

# **Лекция 3. Рациональный унифицированный процесс разработки объектно-ориентированных программных систем**

## *Учебные вопросы:*

- 1. Архитектура программной системы (ПС).**
- 2. Унифицированный процесс разработки ПС.**
- 3. Управление риском.**

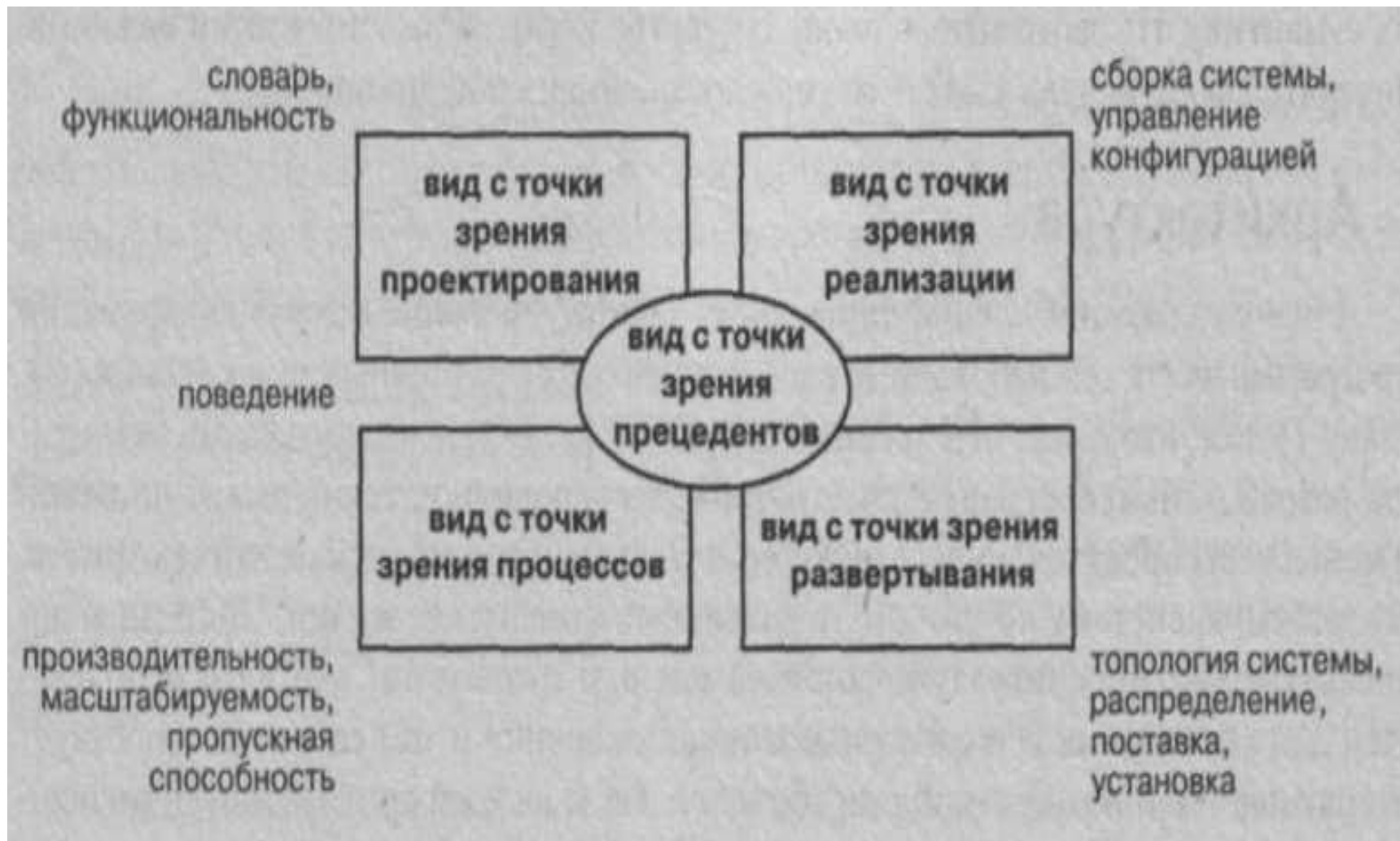
**Литература: [6], [10].**

# Архитектура программной системы (ПС)

**Архитектура программной системы (ПС)** – это набор внутренних структур ПС, которые видны с различных точек зрения и состоят из компонентов, их связей и возможных взаимодействий между компонентами, а также доступных извне свойств этих компонентов.

**Компонент** – это достаточно произвольный структурный элемент ПС, который можно выделить, определив интерфейс взаимодействия между этим компонентом и всем, что его окружает.

**Архитектура ПС** охватывает не только ее структурные и поведенческие аспекты, но и использование, функциональность, производительность, гибкость, возможности повторного применения, полноту, экономические и технологические ограничения и компромиссы, а также эстетические вопросы.



**Рисунок 1.1 – Моделирование системной архитектуры**

# Представления (виды) архитектуры ПС

*Вид с точки зрения прецедентов (Use case view)* охватывает прецеденты, которые описывают поведение системы, наблюдаемое конечными пользователями, аналитиками и тестировщиками.

*Вид с точки зрения проектирования (Design view)* охватывает классы, интерфейсы и кооперации, формирующие словарь задачи и ее решения.

*Вид с точки зрения процессов (Process view)* охватывает нити и процессы, формирующие механизмы параллелизма и синхронизации в системе.

*Вид с точки зрения реализации (Implementation view)* охватывает компоненты и файлы, используемые для сборки и выпуска конечного программного продукта.

*Вид с точки зрения развертывания (Deployment view)* охватывает узлы, формирующие топологию аппаратных средств системы, на которой она выполняется.

# Ключевые идеи RUP

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

Весь ход работ направляется итоговыми целями проекта, выраженными в виде прецедентов использования (use cases) – сценариев взаимодействия результирующей ПС с пользователями или другими системами, при выполнении которых пользователи получают значимые для них результаты и услуги.

Разработка начинается с выделения прецедентов использования и на каждом шаге контролируется степенью приближения к их реализации.

Прецеденты должны быть основным артефактом, 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.

Архитектура является одновременно основой для получения качественного ПС и базой для планирования работ и оценок проекта в терминах времени и ресурсов, необходимых для достижения определенных результатов. Она оформляется в виде набора графических моделей на языке UML.

# 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.

Процесс, являющийся одновременно итеративным и инкрементным, называется **управляемым рисками**.

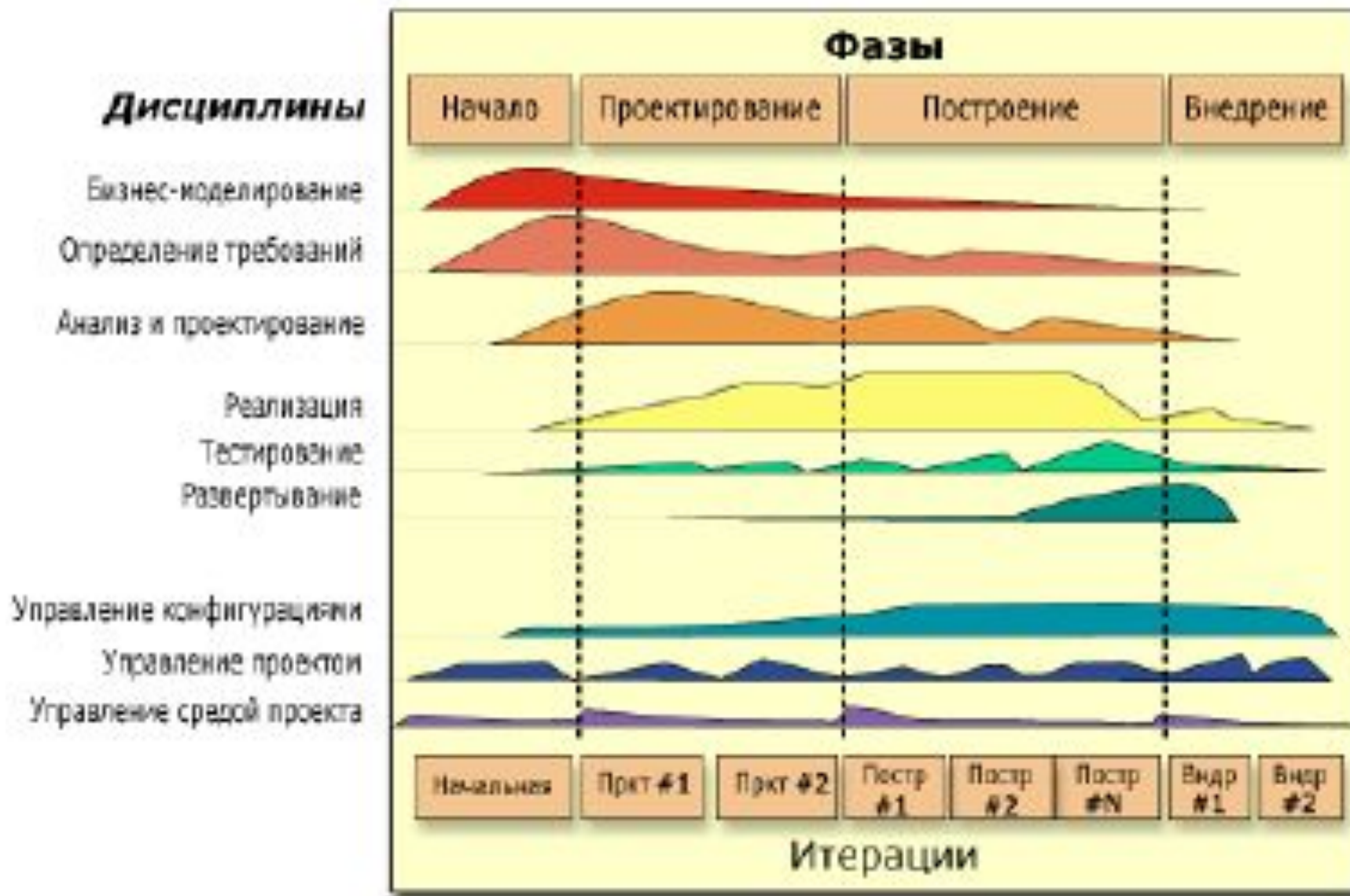


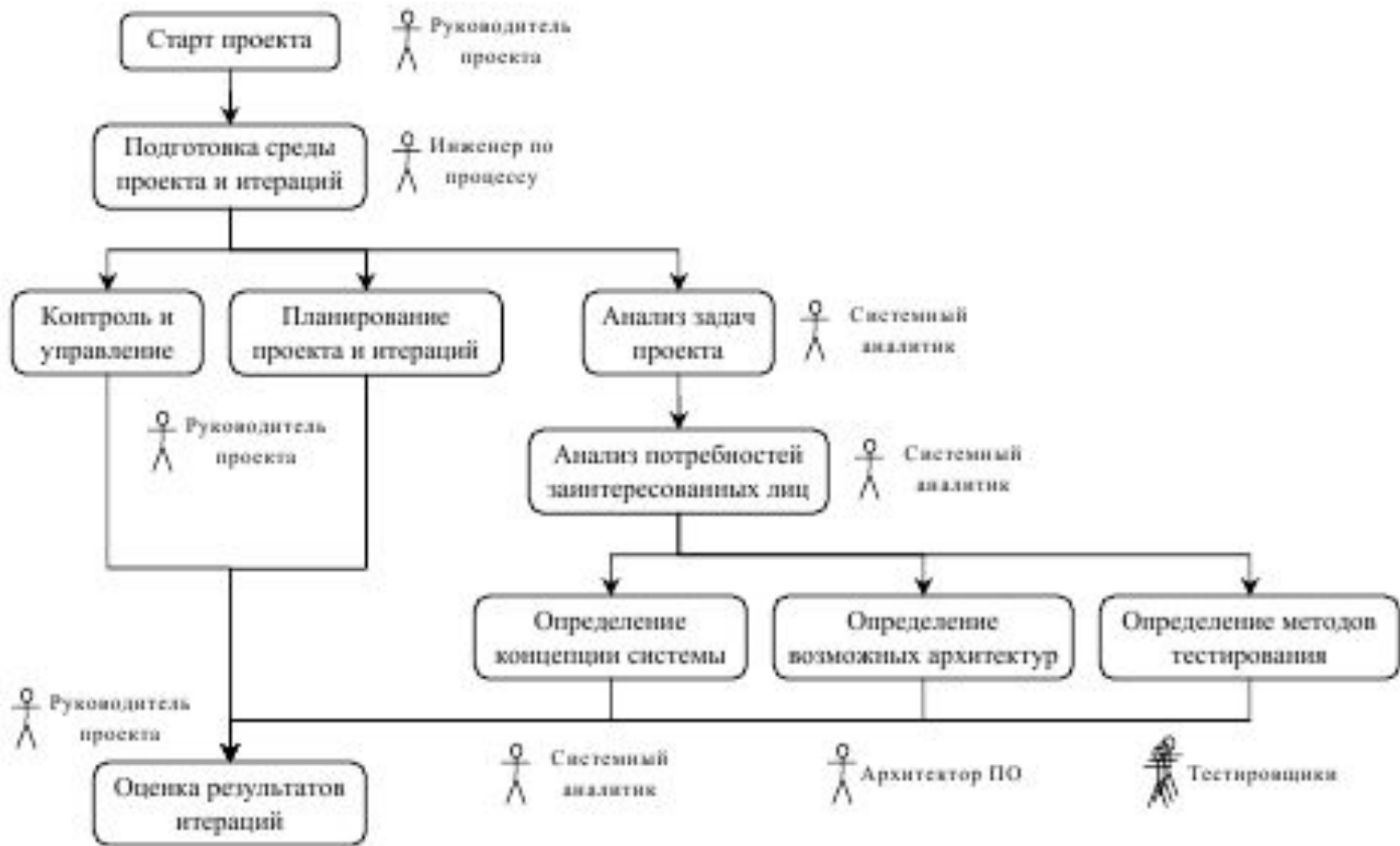
Рисунок 2.1 – Жизненный цикл процесса разработки ПС



# Фазы RUP

- Начало
- Проектирование
- Построение
- Внедрение

**Фаза начала проекта.** Основная цель этой фазы – достичь компромисса между всеми заинтересованными лицами относительно задач проекта и выделяемых на него ресурсов. На этой стадии определяются основные цели проекта, руководитель и бюджет, основные средства выполнения – технологии, инструменты, ключевые исполнители. Также, возможно, происходит апробация выбранных технологий, чтобы убедиться в возможности достичь целей с их помощью, и составляются предварительные планы проекта. На эту фазу может уходить около 10% времени и 5% трудоемкости одного цикла.



**Рисунок 2.2 – Пример хода работ на фазе начала проекта**

# Фазы RUP

- Начало
- Проектирование
- Построение
- Внедрение

## **Фаза проектирования.**

Основная цель этой фазы – на базе основных, наиболее существенных требований разработать стабильную базовую архитектуру продукта, которая позволяет решать поставленные перед системой задачи и в дальнейшем используется как основа разработки системы. На эту фазу может уходить около 30% времени и 20% трудоемкости одного цикла.



Рисунок 2.3 – Пример хода работ на фазе проектирования

# Фазы RUP

- Начало
- Проектирование
- Построение
- Внедрение

**Фаза построения.** Основная цель этой фазы – детальное прояснение требований и разработка системы, удовлетворяющей им, на основе спроектированной ранее архитектуры. В результате должна получиться система, реализующая все выделенные варианты использования. На эту фазу уходит около 50% времени и 65% трудоемкости одного цикла.



**Рисунок 2.4 – Пример хода работ на фазе построения**

# Фазы RUP

- Начало
- Проектирование
- Построение
- Внедрение

**Фаза внедрения.** Цель этой фазы – сделать систему полностью доступной конечным пользователям. На этой стадии происходит развертывание системы в ее рабочей среде, бета-тестирование, подгонка мелких деталей под нужды пользователей. На эту фазу может уходить около 10% времени и 10% трудоемкости одного цикла.

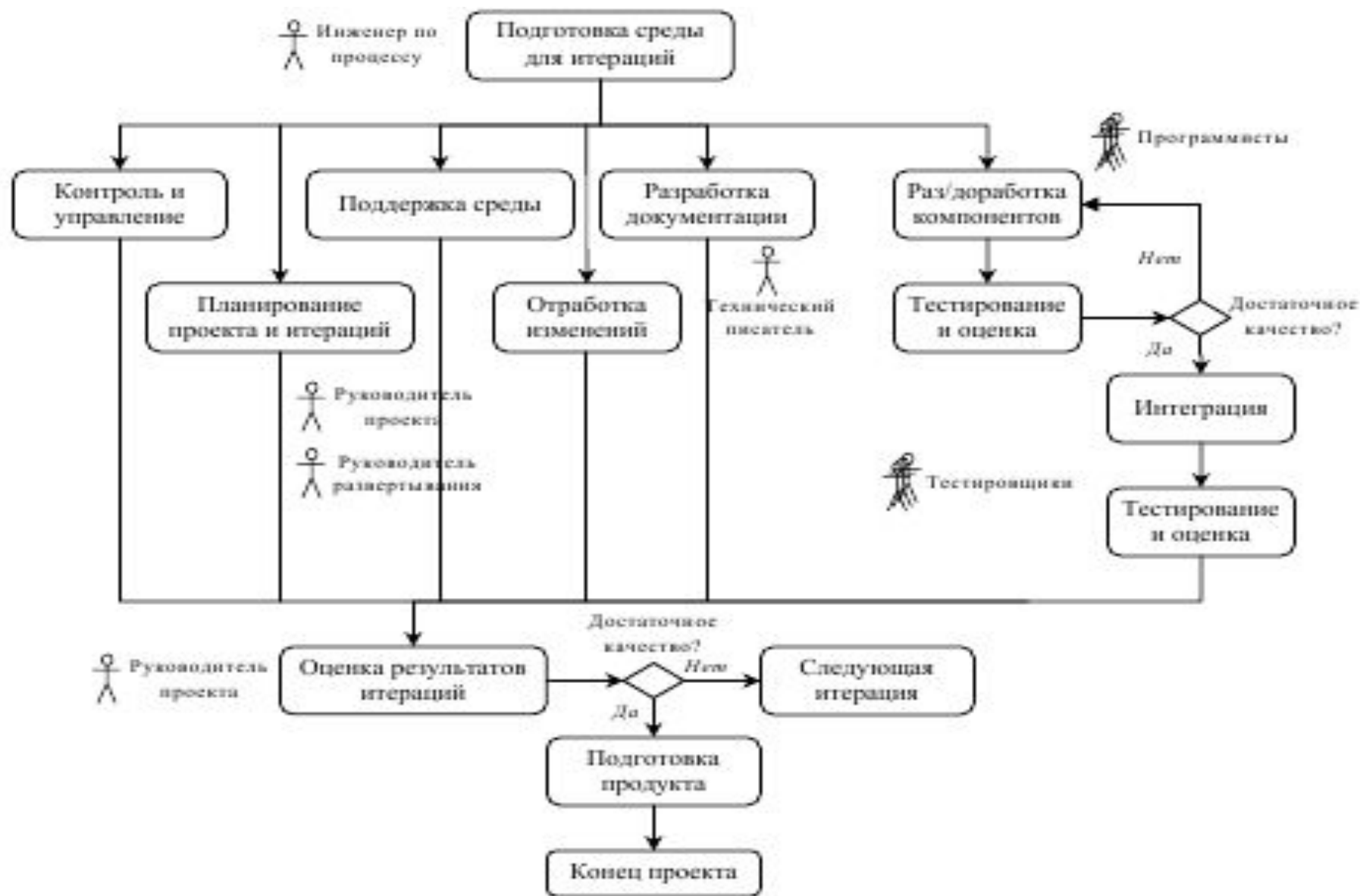


Рисунок 2.5 – Пример хода работ на фазе внедрения



# Дисциплины 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.

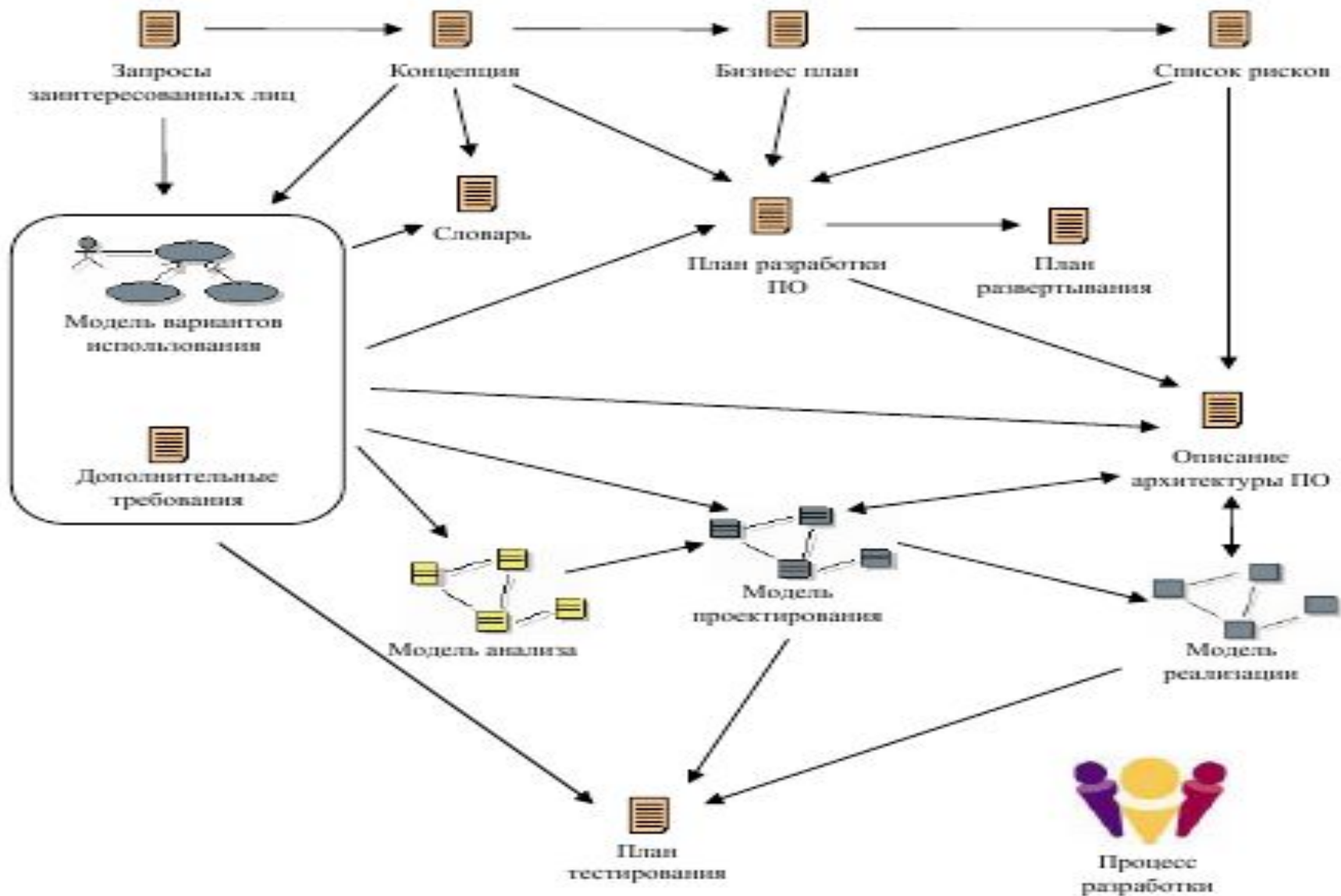
**Управление средой проекта.** Задачи – подстройка процесса под конкретный проект, выбор и замена технологий и инструментов, используемых в проекте.

# Дисциплины 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.

# 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.



**Рисунок 2.6 – Основные артефакты проекта по RUP и потоки данных между ними**

# Управление риском

Влияние риска вычисляют по выражению:

$$RE = P(UO) \times L(UO), \text{ где:}$$

**RE** – показатель риска (Risk Exposure – подверженность риску);

**P (UO)** – вероятность неудовлетворительного результата (Unsatisfactory Outcome);

**L (UO)** – потеря при неудовлетворительном результате.

Управление риском включает шесть действий:

1. **Идентификация риска** – выявление элементов риска в проекте.
2. **Анализ риска** – оценка вероятности и величины потери по каждому элементу риска.
3. **Ранжирование риска** – упорядочение элементов риска по степени их влияния.
4. **Планирование управления риском** – подготовка к работе с каждым элементом риска.
5. **Разрешение риска** – устранение или разрешение элементов риска.
6. **Наблюдение риска** – отслеживание динамики элементов риска, выполнение корректирующих действий.