программного обеспечения;
- изменение требований приветствуется, даже на поздних стадиях разработки;
- работающий продукт следует выпускать как можно чаще, с периодичностью от пары недель до пары месяцев;
- на протяжении всего проекта разработчики и представители бизнеса должны ежедневно работать вместе;
- над проектом должны работать мотивированные профессионалы;
- непосредственное общение является наиболее практичным и эффективным способом обмена информацией как с самой командой, так и внутри команды;
- работающий продукт – основной показатель прогресса;
- инвесторы, разработчики и пользователи должны иметь возможность поддерживать постоянный ритм бесконечно;
- постоянное внимание к техническому совершенству и качеству проектирования повышает гибкость проекта;
- простота – искусство минимизации лишней работы – крайне необходима;
- самые лучшие требования, архитектурные и технические решения рождаются у самоорганизующихся команд;
- команда должна систематически анализировать возможные способы улучшения эффективности и соответственно корректировать стиль своей работы.

Примечательно, что Agile Manifesto не содержит практических советов.

Agile Modeling (AM) – это набор понятий, принципов и приемов (практик), позволяющих быстро и просто выполнять моделирование и документирование в проектах разработки программного обеспечения.

AM описывает стиль моделирования, который позволит повысить качество и сократить сроки. AM не является технологическим процессом. Это не детальная инструкция по проектированию, он не содержит описаний, как строить диаграммы на UML. AM сосредоточен на эффективном моделировании и документировании. Он не охватывает программирование и тестирование, хотя в нем и говорится о проверке

модели кодом и рассматривается тестируемость моделей. АМ также не включает вопросы управления проектом, развертывания и сопровождения системы.

АМ должен рассматриваться как дополнение к существующим методам, а не самостоятельная технология. Этот метод должен использоваться для повышения эффективности труда разработчиков, использующих процессы eXtreme Programming (XP), Dynamic Systems Development Method (DSDM), или RUP.

**37. [НЕТУ] Понятие CASE-технологии проектирования ИС. Основные принципы CASE - технологии.**

**38. [НЕТУ] Классификация CASE-средств, стратегия их выбора.**

**39. [НЕТУ] Технологии планирования проектов по стандарту PMBOK Guide – Руководству к своду знаний по управлению проектами.**

**40. Метод PERT (Program Evaluation and Review Technique) при проектировании ИС.**

Метод PERT – метод событийного сетевого анализа, используемый для определения длительности проекта при наличии неопределенности в оценке продолжительностей индивидуальных операций.

PERT основан на методе критического пути, длительность операций в котором рассчитывается как взвешенная средняя оптимистического, пессимистического и ожидаемого прогнозов. PERT рассчитывает стандартное отклонение даты завершения от длительности критического пути.

Продолжительность работы рассчитывается в данном случае как средневзвешенная средняя оптимистического, пессимистического и ожидаемого прогнозов.

**ПРИМЕР**

«Поиск практического материала для написания дипломного проекта»

Указываем оптимистический прогноз продолжительности выполнения данной работы, пессимистический прогноз, вероятности.

	Оптимистиче ский	Наиболе е вероятный	Пессимистич еский
Прогнозное значение	3д	10д	30д
Вероятность осуществления прогноза	0,2	0,6	0,2
Взвешенное значение	0,6д	6д	6д
Сумма взвешенных значений	12,6д		

Необходимо перемножить значение каждого прогноза на его вероятность и полученные величины сложить.

Таким образом, получится средневзвешенное значение продолжительности выполнения работы.

$$3д * 0,2 + 10д * 0,6 + 30д * 0,2 = 0,6д + 6д + 6д = 12,6д$$

Примерная продолжительность работы около 13 дней.

**41. [НЕТУ] Средства планирования проектов: OpenProj, Microsoft Project.**

**42. [НЕТУ] Проблемы и перспективы проектирования ИС: мультизадачность ИС, гетерогенность системы и среды, неопределенность задач и требований, Runtime, On-line проектирование.**