

Модели процесса разработки

Лекция 3

Модели процесса разработки

- Наиболее интересной фазой жизненного цикла ПО с точки зрения технологии программирования является фаза разработки
- Особенности применяемых методов разработки описываются с помощью *моделей процесса разработки ПО*

Модели процесса разработки

- Модель процесса разработки ПО выделяет конкретные наборы видов деятельности, артефактов, ролей и их взаимосвязи, а также дает рекомендации по организации процесса в целом

Выбор модели разработки

- Реальный процесс разработки обычно жестко не увязывается с какой-либо одной моделью, хотя одна из них может быть ведущей
- Выбор модели определяется:
 - объемом и сложностью проекта;
 - количеством и качеством команды разработчиков
 - квалификацией заказчика, его способностью обеспечить достаточно четкую постановку задачи

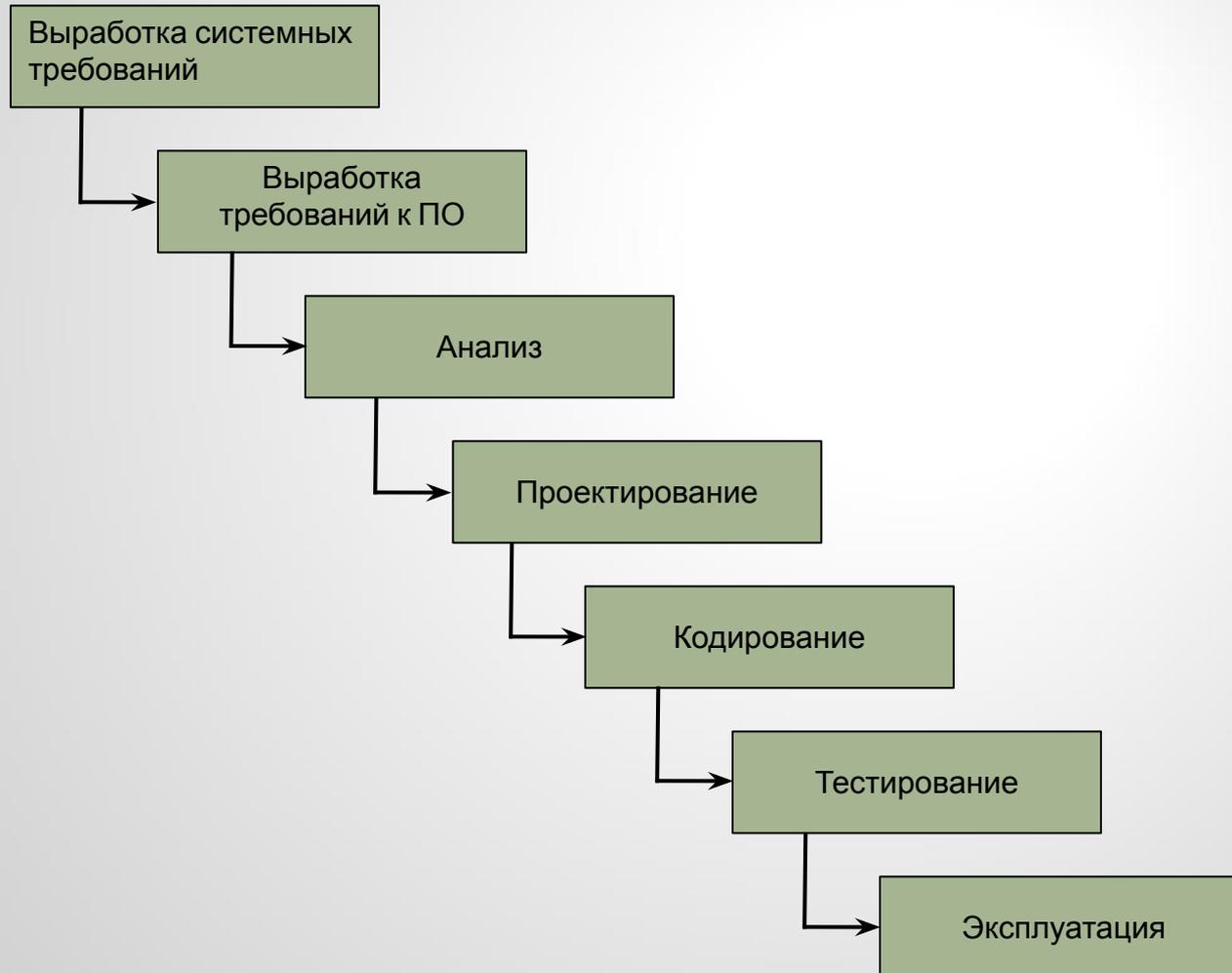
Каскадная модель

- ⦿ Наиболее широко известной и применяемой долгое время оставалась так называемая *каскадная или водопадная (waterfall) модель жизненного цикла*
- ⦿ Впервые четко сформулирована в 1970 году Уильямом Ройсом (**W.W.Rouse**) и затем закреплена в стандартах Министерства обороны США

Каскадная модель

- Предполагает строго последовательное поэтапное выполнение различных видов деятельности с четким определением границ между этапами
- Набор документов, созданный на предыдущем этапе, передается в качестве входных данных для следующего этапа

Каскадная модель



Характеристика модели

- ◎ Достоинства модели:
 - упорядоченность процесса разработки
 - возможность его строгого планирования во времени
- ◎ Недостатки модели:
 - необходимость точной и полной формулировки требований к ПС перед началом разработки
 - невозможность изменения решений, принятых на предыдущих этапах
 - результаты проекта становятся доступны заказчику только по завершении работ

Итеративные модели

- ⦿ Итеративный подход – это выполнение работ параллельно с непрерывным анализом полученных результатов и корректировкой предыдущих этапов
- ⦿ Проект при этом подходе в каждой фазе развития проходит повторяющийся цикл: Планирование — Реализация — Проверка — Оценка

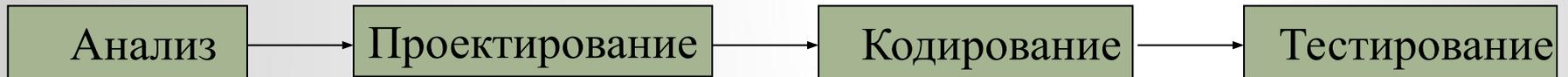
Инкрементная модель

- Предусматривает дробление продукта на относительно независимые составляющие, которые разрабатываются и вводятся в эксплуатацию по отдельности

Инкрементная модель

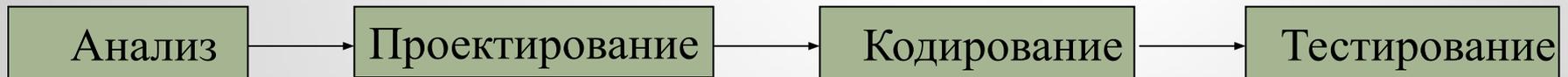
1-й функциональный блок

Поставка 1-го блока



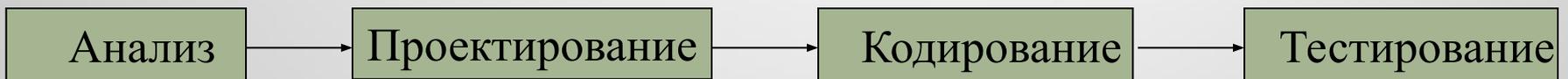
2-й функциональный блок

Поставка 2-го блока



3-й функциональный блок

Поставка 3-го блока



ДОСТОИНСТВО МОДЕЛИ

- ⦿ Достоинством данной модели по сравнению с каскадной является возможность передать заказчику работающий прототип системы до полного завершения процесса разработки

Недостатки модели

- Деление на функциональные блоки в целом замедляет процесс, так как возникает необходимость обеспечения их взаимодействия
- Для многих решений этот метод неприменим, поскольку из них нельзя вычленить отдельные составляющие, которые могут быть поставлены и функционировать независимо

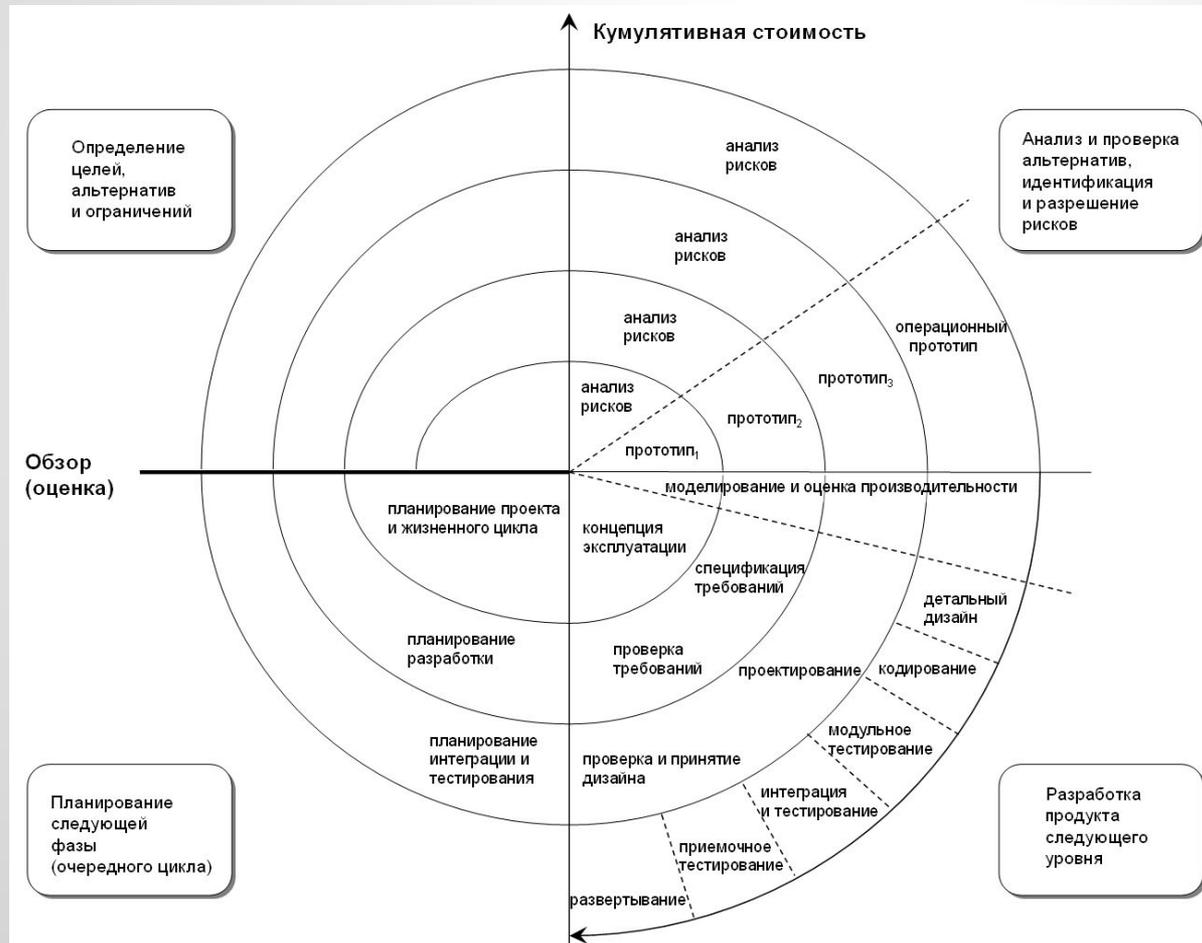
Недостатки модели

- Существенно усложняется управление проектом в связи с усложнением задач по координированию работ над отдельными составляющими системы
- Увеличивается стоимость внесения изменений в готовые компоненты, которые уже установлены и работают у заказчика

Спиральная модель

- Предложена в 1988 г. Барри Боэмом (**Barry W. Boehm**) и является классическим примером реализации эволюционной стратегии.
- Модель определяет четыре действия:
 - планирование,
 - анализ рисков,
 - конструирование,
 - оценивание

Спиральная модель



Основные действия модели

- ◎ *Планирование* заключается в определении целей очередной итерации процесса разработки, выборе вариантов решения и оценки ограничений
- ◎ *Анализ рисков* – анализ вариантов решения и оценка связанных с ними рисков, т.е. возможностей получения неудовлетворительных результатов

Основные действия модели

- ⦿ *Конструирование* – это основное действие, заключающееся в создании следующей версии ПО
- ⦿ *Оценивание* – оценка заказчиком качества очередной версии ПО, внесение им предложений по модификации продукта, корректировка требований

Риски

- Отличительной особенностью спиральной модели является специальное внимание рискам
- *Риском* называется возможность получения неудовлетворительного результата в том или ином виде деятельности

Риски

- ◎ При разработке ПО неудовлетворительным результатом может быть:
 - превышение бюджета,
 - низкая надежность продукта,
 - неправильное функционирование и пр.

Итерации и риски

- С каждой итерацией связан некоторые начальные риски, которые уменьшаются при успешном завершении итерации
- Началу следующей итерации предшествует пересмотр и новая оценка рисков

Показатель риска

- Для ранжирования рисков по степени значимости используют величину *показатель риска RE (Risk Exposure)*

$$RE=P*L,$$

где **P** – вероятность неудовлетворительного результата, **L** – потеря (в 10 или 100-балльной шкале) при получении неудовлетворительного результата

Управление рисками

- ⦿ Включает 6 действий:
 - идентификация риска – выявление риска в проекте;
 - анализ риска – оценка вероятности и величины потери;
 - ранжирование рисков – упорядочение по степени влияния;
 - планирование управления рисками – подготовка к работе с каждым риском;

Управление рисками

- разрешение риска – устранение риска;
 - наблюдение рисков – отслеживание динамики изменения рисков, выполнение корректирующих действий
- ◎ Бозэм формулирует десять наиболее распространённых (по приоритетам) рисков

Список рисков по Боэму

- дефицит специалистов;
- нереалистичные сроки и бюджет;
- реализация несоответствующей функциональности;
- разработка неправильного пользовательского интерфейса;
- «золотая сервировка», ненужная оптимизация и оттачивание деталей;
- непрекращающийся поток изменений;
- нехватка информации о внешних компонентах;

Список рисков по Боэму

- недостатки в работах, выполняемых внешними ресурсами;
 - недостаточная производительность получаемой системы;
 - «разрыв» в квалификации специалистов разных областей знаний
- ◎ Большая часть этих рисков связана с организационными и процессными аспектами взаимодействия специалистов в проектной команде

Характеристика модели

- ◎ Достоинства спиральной модели:
 - данная модель отображает процесс разработки ПО в наиболее реальном виде;
 - позволяет явно учитывать риски на каждом витке эволюционного процесса и принимать различные управленческие решения вплоть до прекращения работ

Характеристика модели

- ◎ Недостатки спиральной модели:
 - повышенные требования к заказчику;
 - трудности контроля и управления временем разработки

RUP-процесс разработки ПС

- ◎ RUP является развитием спиральной модели и представляет процесс разработки ПО в виде *эволюционно-инкрементного цикла*
- ◎ Эволюционная составляющая цикла основывается на дополнении требований в ходе работы
- ◎ Инкрементная составляющая – на планомерном приращении реализации требований

Этапы разработки

- ◎ RUP выделяет в процессе разработки 4 этапа:
 - начало (Inception)
 - развитие (Elaboration)
 - конструирование (Construction)
 - внедрение (Transition)

Этапы и итерации

- ⦿ В рамках каждого из этапов возможно проведение нескольких итераций
- ⦿ Итерация – это полный цикл разработки, имеющий своим результатом промежуточный продукт
- ⦿ На каждой итерации промежуточный продукт инкрементно усложняется, постепенно превращаясь в конечную систему

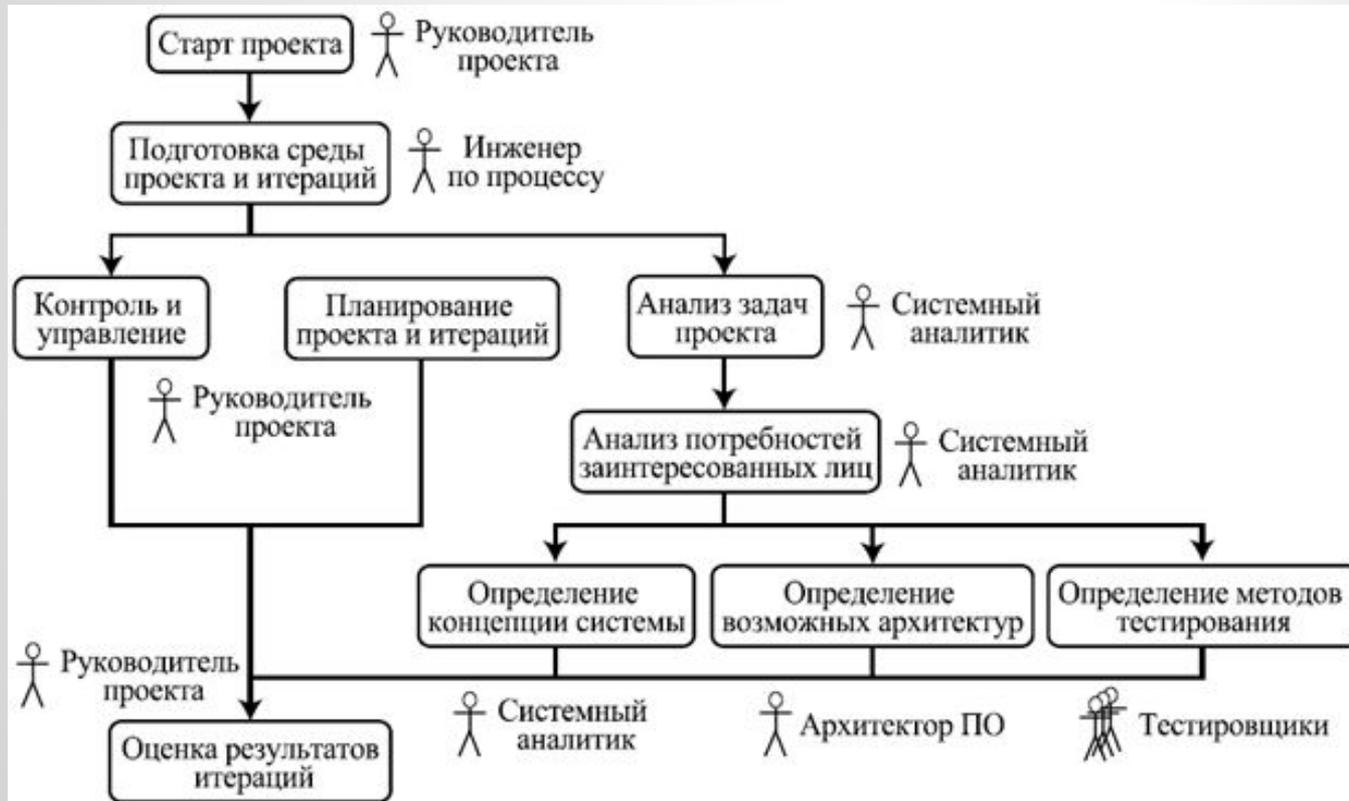
Контрольные вехи

- ◎ Каждый этап и итерация завершаются *контрольной вехой*
- ◎ Контрольная веха – это проверка состояния разработки с целью определения степени достижения ключевых целей

Этап начала проекта (Inception)

- Основная цель этой этапа — достичь компромисса между всеми заинтересованными лицами относительно задач проекта и выделяемых на него ресурсов
- Определяются основные цели проекта, руководитель и бюджет, основные средства выполнения — технологии, инструменты, ключевые исполнители

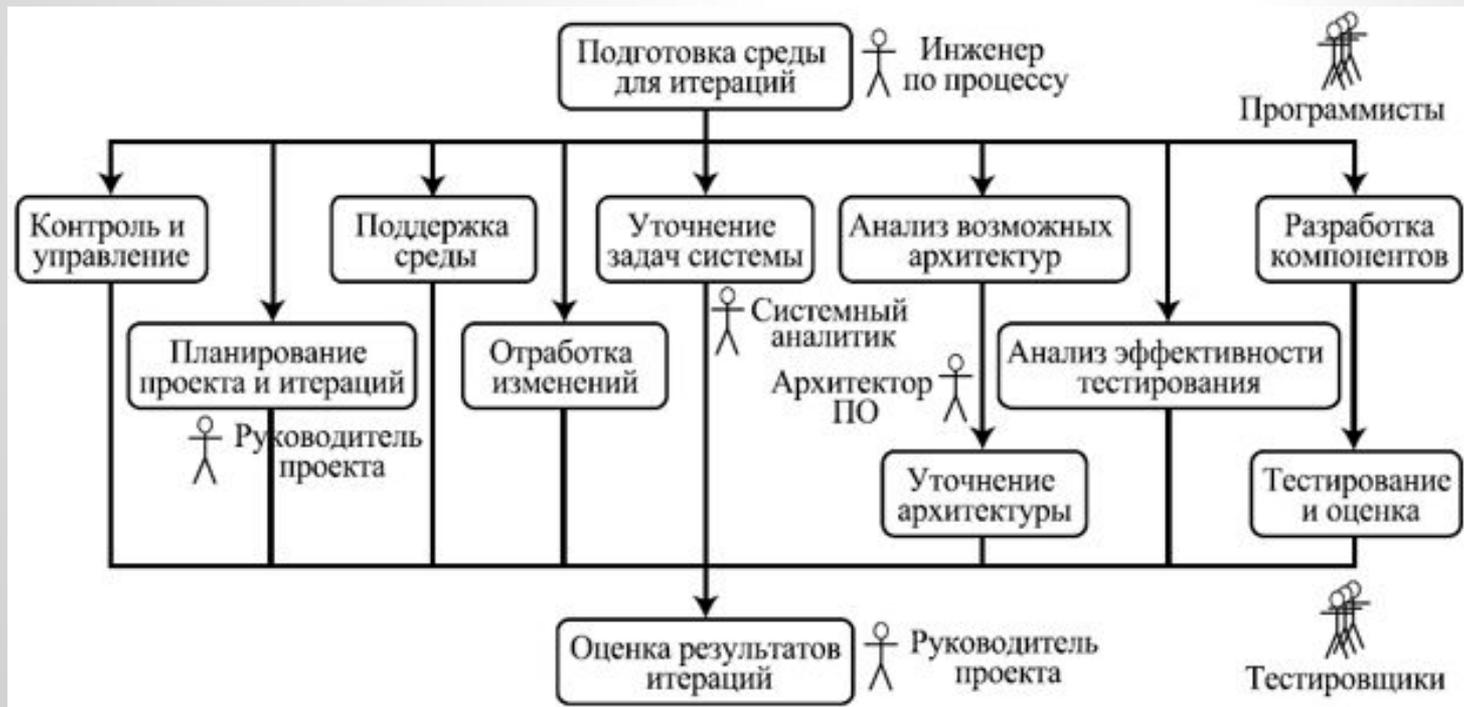
Ход работ для этапа Insertion



Этап развития (Elaboration)

- Основная цель данного этапа — исходя из основных требований разработать стабильную базовую архитектуру продукта
- Эта архитектура в дальнейшем используется как основа разработки системы

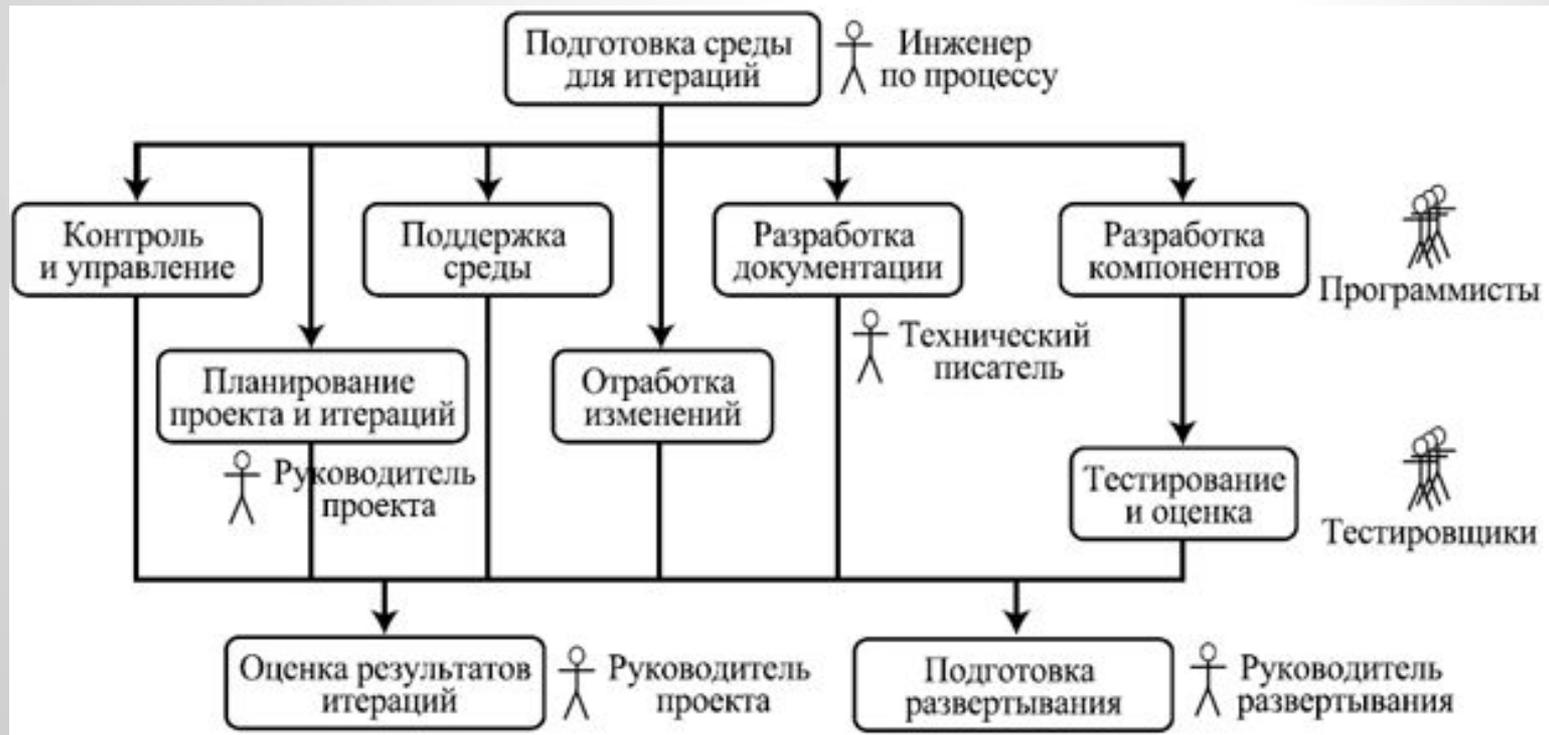
Ход работ для этапа Elaboration



Этап конструирования (Construction)

- ⦿ Основная цель данного этапа — детальное прояснение требований и разработка системы, удовлетворяющей им, на основе спроектированной ранее архитектуры
- ⦿ В результате должна получиться система, реализующая все выделенные варианты использования

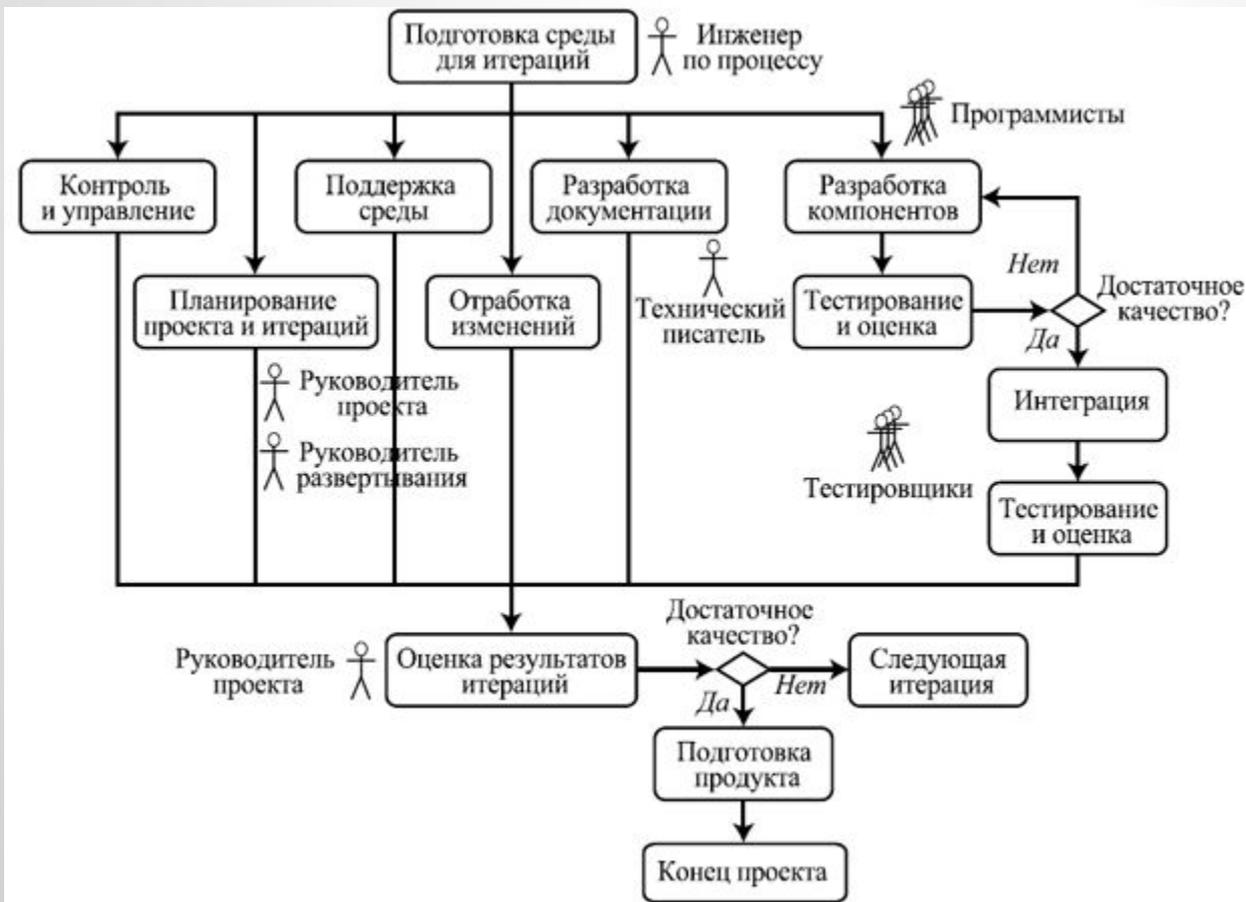
Ход работ для этапа Construction



Этап перехода (Transition)

- ◎ Цель данного этапа — сделать систему полностью доступной конечным пользователям
- ◎ Здесь происходит развертывание системы в ее рабочей среде, бета-тестирование, подгонка мелких деталей под нужды пользователей.

Ход работ для этапа Transition



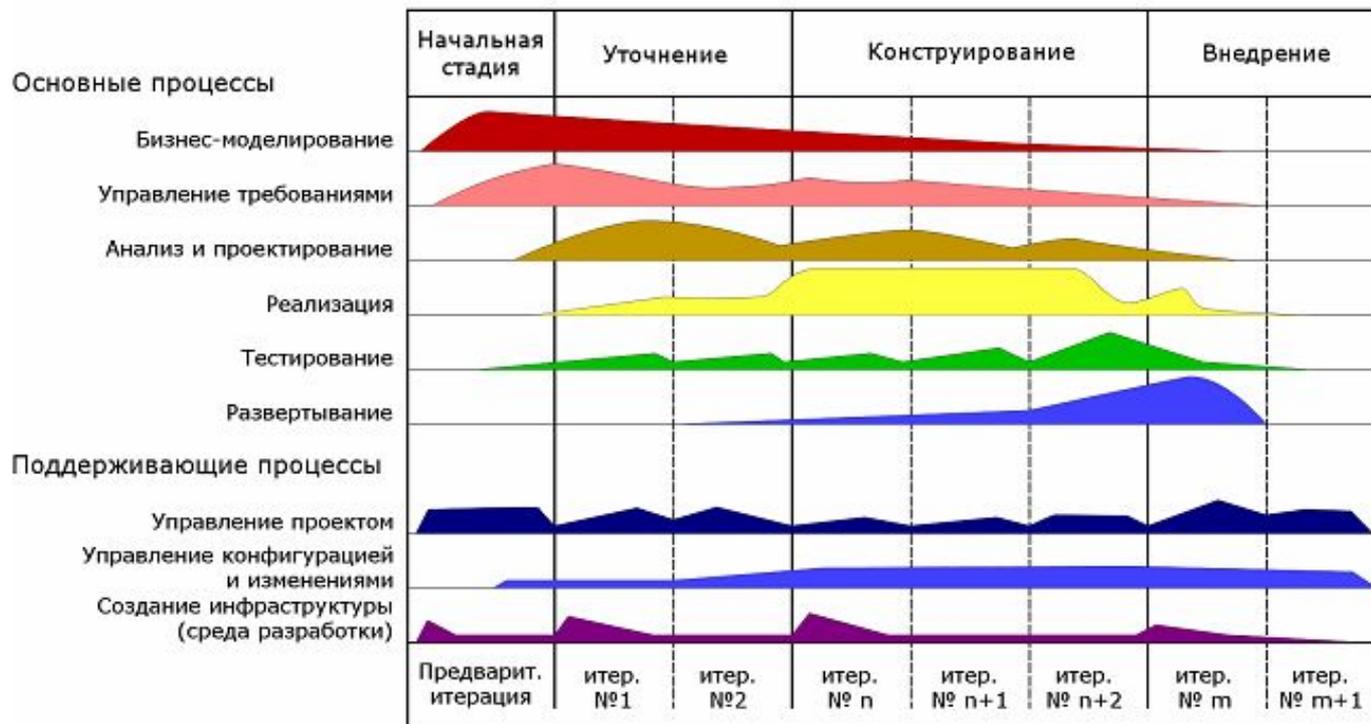
Рабочие потоки

- ◎ Каждая итерация включает несколько *рабочих потоков*:
 - моделирование предметной области (Business Modeling);
 - определение требований (Requirements);
 - анализ и проектирование (Analysis and Design);
 - реализация (Implementation);
 - тестирование (Test);
 - развертывание (Deployment);

Распределение объемов работ

Рабочие процессы

Стадии



Итерации

Моделирование предметной области

- ⦿ В результате моделирования предметной области должна появиться ее модель в виде набора диаграмм классов (объектов предметной области) и деятельностей (представляющих бизнес-операции и бизнес-процессы)
- ⦿ Модель предметной области служит основой модели проектирования

Определение требований

- ◎ Задачи этого рабочего потока:
 - понять, что должна делать система, и убедиться во взаимопонимании по этому поводу между заинтересованными лицами;
 - определить границы системы;
 - создать основу для планирования проекта и оценок затрат ресурсов в нем.
- ◎ Требования принято фиксировать в виде модели вариантов использования

Анализ и проектирование

- ◎ Задачи этого рабочего потока:
 - разработка архитектуры системы на основе требований
 - убедиться, что данная архитектура может быть основой работающей системы в контексте ее будущего использования

Анализ и проектирование

- ⦿ В результате проектирования должна появиться модель проектирования, включающая:
 - диаграммы классов системы,
 - диаграммы ее компонентов,
 - диаграммы взаимодействий между объектами в ходе реализации вариантов использования,
 - диаграммы состояний для отдельных объектов,
 - диаграммы развертывания

Реализация

- ◎ Задачи рабочего потока:
 - определить структуру исходного кода системы,
 - разработать код ее компонентов
 - протестировать компоненты,
 - интегрировать систему в работающее целое

Тестирование

- ◎ Задачи рабочего потока Тестирование:
 - поиск и описание дефектов системы (проявления недостатков ее качества),
 - оценка ее качества в целом,
 - оценка степени выполнения гипотез, лежащих в основе проектирования,
 - оценка степени соответствия системы требованиям

Развертывание (Deployment)

- ◎ Задачи рабочего потока Развертывание:
 - установка системы в ее рабочем окружении,
 - оценка ее работоспособность на том месте, где она должна будет работать

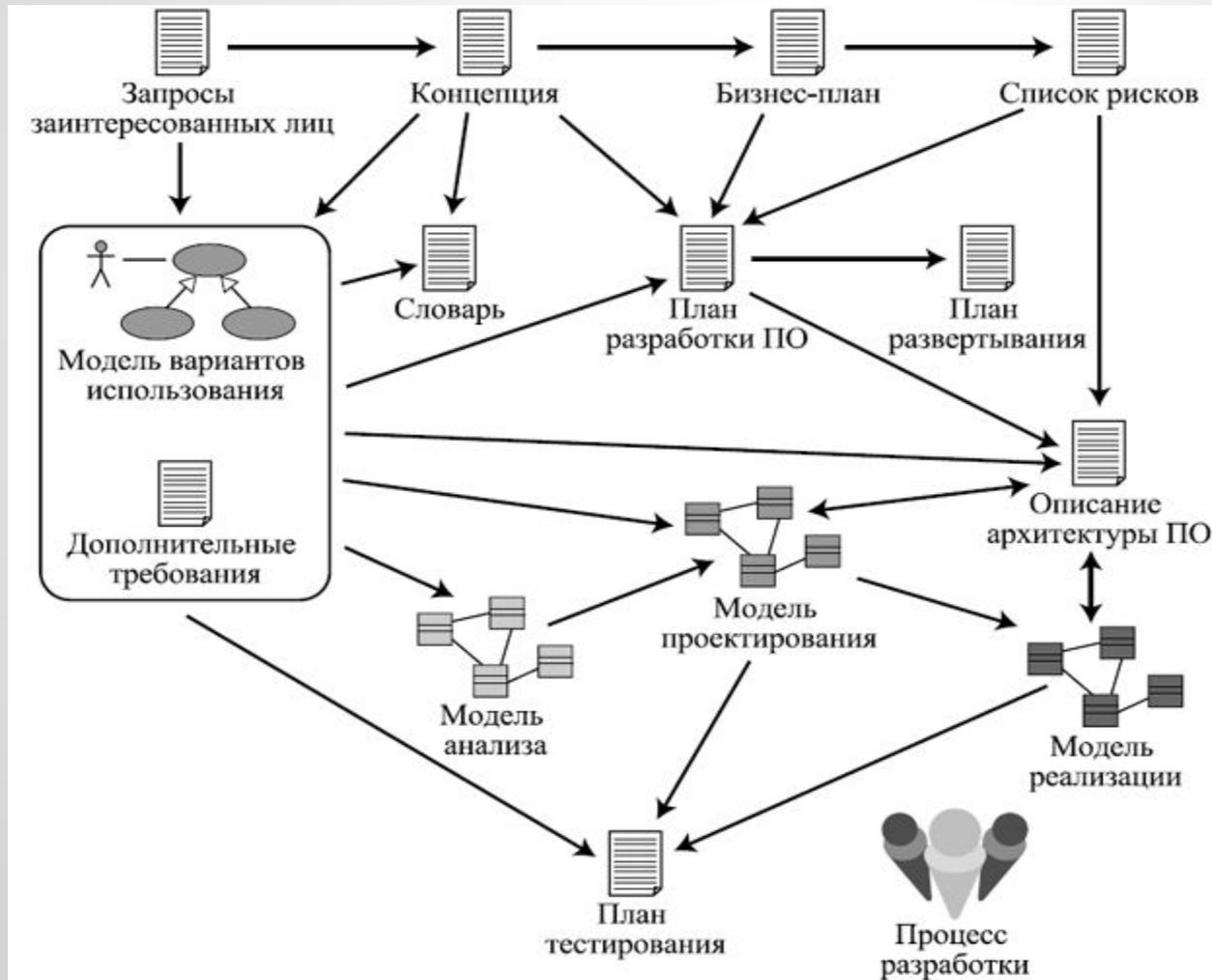
Структура типовой итерации



Артефакты

- ⦿ Каждый рабочий поток определяет набор связанных с ним артефактов
- ⦿ Артефакты, вырабатываемые в ходе проекта, могут быть представлены:
 - в виде баз данных и таблиц с информацией различного типа,
 - разных видов документов,
 - исходного кода и объектных модулей,
 - моделей, состоящих из отдельных элементов

Зависимости между артефактами



V-модель

- Концепция V-образной модели была разработана Германией и США в конце 1980-х годов независимо друг от друга
- Немецкая V-модель была разработана аэрокосмической компанией IAVG, американская – Национальным советом по системной инженерии и предназначалась для спутниковых систем

Схема V-модели



Особенности модели

- ◎ V-Model делает упор на тестирование как составную часть всех этапов разработки, а также на разработку прототипов конечного продукта
- ◎ Основной принцип V-модели заключается в том, что детализация проекта возрастает при движении слева направо, одновременно с течением времени

Достоинства

- ◎ Минимизация рисков
 - V-модель делает проект более прозрачным и повышает качество контроля проекта, что позволяет выявлять отклонения в проекте и риски на ранних стадиях
- ◎ Повышение качества
 - V-модель является стандартизированной моделью разработки, что позволяет добиться от проекта результатов желаемого качества

Достоинства

- ◎ Уменьшение стоимости проекта
 - Ресурсы на разработку, производство, управление и поддержку могут быть заранее просчитаны и проконтролированы.
- ◎ Повышение качества коммуникации между участниками проекта
 - Универсальное описание всех элементов и условий облегчает взаимопонимание всех участников проекта

Недостатки

- ⦿ Модель не предусматривает работу с параллельными событиями
- ⦿ В модель не входят действия, направленные на анализ рисков
- ⦿ Результат разработки становится понятным только при достижении низа буквы V

Конец лекции
