Лекция 7

Тема: «Проектирование программного обеспечения ИС»

Вопросы:

1. **Основы технологии разработки ПО ИС (СРС).**
2. **Технология Microsoft Solution Framework.**
3. **Технология RUP.**
4. **Технология Borland (СРС).**

Основные понятия технологии разработки ПО

Технология разработки ПО — упорядоченная совокупность взаимосвязанных технологических процессов в рамках ЖЦ ПО.

Технологический процесс — совокупность взаимосвязанных технологических операций.

Технологическая операция — основная единица работы, выполняемая определенной ролью, которая:

- подразумевает четко определенную ответственность роли;
- дает четко определенный результат (набор рабочих продуктов), базирующийся на определенных исходных данных (другом наборе рабочих продуктов);
- представляет собой единицу работы с жестко определенными границами, которые устанавливаются при планировании проекта.

Рабочий продукт — информационная или материальная сущность, которая создается, модифицируется или используется в некоторой технологической операции (модель, документ, код, тест и т.п.). Рабочий продукт определяет область ответственности роли и является объектом управления конфигурацией.

Роль — определение поведения и обязанностей отдельного лица или группы лиц в среде организации — разработчика ПО, осуществляющих деятельность в рамках некоторого технологического процесса и ответственных за определенные рабочие продукты.

Руководство — практическое руководство по выполнению одной или совокупности технологических операций. Руководства включают и себя методические материалы, инструкции, нормативы, стандарты и критерии оценки качества рабочих продуктов.

Инструментальное средство (CASE-средство) — программное средство, обеспечивающее автоматизированную поддержку деятельности, выполняемой в рамках технологических операций.

Стандарты технологии разработки ПО

ГОСТ 34.201—89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначения документов при создании автоматизированных систем»;

ГОСТ 34.320—96 «Информационные технологии. Система стандартов по базам данных. Концепции и терминология для концептуальной схемы и информационной базы»;

ГОСТ 34.321—96 «Информационные технологии. Система стандартов по базам данных. Эталонная модель управления данными»;

ГОСТ 34.601—90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»;

ГОСТ 34.602—89 «Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;

ГОСТ 34.603—92 «Информационная технология. Виды испытаний автоматизированных систем»;

РД 50-34.698-90 «Автоматизированные системы. Требования к содержанию документов»;

ГОСТ Р ИСО/МЭК 8824-3—2002 «Информационная технология. Абстрактная синтаксическая нотация версии один (АСН. 1). Ч. 3. Спецификация ограничения»;

ГОСТ Р ИСО/МЭК 10746-3—2001 «Информационная технология. Взаимосвязь открытых систем. Управление данными и открытая распределенная обработка. Ч. 3. Архитектура»;

ГОСТ Р ИСО/МЭК 15271-02 «Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207. Процессы жизненного цикла программных средств»;

ГОСТ Р ИСО/МЭК 15910—2002 «Информационная технология. Процесс создания документации пользователя программного средства»;

ГОСТ Р ИСО/МЭК 15408—2001 «Методы и средства обеспечения безопасности. Критерии оценки безопасности ИТ (ч. 1, 2, 3)»;

ГОСТ Р ИСО/МЭК 9594-8—98 «ИТ. ВОС. Справочник. Ч. 8. Основы аутентификации»;

ISO 12207, ISO 9000, CMM

Модель команды MSF

Модель команды отвечает на вопрос: как должна быть построена и структурирована проектная группа, для того чтобы можно было опти мизировать процесс проектирования по срокам, качеству и затратам. Структура команды должна позволять отслеживать постоянно изме няющиеся требования в проекте и создавать качественный продукт в условиях ограничений на время и ресурсы. Чтобы быть эффективной, группа, как правило, должна быть небольшой по численности.

Предлагаются следующие характеристики такой команды:

- каждый общается с каждым и каждый делает реальную работу;
- общие для всех членов группы цели и планы;
- каждый понимает как проблемы конечного пользователя, так и проблемы разработчика;
- каждый несет ответственность за свою работу, в том числе и перед группой.

Модель проектной группы MSF

Модель проектной группы создана на основе характеристик модели команды, рассмотренных выше. Она определяет основные роли, закрепленные за членами проектной группы. Для каждого члена группы определяется роль и зоны ответственности на весь период работы над проектом. То есть каждый сфокусирован на успехе проекта и настроен на работу в течение всего цикла проекта. Ключевым фактором успеха являются коммуникации между членами проектной группы. При этом реализуется параллельная работа всех участников группы над проектом, в состав которого в обязательном порядке включаются пользователи и обучающий персонал.

Модель проектной группы MSF практически не связана с организационной структурой, в одной проектной группе могут работать специалисты из различных подразделений. За каждым членом проектной группы закрепляется конкретная роль, для которой строится план работы, являющийся составной частью общего плана проекта. При этом каждая роль в модели проектной группы имеет свою особую компетенцию.

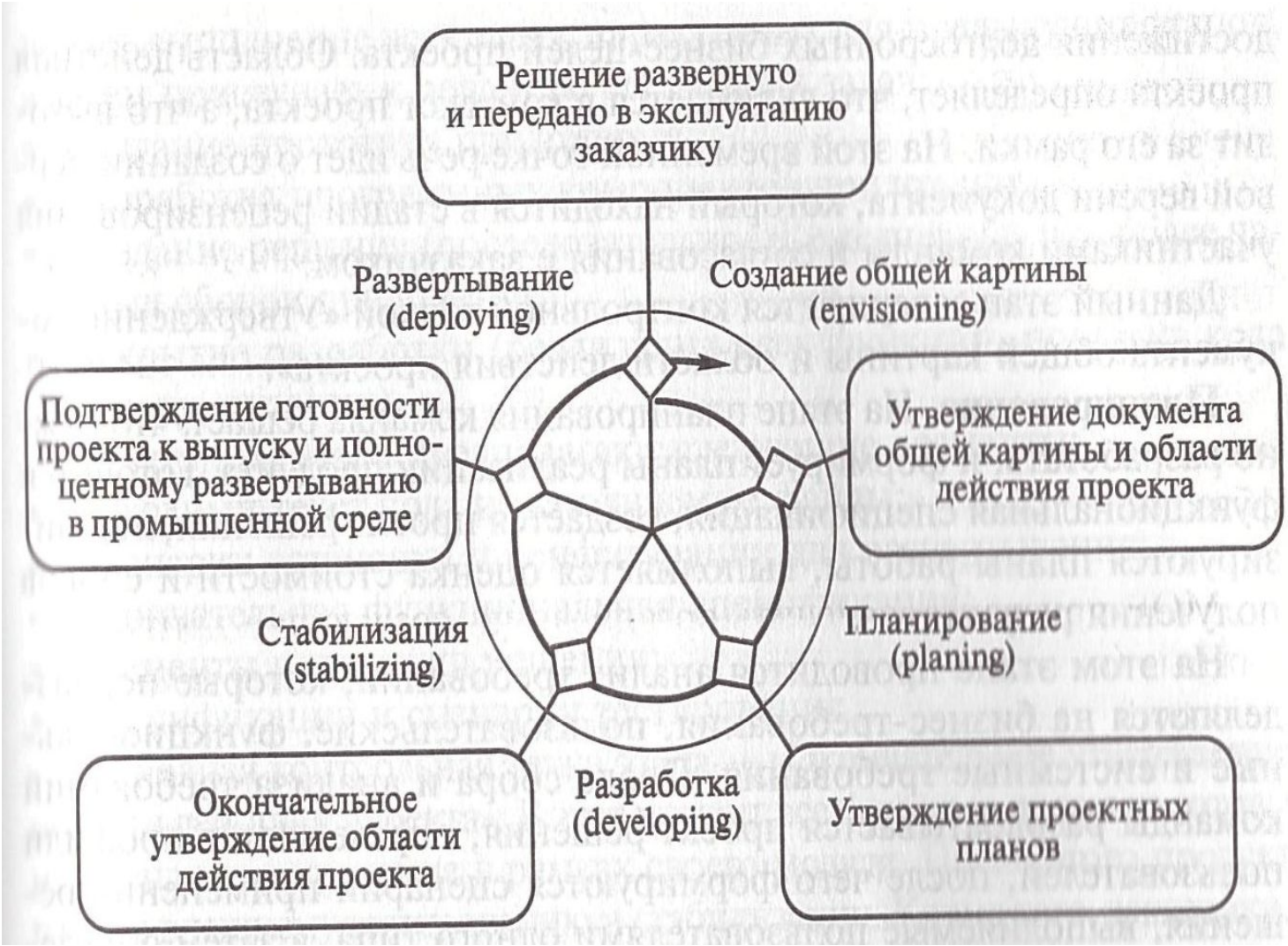
Если численность проектной группы меньше шести человек, то часть ролей может совмещать и выполнять один человек. При численности проектной группы больше шести человек одна роль может быть закреплена за несколькими специалистами, среди которых назначается лидер группы.

Модель проектной группы MSF

Параметры ролей проектной группы

Роль	Ответственность	Приоритеты
Менеджер проекта	Определение проблемы. Продвижение продукта	Удовлетворенность заказчика
Менеджер программы	Управление спецификациями. Координация работ. Отслеживание состояния проекта	Разработка качественного продукта в срок и в рамках выделенного бюджета. Поиск и решение проблем
Разработчик	Проектирование функциональности. Создание продукта. Тестирование продукта	Надежный и полный продукт
Тестер	Определение стратегии тестирования. Проведение тестирования. Отслеживание результатов тестирования	Согласованный и надежный проект
Инструктор	Разработка документации. Ведение глоссария. Тестирование. Обучение пользователей	Продукт, который можно использовать и сопровождать
Логистик	Прогнозирование ситуации. Подготовка внедрения. Сопровождение продукта на этапе внедрения. Обеспечение адекватной инфраструктуры	Обеспечение гладкого внедрения и развития продукта

Модель процесса проектирования MSF



Модель процесса проектирования MSF

Фазы модели процесса проектирования MSF

На фазе проекта **«Выработка концепции» («Анализ»)** проводятся работы до того, как сформированы требования. Все работы до заключения договора осуществляются обычно бесплатно для заказчика и длятся одну-две недели. Эта фаза необходима для того, чтобы команда разработчиков получила данные и оценила усилия, необходимые для создания функциональной спецификации, которая впоследствии будет использована при разработке.

Фаза **«Планирование»** включает в себя подготовку проектной группой функциональной спецификации, разработку дизайнов, подготовку рабочих планов, оценку проектных затрат и сроков разработки различных составляющих проекта.

Фаза **«Разработка»** предполагает активное и деятельное участие всех ролей в соответствии с их обязанностями в тестировании, выявлении дефектов, анализе удовлетворения нужд заказчика и других задачах. К моменту завершения этой фазы разработка всех компонент завершена и решение готово к комплексному тестированию.

На фазе **«Стабилизация»** фокус смещается в область тестирования и документирования решения, а также создания пилотного внедрения. Результатами этой фазы являются: окончательный продукт, документация выпуска, материалы поддержки решения, результаты тестирования, проектная документация и анализ пройденной фазы.

На фазе **«Внедрение»** выполняется установка и отладка системы в реальных условиях эксплуатации, передача системы персоналу поддержки и сопровождения, получение окончательного одобрения результатов проекта со стороны заказчика.

Инструменты разработки в MSF

- **Microsoft Repository –** механизм хранения и совместного использования объектов различными приложениями и средствами разработки программного обеспечения.
- **Информационная модель инструментария –** это описание систем или процессов, опубликованных в репозитарии. В нем может находиться любое число таких моделей, каждая с описанием своей системы или процесса. Модель определяет объекты, интерфейсы, отношения и связи, образующие систему. С другой стороны, данную модель можно рассматривать и как своего рода шаблон для систем конкретного типа. Стандартизация моделей позволяет разным средам и средствам разработки взаимодействовать с данными в хранилище.
- **Repository Browser –** позволяет просматривать содержимое репозитария.
- **Визуальный диспетчер компонентов -** обеспечивает разработчику возможность сохранять программные компоненты в репозитарии и тем самым упростить повторное использование кода. Таким образом, появляется возможность повысить производительность труда коллектива разработчиков за счет классификации и публикации компонентов и обеспечения доступа к ним. Работая с диспетчером, программист может найти необходимый фрагмент кода и воспользоваться им, загрузить шаблон проекта, запустить мастер или добавить в проект управляющий элемент ActiveX. Кроме того, репозитарии может играть роль центра хранения документации, включая принятые стандарты программирования, функциональные спецификации или архитектурные диаграммы.

Ключевые идеи RUP

- управляется прецедентами использования
- основан на архитектуре
- является итеративным и инкрементным

Прецеденты – это основной артефакт, на основании которого устанавливается желаемое поведение системы, проверяется и подтверждается правильность выбранной системной архитектуры, производится тестирование.

Ключевые идеи RUP

- **управляется прецедентами использования**
- **основан на архитектуре**
- **является итеративным и инкрементным**

Основным решением, принимаемым в ходе проекта, является **архитектура ПС.** Она устанавливает набор компонентов, из которых будет построено ПС, ответственность каждого из компонентов, четко определяет интерфейсы, через которые они могут взаимодействовать, а также способы взаимодействия компонентов друг с другом.

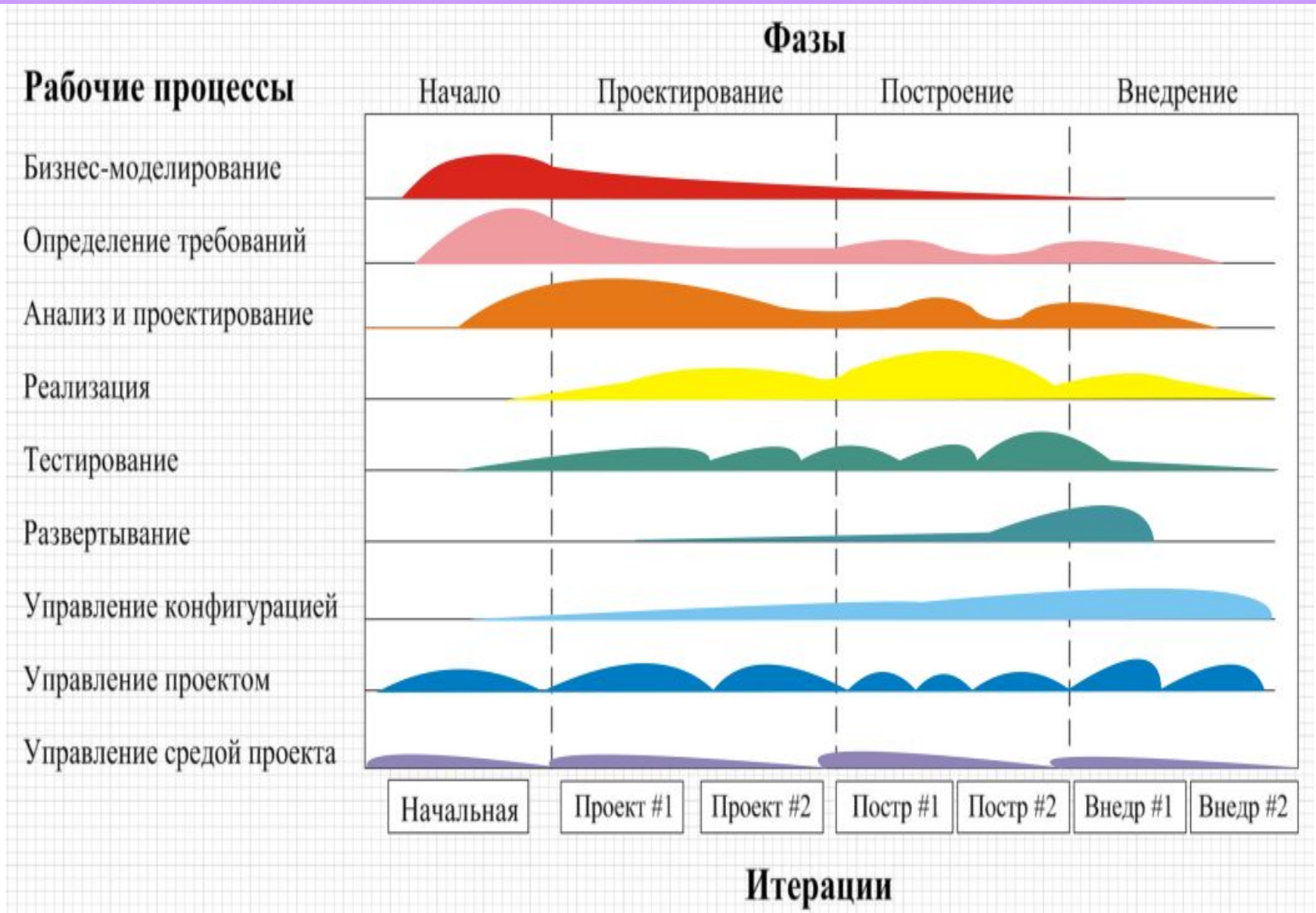
Ключевые идеи RUP

- **управляется прецедентами использования**
- **основан на архитектуре**
- **является итеративным и инкрементным**

Итеративным называется процесс, который предполагает управление потоком исполняемых версий системы.

Инкрементный процесс подразумевает постоянное развитие системной архитектуры при выпуске новых версий, причем каждая следующая версия усовершенствована в сравнении с предыдущей.

Жизненный цикл RUP разработки ПС

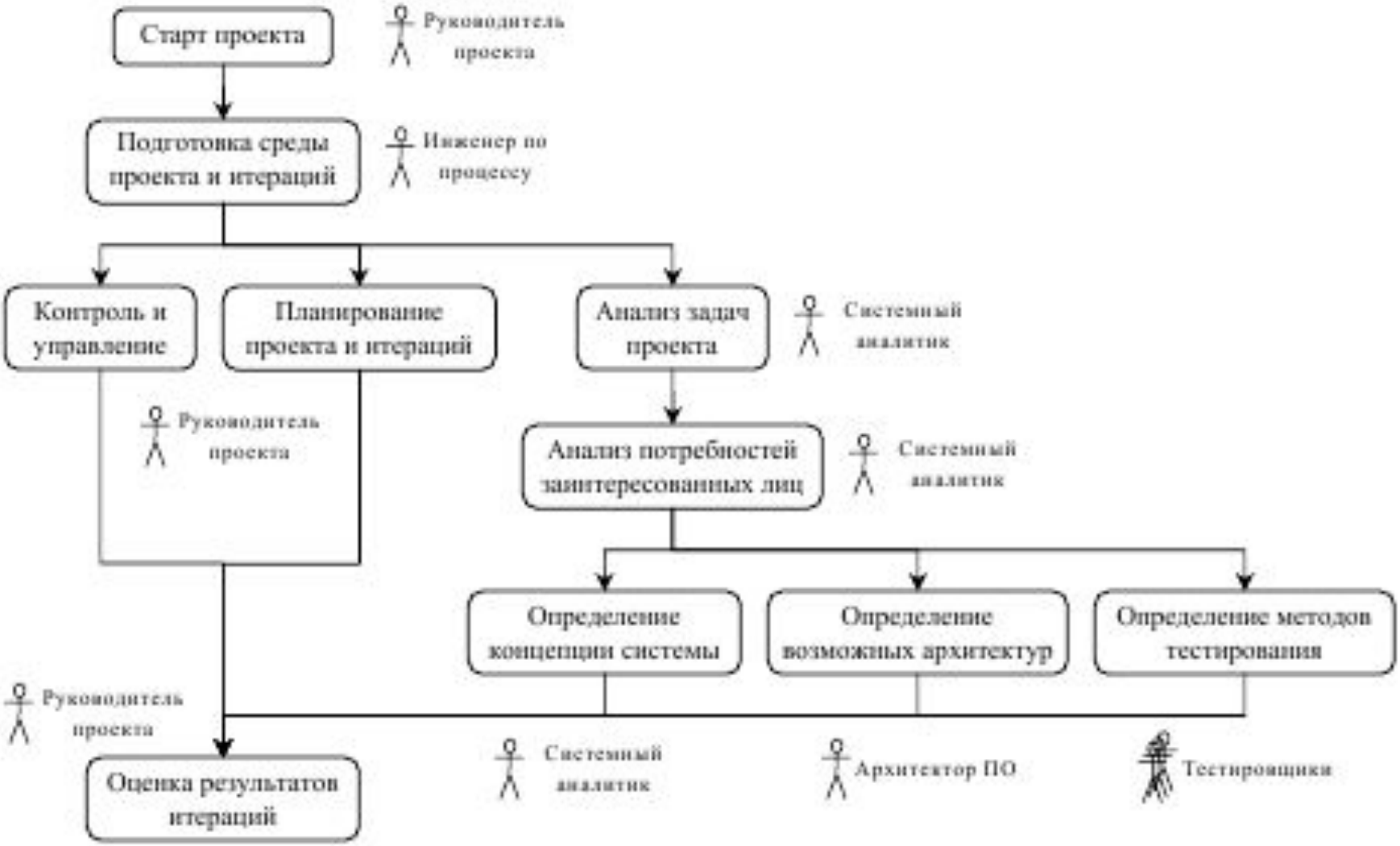


Фазы RUP

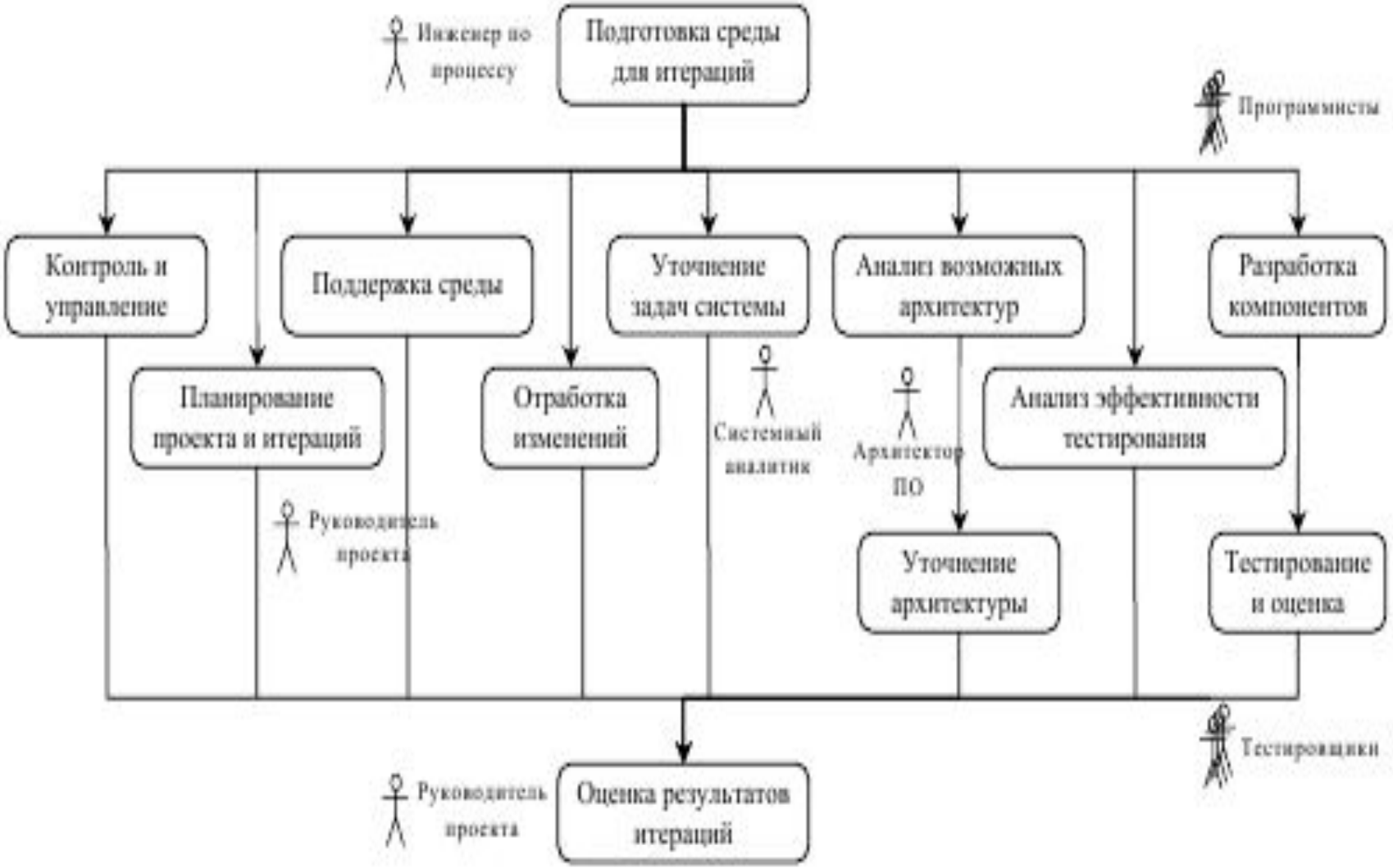
RUP выделяет в ЖЦ 4 основные фазы, в рамках каждой из которых возможно проведение нескольких итераций. Кроме того, разработка системы может пройти через несколько циклов, включающих все 4 фазы.

- **Фаза начала проекта.** Определяются основные цели проекта, руководитель и бюджет, технологии, инструменты, ключевые исполнители.
- **Фаза проектирования.** Разработка стабильной базовой архитектуры продукта, которая позволяет решать поставленные перед системой задачи и в дальнейшем используется как основа разработки системы.
- **Фаза построения.** Детальное прояснение требований и разработка системы, удовлетворяющей им, на основе спроектированной ранее архитектуры. В результате должна получиться система, реализующая все выделенные варианты использования.
- **Фаза внедрения.** Развертывание системы в ее рабочей среде, бета-тестирование, подгонка мелких деталей под нужды пользователей.

Пример хода работ на фазе начала проекта

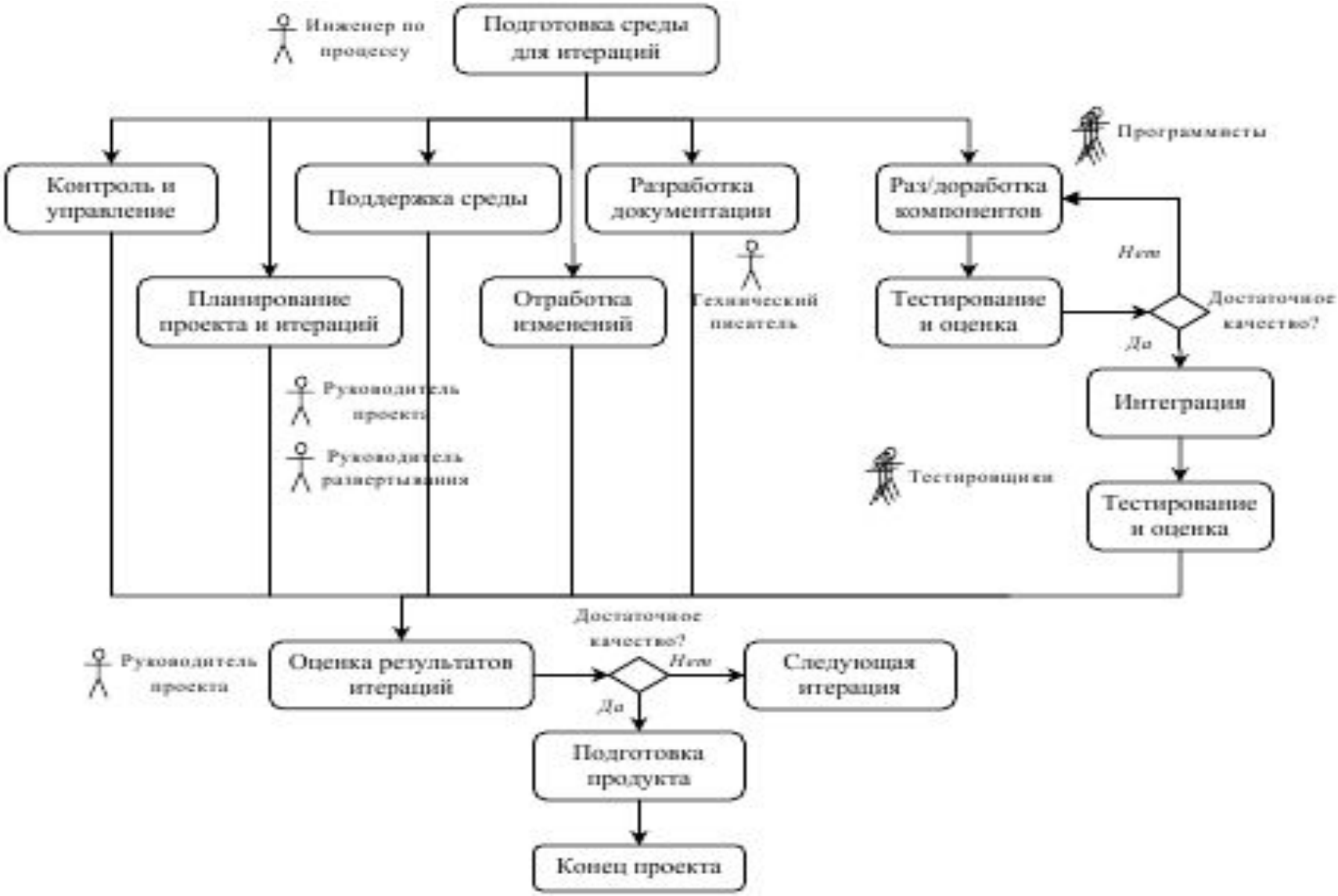


Пример хода работ на фазе проектирования



Пример хода работ на фазе построения

Пример хода работ на фазе внедрения



Рабочие процессы RUP

- **моделирование предметной области –** описывается структура и динамика организации;
- **определение требований –** описывается основанный на прецедентах метод постановки требований;
- **анализ и проектирование –** описываются различные виды архитектуры системы;
- **реализация –** собственно разработка программ, автономное тестирование и интеграция;
- **тестирование –** описываются тестовые сценарии, процедуры и метрики для измерения числа ошибок;
- **развертывание –** охватывает конфигурирование поставляемой системы;
- **управление конфигурациями и изменениями –** управление изменениями и поддержание целостности артефактов проекта;
- **управление проектом –** описывает разные стратегии работы с итеративным процессом;
- **управление средой разработки –** рассматриваются вопросы инфраструктуры, необходимой для разработки системы.

Рабочие процессы RUP

- **Моделирование предметной области (бизнес-моделирование).** Задачи этой деятельности – понять предметную область или бизнес-контекст, в которых должна будет работать система, и убедиться, что все заинтересованные лица понимают его одинаково, осознать имеющиеся проблемы, оценить их возможные решения и их последствия для бизнеса организации, в которой будет работать система. 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). Эта модель служит основой модели анализа.
- **Определение требований.** Задачи – понять, что должна делать система, и убедиться во взаимопонимании по этому поводу между заинтересованными лицами, определить границы системы и основу для планирования проекта и оценок затрат ресурсов в нем. Требования принято фиксировать в виде модели вариантов использования.
- **Анализ и проектирование.** Задачи – выработать архитектуру системы на основе требований, убедиться, что данная архитектура может быть основой работающей системы в контексте ее будущего использования. 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 и диаграммы развертывания.
- **Реализация.** Задачи – определить структуру исходного кода системы, разработать код ее компонентов и протестировать их, интегрировать систему в работающее целое.
- **Тестирование.** Задачи – найти и описать дефекты системы (проявления недостатков ее качества), оценить ее качество в целом, оценить выполнены или нет гипотезы, лежащие в основе проектирования, оценить степень соответствия системы требованиям.
- **Развертывание.** Задачи – установить систему в ее рабочем окружении и оценить ее работоспособность на том месте, где она должна будет работать.
- **Управление конфигурациями и изменениями.** Задачи – определение элементов, подлежащих хранению в репозитории проекта и правил построения из них согласованных конфигураций, поддержание целостности текущего состояния системы, проверка согласованности вносимых изменений.
- **Управление проектом.** Задачи – планирование, управление персоналом, обеспечение взаимодействия на благо проекта между всеми заинтересованными лицами, управление рисками, отслеживание текущего состояния проекта.
- **Управление средой проекта.** Задачи – подстройка процесса под конкретный проект, выбор и замена технологий и инструментов, используемых в проекте.

Модели RUP

- **модель бизнес-процессов** – формализует абстракцию организации;
- **модель предметной области** – формализует контекст системы;
- **модель вариантов использования** – формализует функциональные требования к системе;
- **модель анализа** – формализует идею проекта;
- **модель проектирования** – формализует словарь предметной области и области решения;
- **модель процессов** (необязательная) – формализует механизмы параллелизма и синхронизации в системе;
- **модель развертывания** – формализует топологию аппаратных средств, на которых выполняется система;
- **модель реализации** – описывает части, из которых собирается физическая система;
- **модель тестирования** – формализует способы проверки и приемки системы.

Взаимосвязь артефактов проекта по RUP