

- 1. Принципы классификации информационных систем. Основания классификации.
- **Информационная система** — прикладная программная система, ориентированная на сбор, хранение, поиск и обработку информации.
- **Информационная система** – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.
- **Основания классификации:**
 - - область применения;
 - - уровень функциональности;
 - - масштаб использования;
 - - способ организации работ.

- 2. Классификация информационных систем по области применения.
- Деление систем, входящих в ТхПП и ТПП на подклассы.
- **По области применения разделим ИС (информационных систем) на следующие классы:**
 - - системы для технической (ТхПП) и технологической подготовки производства (ТПП);
 - - системы для управления производственным процессом в режиме реального времени;
 - - системы для управления предприятием;
 - - системы для отслеживания жизненного цикла изделия (**PDM** - системы);
 - - системы общего назначения.
- **Системы для ТхПП и ТПП могут быть разделены на следующие подклассы:**
 - - системы конструирования изделий, сборочных единиц и деталей (**CAD** – системы);
 - - системы расчета (анализа) конструкций и деталей (**CAE** - системы);
 - - системы проектирования электронных компонентов изделий (**ECAD** - системы);
 - - системы разработки и верификации управляющих программ (**CAM** - системы);
 - - системы проектирования технологических процессов (**CAPP** - системы);
 - - системы проектирования средств технологического оснащения;
 - - информационно-поисковые системы технологического назначения (**ИПС ТН**);
 - - автономные системы для решения отдельных технологических задач.

- 3. Классификация информационных систем по области применения.
- Системы для управления предприятием.
- **По области применения разделим ИС (информационных систем) на следующие классы:**
- - системы для технической (ТхПП) и технологической подготовки производства (ТПП);
- - системы для управления производственным процессом в режиме реального времени;
- - системы для управления предприятием;
- - системы для отслеживания жизненного цикла изделия (**PDM** - системы);
- - системы общего назначения.
- **Системы для управления предприятием:**
- - оперативное управление предприятием;
- - управление финансовыми потоками — расчет поставщиков с потребителями;
- - управление складом, ассортиментом и закупками;
- - бухгалтерский учёт;
- - управление маркетингом.

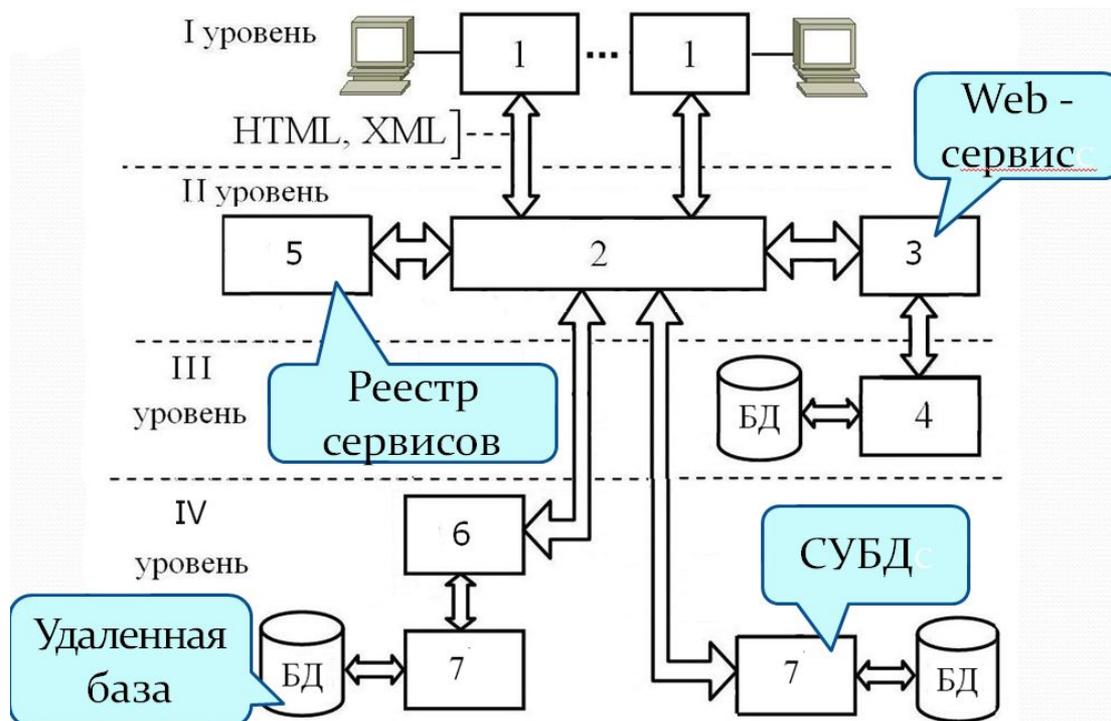
- 4. Классификация информационных систем по области применения.
- Системы общего назначения.
- **По области применения разделим ИС (информационных систем) на следующие классы:**
 - - системы для технической (ТхПП) и технологической подготовки производства (ТПП);
 - - системы для управления производственным процессом в режиме реального времени;
 - - системы для управления предприятием;
 - - системы для отслеживания жизненного цикла изделия (PDM - системы);
 - - системы общего назначения.
- **Системы общего назначения:**
 - - системы для автоматизации документооборота;
 - - системы управления базами данных (СУБД);
 - - редакционно-издательские системы;
 - - графические редакторы;
 - - экспертные оболочки;
 - - средства для создания электронных курсов и тестов;
 - - средства анализа и моделирования систем;
 - - средства проектирования автоматизированных систем.

- 5. Классификация информационных систем по уровню функциональности и масштабу использования.
- **Уровень функциональности**
- - на узкоспециализированные системы;
- - на системы с широким набором функций;
- - на системы представляющие собой совокупность программных комплексов.
- **Масштаб использования :**
- - одиночные — реализованы на отдельных ПК, рассчитаны на одного пользователя;
- - групповые — рассчитаны на одновременное использование коллективом предприятия и строятся на базе локальной сети;
- - корпоративные — рассчитаны на крупные организации
- 6. Классификация информационных систем по способу организации их работы.
- **По способу организации работы:**
- - Файл-сервер;
- - Клиент-сервер (3-х уровневая архитектура)
- 1. выделенный сервер БД;
- 2. серверное приложение;
- 3. клиентское приложение.
- - Многоуровневая архитектура (удаленные базы данных)
- - Многоуровневая архитектура на основе Интернет технологий (удаленные приложения и удаленные базы данных)
-

- 7. Основные требования к информационным системам.
- **Информационная система** — прикладная программная система, ориентированная на сбор, хранение, поиск и обработку информации.
- **Информационная система** – взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.
- **Требования к ИС:**
 - - повышение функциональности и повышение адаптации к изменяющимся условиям ее функционирования
 - - повышение пропускной способности системы
 - - короткое время реакции системы на запрос
 - - безотказная работа системы в требуемом режиме, иными словами – готовность и доступность системы для обработки запросов пользователей
 - - простота эксплуатации и поддержки системы
 - - необходимая безопасность
- **Техническое задание (ТЗ)** – это документ, определяющий цели, требования и основные исходные данные, необходимые для разработки автоматизированной системы управления.
- **Под проектированием ИС** – понимается процесс преобразования входной информации об объекте, методах и опыте проектирования объекта аналогично назначению в соответствии с ГОСТом в проекте ИС.
-

- 8. Виды обеспечения информационных систем.
- Входной поток $X=\{x(i)\}$
- Выходной поток $Y=\{x(i)\}$
- Правила (алгоритмы) F поведения системы $X \rightarrow Y$
- **Виды обеспечения:**
- - **методическое** (комплекс эксплуатационных документов, а так же технический и рабочий проект ИС);
- - **математическое** (методы и алгоритмы для ИС);
- - **программное** (комплекс программ для ИС);
- - **информационное** (информационная база и базы данных)
- - **техническое** (компьютеры, сетевые средства и серверы);
- - **организационно-правовое** (комплекс документов, регламентирующих организацию ИС, функции подразделений, права доступа и меру ответственности отдельных лиц).

- 9. Архитектура информационных систем типа «клиент – сервер».
- **Архитектура ИС типа «Клиент – сервер»**
- I Уровень. Клиентское приложение (HTML, XML). Работа с пользователями, получение информации с сервера. Отправка информации на сервер.
- II Уровень. Серверное приложение. Общение с I Уровнем. Вычисление, оптимизация, передача на III Уровень.
- III Уровень. Удаленное СУБД-> удаленная БД. Удаленные приложения могут «блуждать» по интернету, нужно хранить информацию о приложении – реестр.
- Добавляется IV Уровень если есть удаленные БД



- 10. Основные требования к корпоративным информационным системам.
- **КИС (Корпоративные информационные системы)** – это информационная система с многоуровневой архитектурой, имеющая удаленные приложения и базы данных, и ориентированная на использование Web -сервисов.
- **Требования к КИС, необходимо:**
 - 1. организовать доступ к удаленным приложениям, решающим необходимые пользователю задачи;
 - 2. разработать способы адаптации приложений к конкретным «виртуальным» автоматизированным рабочим местам -- удаленное место
 - 3. организовать единое информационное пространство на основе удаленных баз данных и знаний (единый язык «общения»)
 - 4. определить способы сопровождения удаленных приложений, баз данных и знаний (корректировка, добавление, удаление, функциональное расширение и исправление ошибок)
 -

- 11. Основные свойства корпоративных информационных систем.
- 1. Системность (- это совокупность взаимосвязанных элементов, образующих целостность)
- 2. Комплектность (сформулированность, полнота)
- 3. Модульность (разделяет систему на части, со слабыми связями между собой). Модульность – принцип построения технических систем, согласно которому функционально связанные части группируются в законченные узлы – модули.
- 4. Открытость (возможность доступа к алгоритмам программ для их исправления и дополнения)
- 5. Адаптивность (быстро приспосабливаться к условиям предприятия)
- 6. Надежность – свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, хранения и транспортирования
- 7. Безопасность – это такое состояние сложной системы, когда действие внешних и внутренних факторов не приводит к ухудшению системы или к невозможности ее функционирования.
- 8. Масштабируемость (под разные системы, разные платформы) – способность системы справляться с увеличением рабочей нагрузки (увелич. производительность) при добавлении ресурсов (обычно аппаратных)
- 9. Мобильность (легкость установки) – наличие универсального доступа к приложениям
- 10. Простота в изучении
- 11. Поддержка внедрения и сопровождение со стороны разработчика (облачные технологии)

- 12. Использование сервис-ориентированной архитектуры создания КИС. Дать определения следующим понятиям: сервис, поставщик сервиса, потребитель сервиса.
- Использование сервис-ориентированной архитектуры (Service-Oriented Architecture или SOA).
- **SOA** - это компонентная модель, основанная на взаимодействии модулей приложений, называемых **web-сервисами** (или web-службами), посредством стандартных интерфейсов и соглашений между ними.
- **Web-сервис** – это программная компонента, доступная через глобальную (или локальную) вычислительную сеть и не привязанная к каким-либо конкретным языкам программирования или операционным системам. (Плюсы: стоимость ниже до 80%, легче сопровождение)
- Использование SOA позволяет значительно снизить **затраты на внедрение и общую стоимость** владения программным обеспечением
- Терминология
- **Сервис** - задача, выполняемая web-сервисом.
- **Поставщик сервиса** (Service Provider) - модуль, к которому обратились за предоставлением сервиса.
- **Потребитель сервиса** (Service Requestor) – модуль, который затребовал какой-либо сервис.
- Каждый web-сервис может выступать как поставщик, так и как потребитель сервисов!
-

- 13. Состав корпоративных информационных систем.

- **Состав КИС**

- 1. Клиентское приложение;
- 2. Управляющий модуль;
- 3. Реестр сервисов (Service Registry)
 - а) часть управляющего модуля;
 - б) централизованный каталог UDDI (Universal Description Discovery & Integration) – специального модуля для описания web-сервисов на языке WSDL
- 4. Web-сервисы (внутренние и удаленные);
- 5. СУБД

- 14. Определение понятия «бизнес – процесс» и параметры деловой операции.
Пример деловой операции.
- **Бизнес-процесс** (деловой процесс) – последовательность выполнения деловых операций (действий), необходимых для решения поставленной задачи.
Параметры деловой операции:
 - подзадача (действия исполнителя);
 - исполнитель;
 - дата начала операции;
 - плановая дата завершения операции;
 - фактическая дата завершения операции.
- **Пример делового процесса «Технологическое проектирование»**
- Деловые операции:
 - - Проектирование технологического процесса.
 - - Разработка управляющей программы.
 - - Проектирование специальной оснастки (штампы, пресс-формы, приспособления, инструмент).
 - - Утверждение технологической документации.
 - - Изготовление специальной оснастки.
 - - Внедрение технологического процесса.
- *Примечание:* Данный ДП носит обобщенный характер. В действительности ДП имеет обратные связи и может быть выражен в виде графических схем.
-

- 15. Стадии создания и внедрения информационных систем.

- - разработка технического задания;
- - разработка эскизного проекта (не обязательна);
- - разработка технического проекта;
- - разработка рабочего проекта;
- - промышленное внедрение системы.

- Эти стадии приведены в ГОСТ 19.102-77 ЕСПД.

- Аналогичные стадии имеются в стандартах CALS

- 16. Стадии разработки технического задания при проектировании ИС.

- - обоснование необходимости разработки системы;
- - выполнение научно-исследовательских работ (предпроектный анализ бизнес-процессов);
- - разработка и утверждение технического задания.

-

- 17. Предпроектный анализ бизнес – процессов и результаты разработки технического задания.
- **Предпроектный анализ** - на этой стадии мы проверяем, а что у нас есть (составляются бизнес процессы)
- **Точка зрения на БП:**
 - 1. функциональная (на ее основе составляется функциональная модель, связаны информационными потоками – информация, выраженная инф. потоками обычно фиксируется в базах данных и считывается в тот момент, когда информация становится функ-ой.);
 - 2. информационная (рассматривает информационные потоки для создания информационной модели); - это потоки документов.
 - 3. организационная (для создания организационных моделей, в которых фиксируются подразделения и связи между ними.)
- **Результат разработки ТЗ:**
 - 1. Сформулированы требования к системе (определены объекты автоматизации и требования к ним),
 - 2. Указаны этапы и сроки разработки системы;
 - 3. Определена экономическая эффективность результатов автоматизации.
- **Предпроектный анализ (ПА)** проводится с целью выяснения двух принципиальных моментов:
 - 1. целесообразно ли проводить автоматизацию процессов управления на данном конкретном предприятии?
 - 2. готово ли предприятие/заказчик к проведению такой автоматизации?

- В том случае, если на второй вопрос в процессе ПА был получен отрицательный ответ, принимаются дополнительные меры по подготовке предприятия к автоматизации. После этого изучаются и описываются существующие бизнес-процессы, разрабатываются предложения по их совершенствованию (так называемый "реинжиниринг"). Именно проведение качественного предпроектного анализа позволяет впоследствии исключить возможность получения недостоверной или искаженной информации, обеспечивая высокий уровень проектирования и реализации АС. Предпроектный анализ завершается созданием трех основных документов:
- I. "Отчет об обследовании объекта", состоящий из следующих разделов:
 - предварительная функциональная модель предприятия
 - характеристика объекта и результатов его функционирования
 - описание существующей информационной системы и ее качества
 - описание недостатков существующей информационной системы
 - обоснование необходимости совершенствования информационной системы объекта
 - цели, критерии и ограничения создания автоматизированной системы (АС)
 - функции и задачи АС
 - ожидаемые технико-экономические результаты создания АС
 - выводы и предложения
- II. "План мероприятий по переходу из текущего состояния в целевое": включает в себя этапы преобразования бизнес-процессов объекта.
- III. "Описание моделей деятельности": документально оформленные модели деятельности предприятия, отражающие текущую ситуацию "как есть" и совместно выработанную Исполнителем и Заказчиком "как должно быть".

- 18. Принципы моделирования бизнес – процессов и системы моделирования бизнес – процессов.
- **Бизнес-процесс** (деловой процесс) – последовательность выполнения деловых операций (действий), необходимых для решения поставленной задачи.
- Чем отличается UML от ADONIS? В ADONIS только моделирование, в UML возможно еще проектирование. UML — язык графического описания для объектного моделирования в области разработки программного обеспечения.
- **Основные принципы моделирования бизнес-процессов**
- При разработке и использовании на предприятиях модели бизнес-процессов необходимо учитывать шесть принципов, считающихся основными критериями качества в рамках моделирования:
 - принцип достоверности, который является неотъемлемой предпосылкой для создания модели;
 - принцип значимости – модель должна документировать только те объекты, которые имеют значение для соответствующей перспективы;
 - принцип понятности – модель может быть полезной только в том случае, если она понятна пользователю, и гарантируется достаточная степень ее интуитивного восприятия;
 - принцип сопоставимости – обеспечение использования единых правил моделирования;
 - принцип систематичной структуры – для создания системы моделей необходимо предусматривать интерфейсы, обеспечивающие ее взаимосвязь и структурированность;
 - принцип экономической эффективности – обеспечить сбалансированное соотношение между затратами на моделирование и достигнутыми результатами (Беккер и др., 2007).
- **Специализированные инструменты для описания бизнес процессов:** ADONIS, Business Studio, Орг-Мастер (БИГ-СПб), Casewise Corporate Modeler Suite, ERwin Process Modeler (старое название — BPwin), Fox Manager, PayDox AJAX-BPM, Ramus, QPR ProcessGuide , Sparx Enterprise Architect, Sybase PowerDesigner, Бизнес-инженер, ELMA и др.

- 19. Стадии разработки технического и рабочего проектов КИС.
- **Стадия технического проекта. Разрабатывается обеспечение:**
 - методическое (принципы построения, методики решения задач)
 - информационное (модели и структуры баз данных),
 - математическое (методы и алгоритмы);
 - техническое обеспечение (выбор компьютеров сетевых средств и серверов).
- **Стадия рабочего проекта. Выполняются работы:**
 - программирование системы;
 - опытное формирование баз данных (знаний);
 - комплексная отладка системы;
 - составление эксплуатационной документации.
- Чем отличается БД от баз информационной базы (ИБ)? ИБ – источники информации: каталоги, стандарты, ГОСТы и т.п. БД – это собранная конкретно для наших задач информация. База знаний (БЗ) – это особого рода база данных, разработанная для оперирования знаниями (метаданными).
-

- 20. Стадии разработки рабочих проектов КИС и её внедрения.
- **Стадия рабочего проекта. Выполняются работы:**
 - 1. программирование системы;
 - 2. опытное формирование баз данных (знаний);
 - 3. комплексная отладка системы;
 - 4. составление эксплуатационной документации.
- **Стадия внедрения. Выполняются работы:**
 - закупка и установка технических средств;
 - установка программного продукта;
 - обучение пользователей работе с системой;
 - формирование базы данных (знаний);
 - опытная эксплуатация системы;
 - выявление ошибок, накопление замечаний и рекомендаций по совершенствованию системы;
 - исправление разработчиками найденных ошибок;
 - сдача системы в эксплуатацию.
- **Ошибки при внедрении:**
 - 1. Ввод данных
 - 2. Ошибки БД
 - 3. В алгоритмах
 - 4. Программные
 -

- 21. Основные виды моделей проектирования КИС.
- **Основные виды моделей:**
- каскадная - она предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе.
- каскадная с обратной связью;
- спиральная - каждая итерация соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, оценивается качество полученных результатов и планируются работы следующей итерации
- На базе унифицированного процесса (UP - Unified Process) - в процессе разработки программного обеспечения, в котором описывается деятельность компании и определяются требования к системе — те подпроцессы и операции, которые подлежат автоматизации в разрабатываемой информационной системе.
-

- 22. Каскадная модель проектирования КИС и кризис разработки сложных программных систем.
- **Каскадная модель:**
- Анализ (Разработка ТЗ)-Проектирование (Разр.ТП)-Реализация (Разр.РП)-Внедрение-Сопровождение
- **Кризис разработки сложных программных систем:**
- проекты не сдаются в срок;
- существенно перерасходуется бюджет проектов;
- проекты не удовлетворяют заданным спецификациям;
- модификация проектов становится чрезвычайно трудоемкой и рискованной.
- **Водопадная (каскадная, последовательная) модель**
- Водопадная модель жизненного цикла (англ. waterfall model) была предложена в 1970 г. Уинстоном Ройсом. Она предусматривает последовательное выполнение всех этапов проекта в строго фиксированном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе. Требования, определенные на стадии формирования требований, строго документируются в виде технического задания и фиксируются на все время разработки проекта. Каждая стадия завершается выпуском полного комплекта документации, достаточной для того, чтобы разработка могла быть продолжена другой командой разработчиков.
-

Каскадная модель

Разработка ТЗ

Анализ

Проектирование

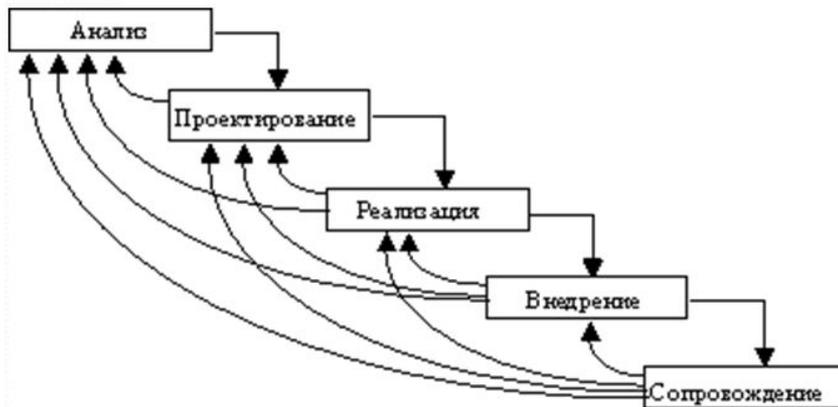
Реализация

Внедрение

Сопровождение

Разработка
РП

Разработка ТП



Реальное
проектирование

- **Этапы проекта в соответствии с каскадной моделью:**

- Формирование требований;
- Проектирование;
- Реализация;
- Тестирование;
- Внедрение;
- Эксплуатация и сопровождение.

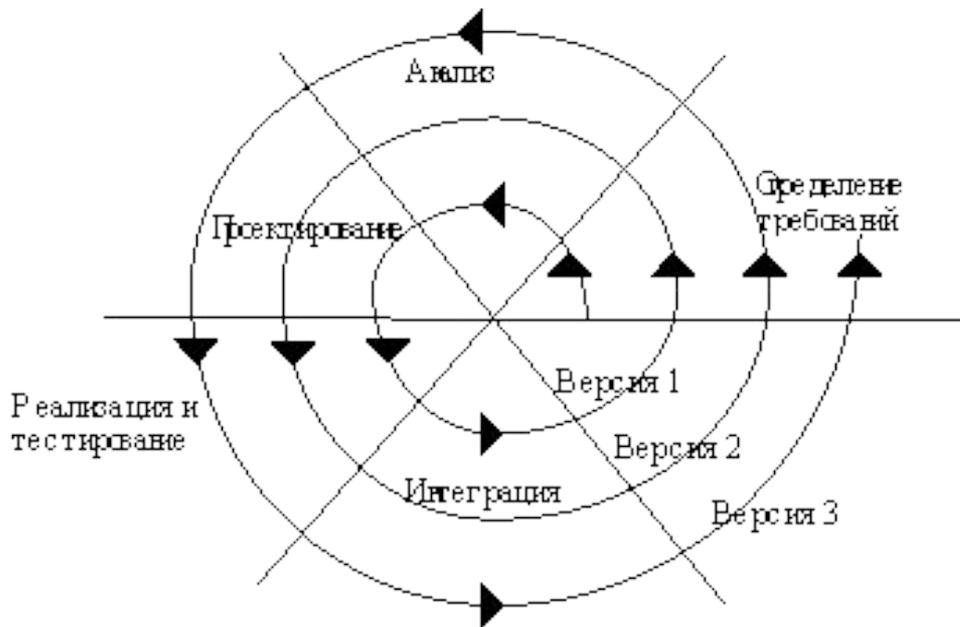
- **Преимущества:**

- - Полная и согласованная документация на каждом этапе;
- - Легко определить сроки и затраты на проект.

- **Недостатки:**

- В водопадной модели переход от одной фазы проекта к другой предполагает полную корректность результата (выхода) предыдущей фазы. Однако неточность какого-либо требования или некорректная его интерпретация в результате приводит к тому, что приходится «откатываться» к ранней фазе проекта и требуемая переработка не просто выбивает проектную команду из графика, но приводит часто к качественному росту затрат и, не исключено, к прекращению проекта в той форме, в которой он изначально задумывался. По мнению современных специалистов, основное заблуждение авторов водопадной модели состоит в предположениях, что проект проходит через весь процесс один раз, спроектированная архитектура хороша и проста в использовании, проект осуществления разумен, а ошибки в реализации легко устраняются по мере тестирования. Эта модель исходит из того, что все ошибки будут сосредоточены в реализации, а потому их устранение происходит равномерно во время тестирования компонентов и системы. Таким образом, водопадная модель для крупных проектов мало реалистична и может быть эффективно использована только для создания небольших систем

- 23. Кризис разработки сложных программных систем и переход к спиральной модели проектирования КИС.
- проекты не сдаются в срок;
- существенно перерасходуется бюджет проектов;
- проекты не удовлетворяют заданным спецификациям;
- модификация проектов становится чрезвычайно трудоемкой и рискованной.
-
- На каждом витке спирали выполняется создание очередной версии продукта, уточняются требования проекта, определяется его качество и планируются работы следующего витка.

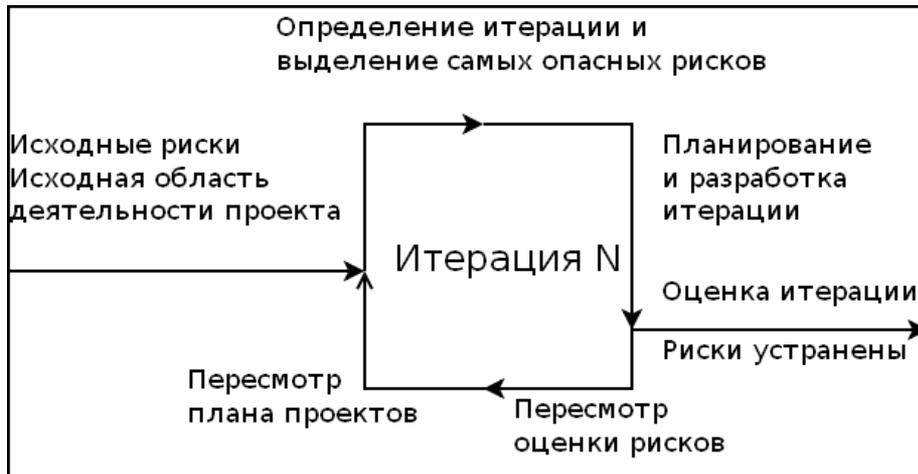


Спиральная модель (англ. spiral model) была разработана в середине 1980-х годов Барри Боэмом. Она основана на классическом цикле Деминга PDCA (plan-do-check-act). При использовании этой модели ПО создается в несколько итераций (витков спирали) методом прототипирования. Каждая итерация соответствует созданию фрагмента или версии ПО, на ней уточняются цели и характеристики проекта, оценивается качество полученных результатов и планируются работы следующей

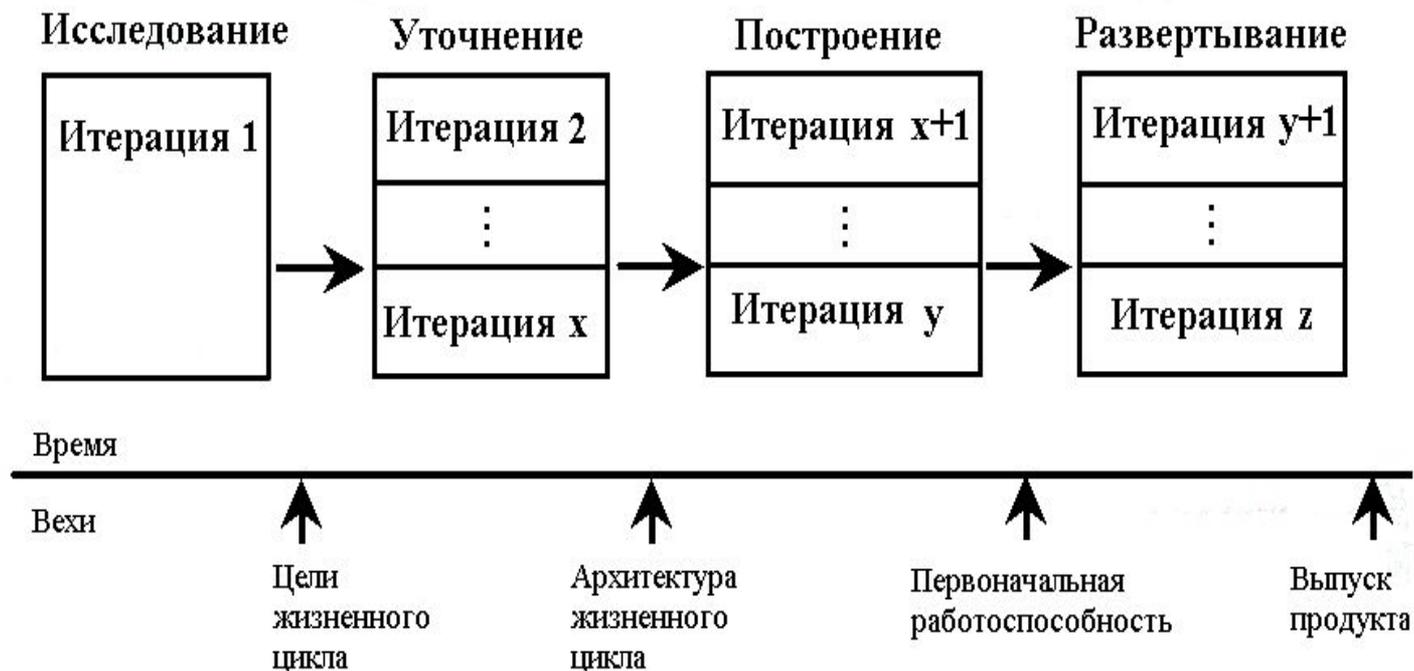
- **На каждой итерации оцениваются:**
- риск превышения сроков и стоимости проекта;
- необходимость выполнения ещё одной итерации;
- степень полноты и точности понимания требований к системе;
- целесообразность прекращения проекта.
- Отличительной особенностью спиральной модели является специальное внимание, уделяемое рискам, влияющим на организацию жизненного цикла, и контрольным точкам. Боэм формулирует **10 наиболее распространённых (по приоритетам) рисков:**
- Дефицит специалистов.
- Нереалистичные сроки и бюджет.
- Реализация несоответствующей функциональности.
- Разработка неправильного пользовательского интерфейса.
- Перфекционизм, ненужная оптимизация и оттачивание деталей.
- Непрерывающийся поток изменений.
- Нехватка информации о внешних компонентах, определяющих окружение системы или вовлеченных в интеграцию.
- Недостатки в работах, выполняемых внешними (по отношению к проекту) ресурсами.
- Недостаточная производительность получаемой системы.
- Разрыв в квалификации специалистов разных областей.
-

- 24. Методология проектирования КИС на базе унифицированного процесса – RUP.
- Rational Unified Process (RUP) — методология разработки программного обеспечения, созданная компанией Rational Software.
- **Методология на базе унифицированного процесса - Rational Unified Process - RUP**
- использование языка UML;
- большой набор справочных пособий и шаблонов;
- наличие инструментария (Rational Suite) для эффективного проектирования сложных программных систем.
- **В основе RUP лежат следующие принципы:**
- Ранняя идентификация и непрерывное (до окончания проекта) устранение основных рисков.
- Концентрация на выполнении требований заказчиков к исполняемой программе (анализ и построение модели прецедентов (вариантов использования)).
- Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.
- Компонентная архитектура, реализуемая и тестируемая на ранних стадиях проекта.
- Постоянное обеспечение качества на всех этапах разработки проекта (продукта).
- Работа над проектом в сплочённой команде, ключевая роль в которой принадлежит архитекторам.

- 25. Процесс разработки подсистем КИС в терминологии RUP.
- **Итеративный и инкрементный процесс разработки подсистем КИС**

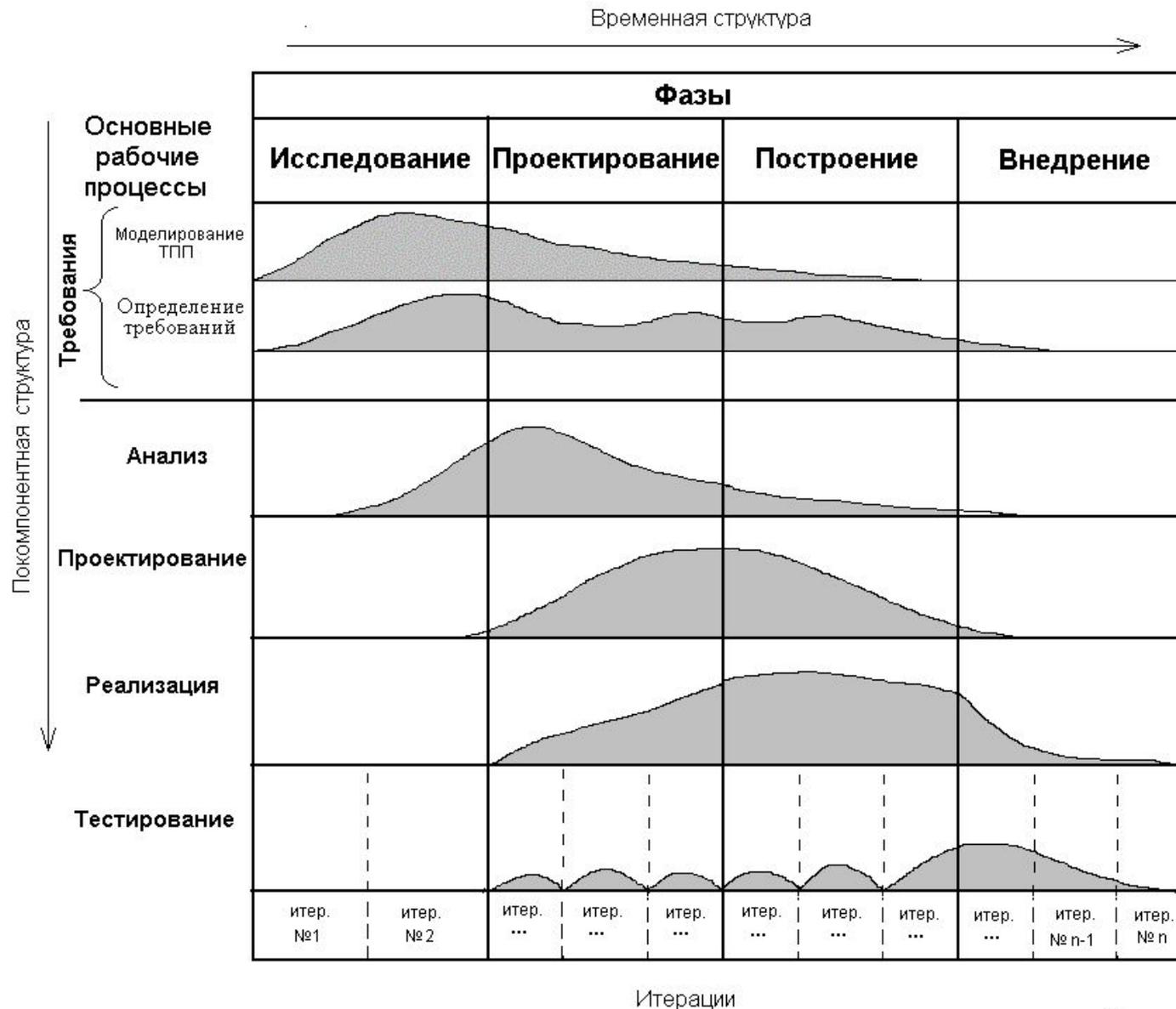


- 26. RUP - фазы и вехи жизненного цикла разработки системы.
- **Фазы и вехи жизненного цикла разработки системы**
- Исследование - Уточнение – Построение – Развертывание (на каждом шаге несколько итераций)
- Время-> Цели ЖЦ – Архитектура ЖЦ – Первоначальная работоспособность – Выпуск продукта
- **Вехи** – отметка пройденного, конец или начало какого-то этапа. Успешное выполнение фазы разработки означает достижение вехи архитектуры жизненного цикла.



- 27. RUP - структура унифицированного процесса проектирования.
- **Структура УП.** RUP использует итеративную модель разработки. В конце каждой итерации (в идеале продолжающейся от 2 до 6 недель) проектная команда должна достичь запланированных на данную итерацию целей, создать или доработать проектные артефакты и получить промежуточную, но функциональную версию конечного продукта. Итеративная разработка позволяет быстро реагировать на меняющиеся требования, обнаруживать и устранять риски на ранних стадиях проекта, а также эффективно контролировать качество создаваемого продукта.
- Полный жизненный цикл разработки продукта состоит из четырех фаз, каждая из которых включает в себя одну или несколько итераций.
- **Исследование:**
 - - формируются видение и границы проекта
 - - создается экономическое обоснование
 - - определяются основные требования, ограничения и функциональность продукта
 - - создаются базовая версия модели прецедентов
 - - оцениваются риски
 - При завершении начальной фазы оцениваются достижения вехи целей жизненного цикла, которое предполагает соглашение заинтересованных сторон о продолжении проекта.
- В фазе **Уточнения** производится анализ предметной области и построение исполняемой архитектуры.
-

- В фазе **Внедрения** создается финальная версия продукта и передается от разработчика к заказчику. Это включает в себя программу бета- тестирования, обучение пользователей, а также определение качества продукта. В случае, если качество не соответствует ожиданиям пользователей и критерием, уст.

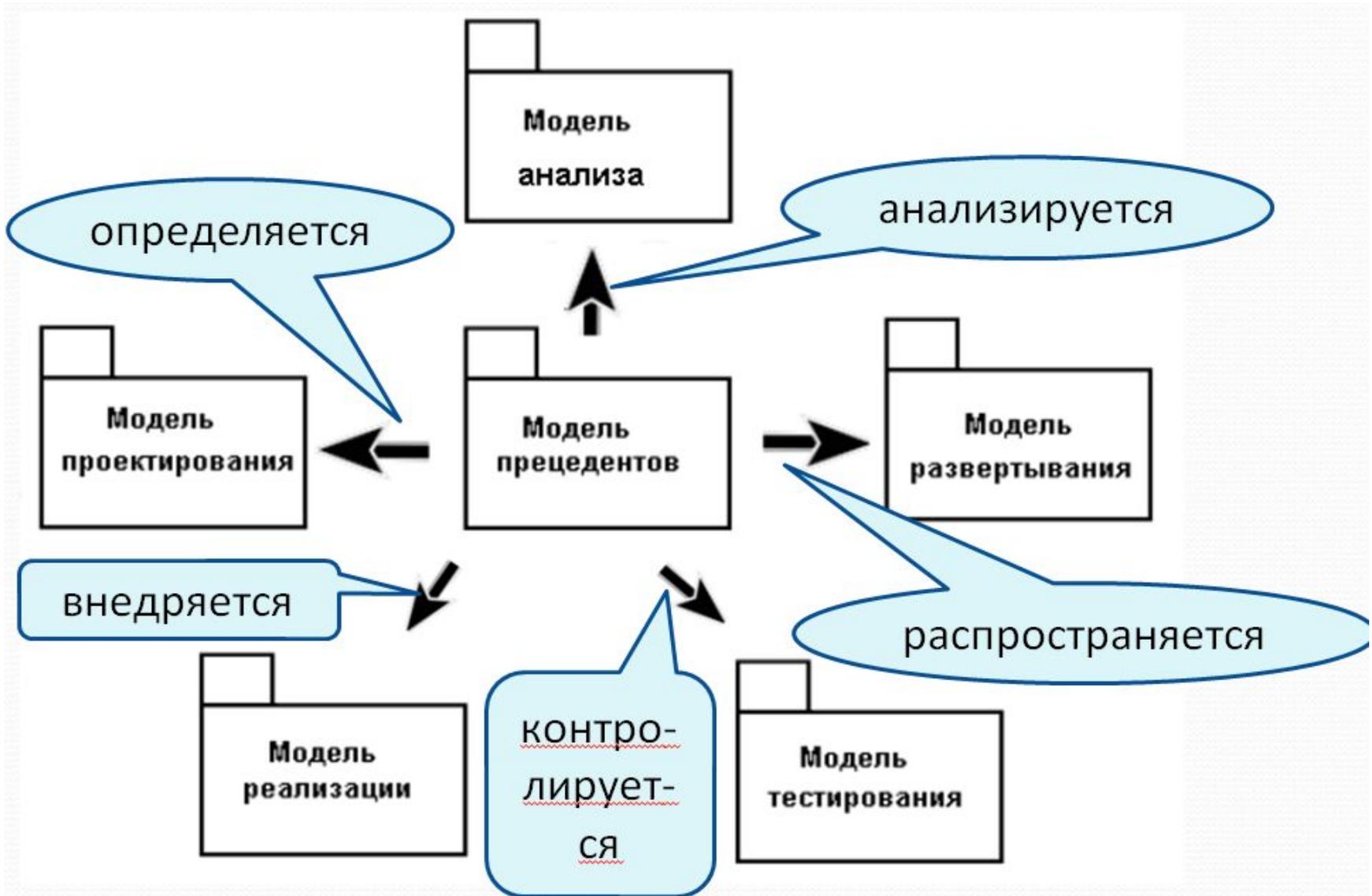


28. RUP - этапы моделирования при проектировании КИС.

Этапы моделирования при проектировании КИС

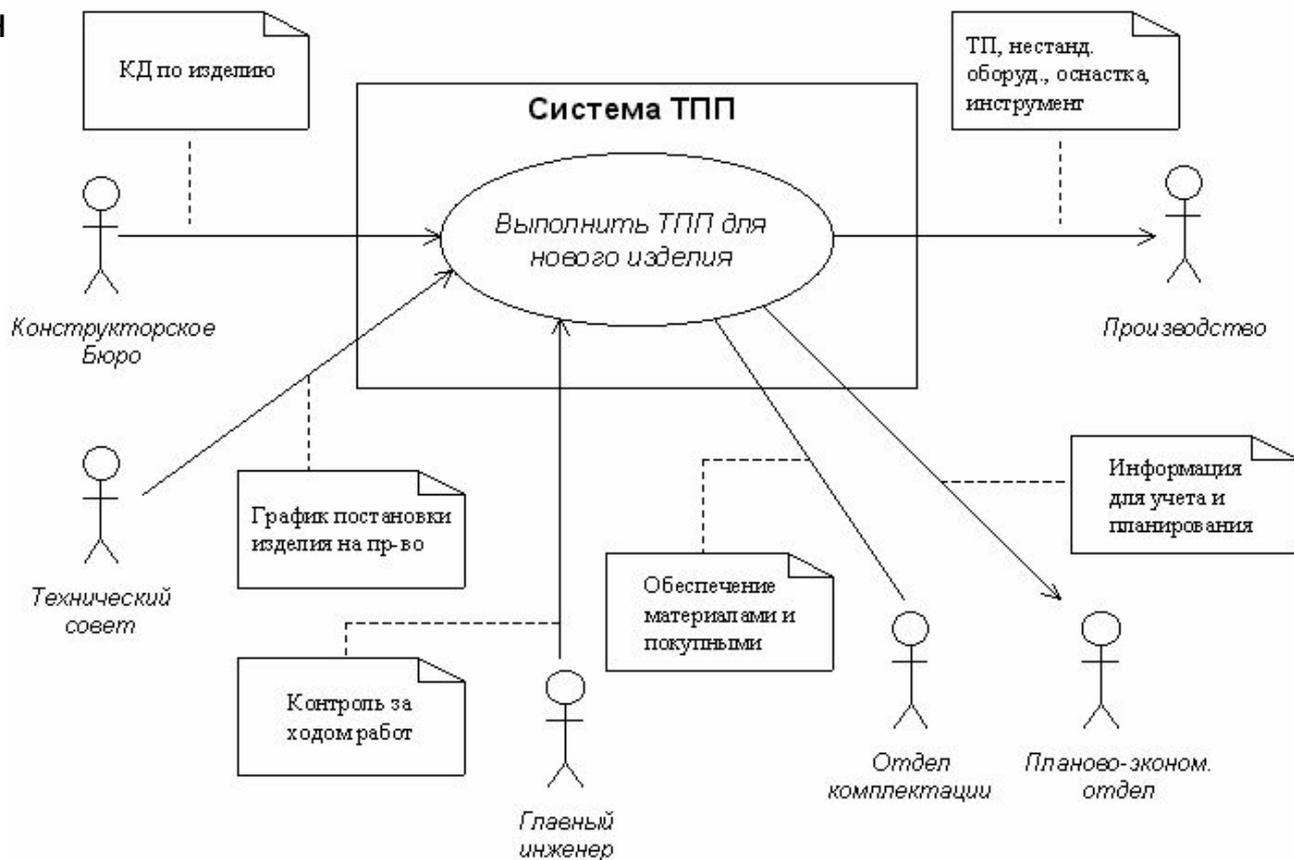


29. RUP - основные модели для разработки КИС
Основные модели для разработки КИС

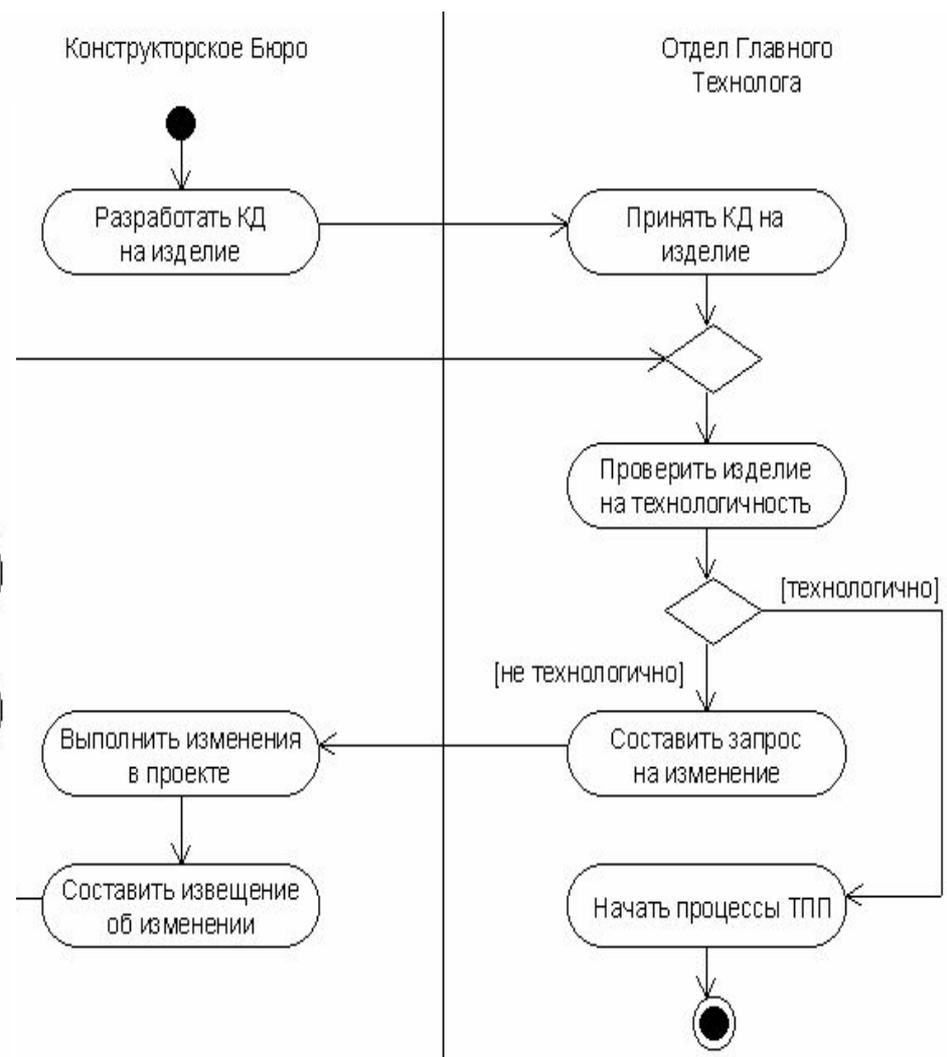
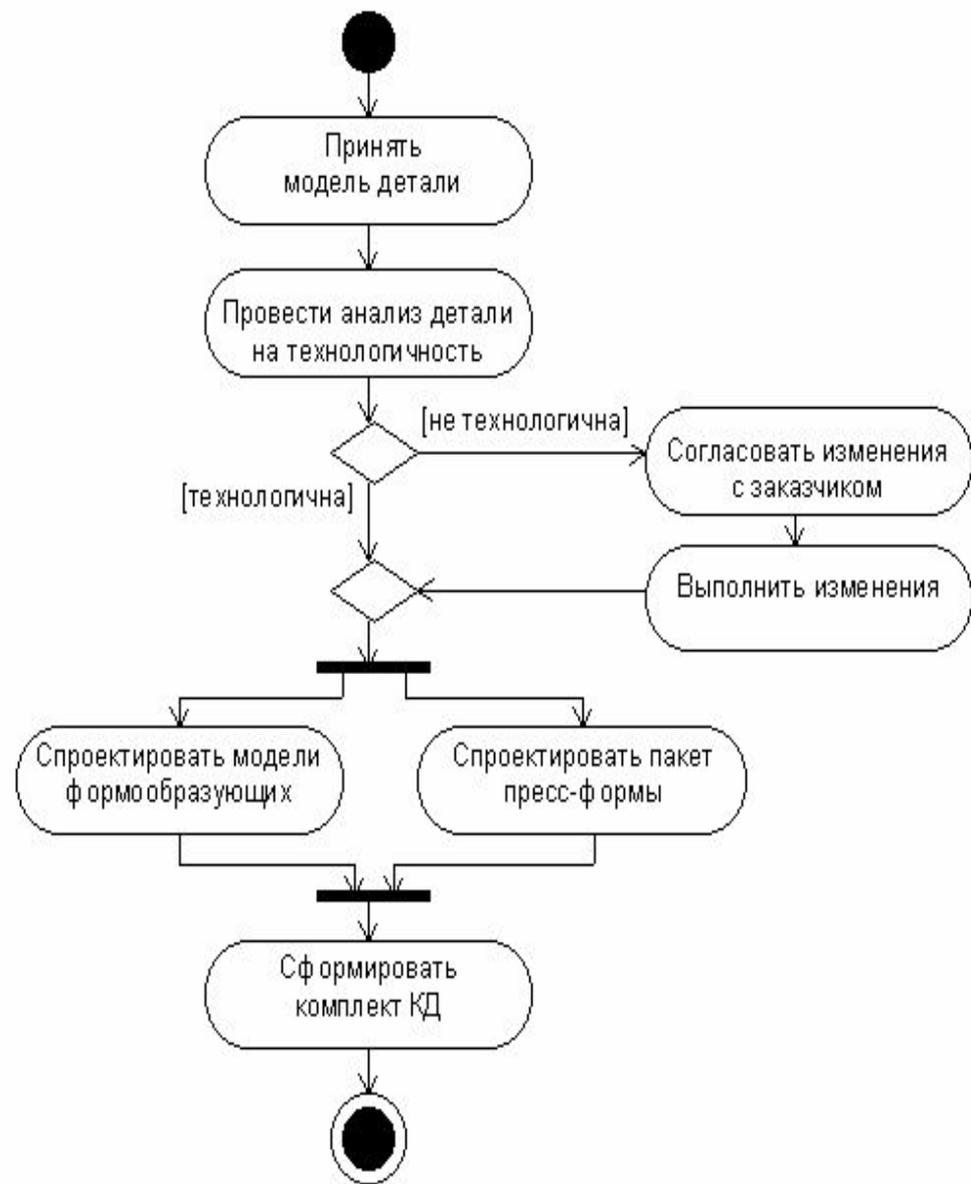


- 30. UML - назначение диаграммы прецедентов.
- **Диаграмма прецедентов** – описание набора, последовательность действий, которые выполняются системой и имеют значение для конкретного действующего лица.
- Графический язык UML включает 8 типов диаграмм, описывающих бизнес- процессы или сложную информационную систему с различных точек зрения Диаграммы: **1- Прецедентов-функциональное** назначение системы. Определяет общие границы модельной области. Основные эл-ты диаграммы-прецеденты(ф-ии) и внешние действующие субъекты, актеры; **2-Классов**- для построения структурированной статической модели предметной области. Класс изображ-ся в виде прям-ка, раздел-го на 3 секции: в 1 имя класса, во 2 перечень атрибутов, в 3 перечень операций; **3- Состояний**-для отображения поведения системы или ее эл-тов; **4-Деятельности** - _напоминает обычные алгоритмы и рассматривается как детализация диаграмм прецедентов; **5-Последовательности** - для моделирования временного взаимодействия объектов предметной области; **6-Кооперации** - для моделирования взаимодействия объектов предметной области, но акцент не на хронологической последовательности, а на структурном взаимодействии участников; **7-Компонентов**; **8-Развертывания** - отображают структуру и состав программных компонентов разрабатываемой системы.

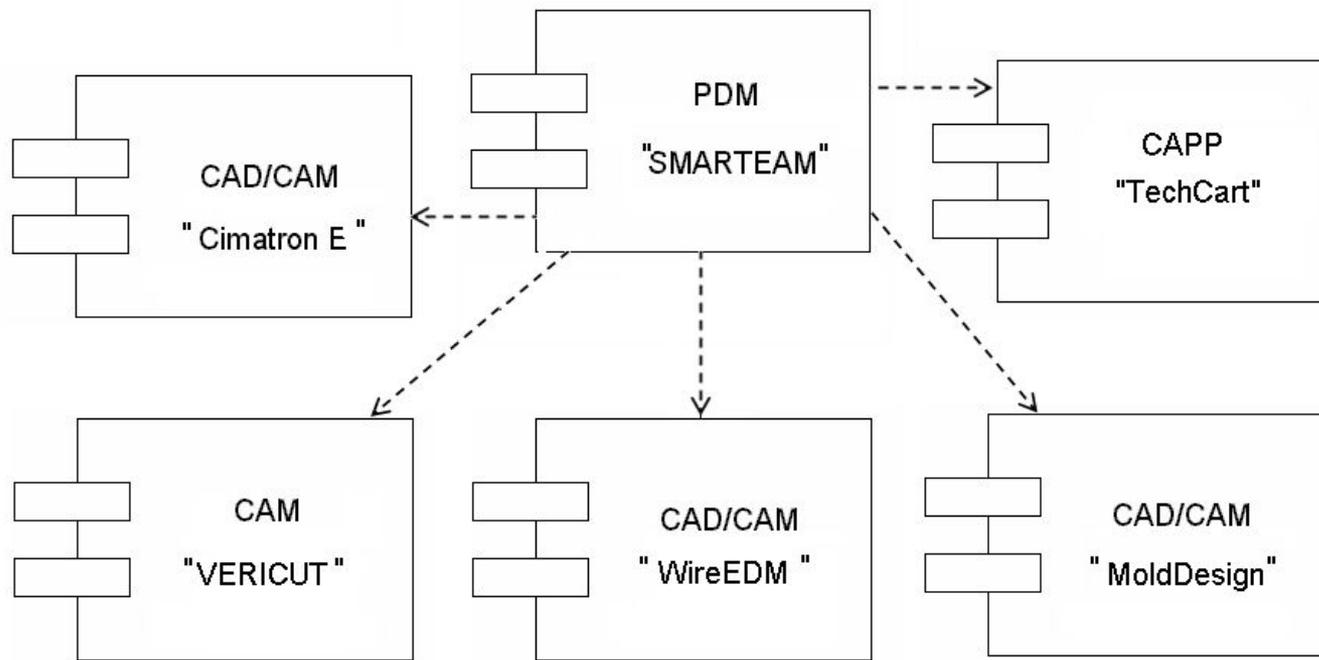
- **Диаграмма прецедентов** (англ. use case diagram, диаграмма вариантов использования) — диаграмма, на которой отражены отношения, существующие между актёрами и прецедентами.
- Основная задача — представлять собой единое средство, дающее возможность заказчику, конечному пользователю и разработчику совместно обсуждать функциональность и поведение системы.
- *При работе с вариантами использования важно помнить несколько простых правил:*
- каждый прецедент относится как минимум к одному действующему лицу;
- каждый прецедент имеет инициатора;
- каждый прецедент приводит к соответствующему результату (результату с «бизнес-зн



- 31. UML - назначение диаграммы деятельности.
- **Диаграмма деятельности** – акцентируют внимание на последовательности выполнения определенных действий, которые приводят к определенному результату. Каждое состояние соответствует элементарным операциям, при завершении которой осуществляется переход в следующее состояние. Графически диаграммы деятельности представляют в форме графа, вершинами которого являются состояния деятельности, а дугами – переходы из одного состояния в другое. ДД – одна из самых важных, т.к. фиксируется операции подразделения.
- **Диаграмма деятельности** англ. activity diagram — диаграмма, на которой показано разложение некоторой деятельности на её составные части. Под деятельностью англ. activity понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.
- Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.



- 32. UML - назначение диаграмм компонентов и развертывания.
- **Диаграмма компонентов** – показывает, какие компоненты есть в данной системе и какие между ними существуют зависимости. Компонентами системы мы называем отдельные программные блоки, из которых состоит вся система. Понимание зависимостей между компонентами дает возможность отслеживать на модели результаты измерений в отдельных компонентах. Помимо того, в этом представлении модели иногда указывается, с какими классами и элементами связан конкретный компонент.
- Диаграмма компонентов, Component diagram — статическая структурная диаграмма, показывает разбиение программной системы на структурные компоненты и связи (зависимости) между компонентами. В качестве физических компонентов могут выступать файлы, библиотеки, модули, исполняемые файлы, пакеты и т. п.
- Компоненты связываются через зависимости, когда соединяется требуемый интерфейс одного компонента с имеющимся интерфейсом другого компонента. Таким образом иллюстрируются отношения клиент-источник между двумя компонентами.
- Зависимость показывает, что один компонент предоставляет сервис, необходимый другому компоненту. Зависимость изображается стрелкой от интерфейса или порта клиента к импортируемому интерфейсу.
- Когда диаграмма компонентов используется, чтобы показать внутреннюю структуру компонентов, предоставляемый и требуемый интерфейсы составного компонента могут делегироваться в соответствующие интерфейсы внутренних компонентов.
- Делегация показывается связь внешнего контракта компонента с внутренней реализацией этого поведения внутренними компонентами.



- **Диаграмма развертывания** – отражает расположение работающих компонентов на узлах. Узел- это ресурс, используемый во время выполнения программы. Это представление служит для изображения расположения ресурсов и их расположения.
- Диаграмма развертывания, Deployment diagram в UML моделирует физическое развертывание артефактов на узлах. Например, чтобы описать веб-сайт диаграмма развертывания должна показывать, какие аппаратные компоненты ("узлы") существуют (например, веб-сервер, сервер базы данных, сервер приложения), какие программные компоненты ("артефакты") работают на каждом узле (например, веб-приложение, база данных), и как различные части этого комплекса соединяются друг с другом (например, JDBC, REST, RMI).
- Узлы представляются как прямоугольные параллелепипеды с артефактами, расположенными в них, изображенными в виде прямоугольников. Узлы могут иметь подузлы, которые представляются как вложенные прямоугольные параллелепипеды. Один узел диаграммы развертывания может концептуально представлять множество физических узлов, таких как кластер серверов баз данных.
- **Существует два типа узлов:**
 - - Узел устройства
 - - Узел среды выполнения
- Узлы устройств - это физические вычислительные ресурсы со своей памятью и сервисами для выполнения программного обеспечения, такие как обычные ПК, мобильные телефоны. Узел среды выполнения - это программный вычислительный ресурс, который работает внутри внешнего узла и который предоставляет собой сервис, выполняющий другие исполняемые программные элементы.



