

Структура RUP

Базовые элементы RUP (статический контекст, типы-классы составляющих элементов):

- Роли;
- Деятельности (activity) – элементарные работы;
- Артефакты – рабочие продукты;
- Технологические процессы (дисциплины).

Фазы ЖЦ и результаты:

- Исследование – **видение проекта и принятие решения**;
- Уточнение плана – **архитектурный прототип**;
- Построение – **бета-версия**;
- Развертывание – **продукт**.

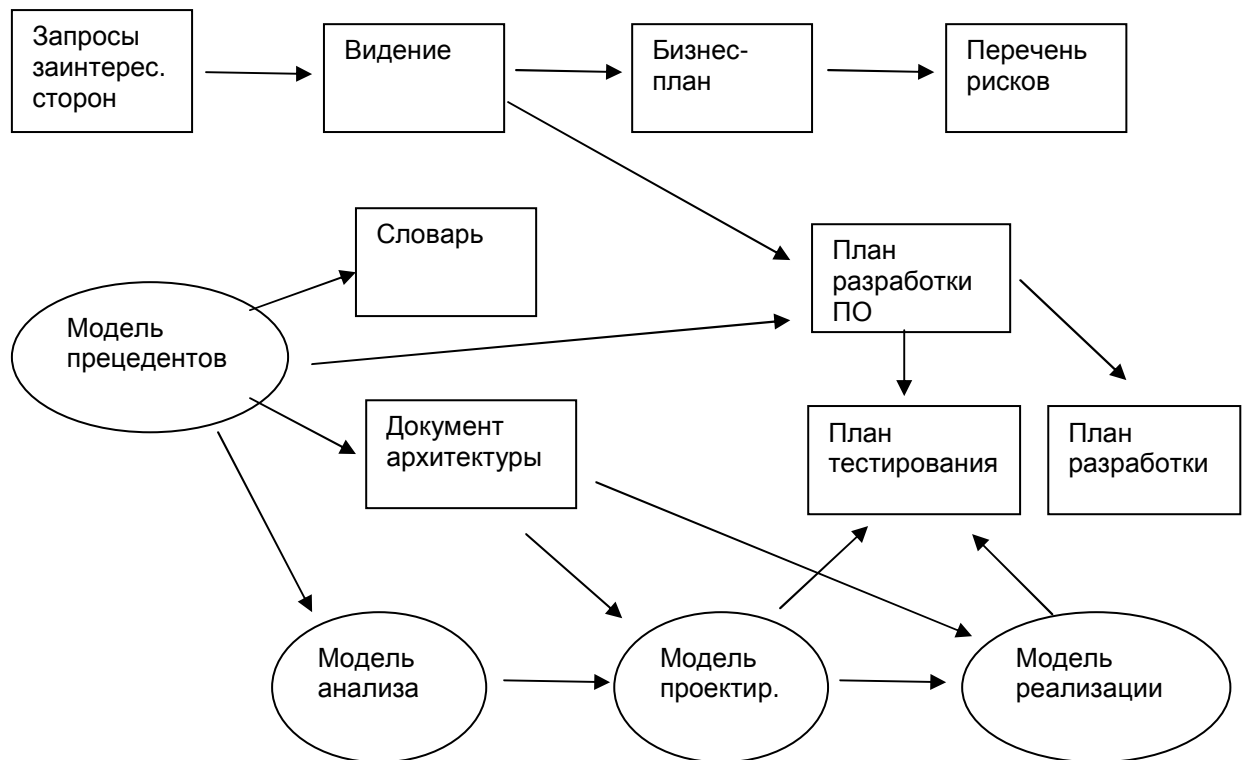
Технологические процессы ЖЦ:

- Моделирование производства (модель предметной области);
- Управление требованиями;
- Анализ и проектирование;
- Реализация (конструирование)
- Тестирование;
- Распространение ПО (установка, обучение);

Вспомогательные:

- Управление проектом (организация, менеджмент);
- Управление конфигурацией (сопровождение);
- Управление средой разработки;

Основные артефакты RUP



Фаза исследования проекта

Цели фазы исследования:

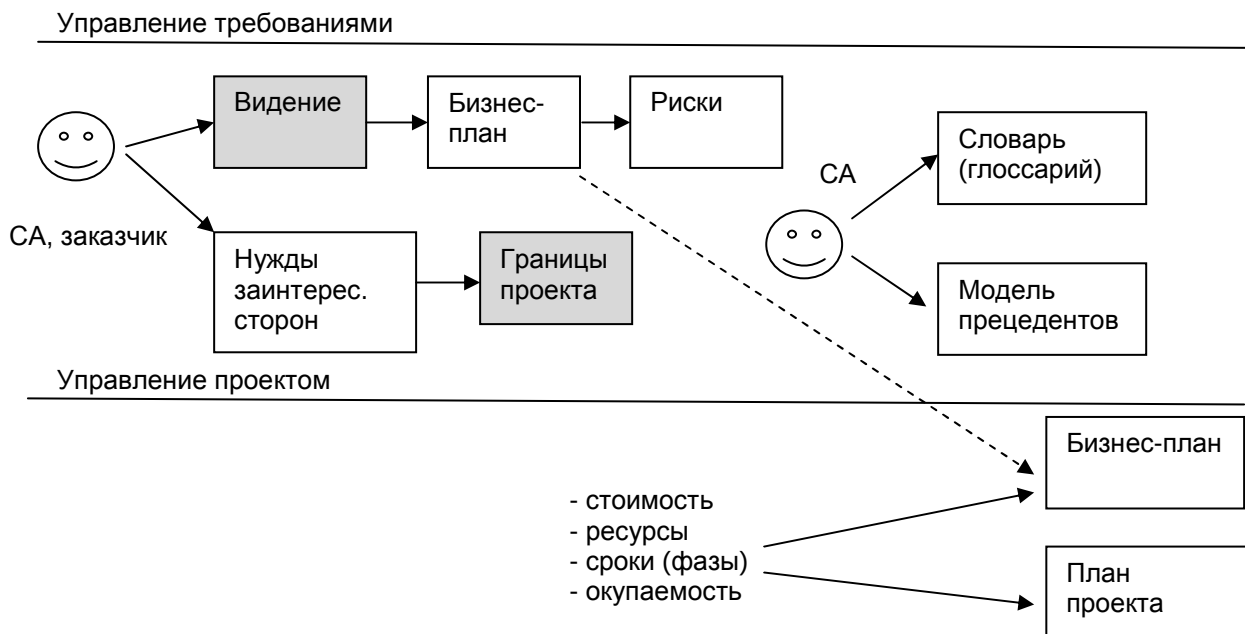
1. Область применимости, граничные условия – **видение проекта**;
2. Основные (критические) прецеденты системы;
3. План возможной архитектуры;
4. Предварительная оценка стоимости;
5. Предварительная оценка рисков.

Деятельности:

1. Формулировка **области действия проекта видения** – важнейшие требования, ограничения, среда исполнения – **рамки проекта**;
2. Разработка **бизнес-плана**, оценка рисков, кадровое обеспечение, цена, график работ, рентабельность;
3. Разработка возможной **архитектуры**.

Артефакты:

1. Документ видения
2. Обзор модели прецедентов
3. Начальный глоссарий проекта
4. Начальный бизнес-план
5. Начальная оценка рисков
6. План проекта – фазы и итерации



Фаза уточнения плана

Цели фазы уточнения – принятие окончательного решения относительно построения и развертывания:

1. Базовая линия архитектуры;
2. Базовая линия видения;
3. Базовая линия точного плана построения;
4. Обоснование, что базовая архитектура позволяет получить требуемое видение за разумную цену, сроки и с разумными рисками.

Деятельности:

1. Детальная проработка видения;
2. Критические прецеденты;
3. Проработка ТП среды разработки;
4. Подробная разработка архитектуры и выбор компонентов. Создание/покупка/повторное использование компонентов, график фазы построения.

Артефакты:

1. Модели прецедентов (80%);
2. Доп. спецификации на нефункциональные требования;
3. Описание архитектуры;
4. Выполнимый архитектурный прототип;
5. Перечень рисков и бизнес-план (скорректированный);
6. План проекта, итерации и критерии оценки итераций;

Итерация 1. Создание архитектурного прототипа:

- План итерации, архитектурные риски.
- Управление требованиями. Выбор прецедентов и сценариев, направляющих разработку архитектуры. На основании этого – обновление плана итерации и перечня рисков.
- Управление требованиями. Создание прототипа пользовательского интерфейса.
- Анализ и проектирование. Выделение очевидных классов, разбиение системы на подсистемы и подробное рассмотрение прецедентов. Артефакт: документ архитектуры.
- Анализ и проектирование. Уточнение и гомогенизация классов, определение архитектурно значимых классов.
- Анализ и проектирование. Рассмотрение низкоуровневого разбиения на пакеты.
- Анализ и проектирование. Согласование среды реализации, принятие решений относительно ключевых сценариев, определение формальных интерфейсов классов.
- Анализ и проектирование. Рассмотрение параллелизма и распределения архитектуры.
- Реализация. Модель реализации – физическое представление.

- Реализация. Планирование интеграции.
- Реализация. Планирование интегральных и системных испытаний.
- Реализация. Реализация классов и интеграция.
- Реализация. Интеграция реализованных подсистем.
- Тестирование. Оценка выполнимой архитектуры.

Фаза построения

Завершающая итерация фазы построения:

- Планирование итерации
- Реализация. Планирование интеграции на системном уровне.
- Тестирование. Планирование и проектирование тестирования на системном уровне.
- Анализ и проектирование. Уточнение реализации прецедентов.
- Планирование и проектирование интегральных тестов на подсистемном и системном уровнях.
- Реализация. Разработка кодов и тестирование блоков.
- Реализация. Планирование и реализация блочных испытаний.
- Реализация. Тестирование блоков в подсистеме.
- Реализация и тестирование. Интеграция подсистемы.
- Реализация. Тестирование подсистемы.
- Реализация. Выпуск подсистемы.
- Реализация. Интеграция подсистемы.
- Тестирование. Тестирование интеграции.
- Тестирование. Тестирование системы.

Фаза развертывания

Деятельности:

- Бета-тестирование
- Параллельное функционирование с существующей системой и ее замещение
- Конвертирование данных (в т.ч. БД)
- Подготовка пользователей и персонала
- Маркетинг, распространение и продажа.

Процесс разработки, управляемый архитектурой

Архитектура - это:

- Общая организация программной системы;
- Структура – структурные элементы, интерфейсы и связи, интерфейсы как элементы поведения (функционала);
- Иерархия и модульность;
- Архитектурный стиль.

Архитектура – минимально возможное описание программной системы, достаточное для понимания ее сущности и функционирования (**удаление с потерей целостности, добавление избыточно**).

Архитектура имеет **множественное представление** (срезы).

Представление	Роль	Содержание
Прецедентное	Системный архитектор	Ключевые сценарии (прецеденты), описывающие реализацию функционала системы
Логическое	Пользователь	Представление системы с точки зрения конечного пользователя (внешний вид, логическая структура).
Реализационное	Программист	Структура программного кода: пакеты, классы, библиотеки, интерфейсы, абстракции.
Процедурное	Системные интеграторы	Процессы, потоки, параллелизм, взаимодействие. Производительность, масштабируемость.
Представление распространения (развертывания)		Топология развертывания системы, распределение компонент по узлам сети, процедуры развертывания и администрирования

Прецедентное представление является **центральной**.

Артефакты, связанные с архитектурой:

1. **Описание архитектуры, SAD – software architecture description (основной).**
2. **Архитектурный прототип – базовая линия разработки (основной).**
3. Директивы проекта
4. Структура ПО в среде разработки.
5. Структура команды разработчиков.

Модульная (компонентная) разработка. Компонента – независимая, замещаемая часть системы, выполняющая очевидную функцию в архитектуре.

Архитектурные концепции – стандарты проектирования.

1. **Архитектурный стиль.** Примеры: клиент-сервер, система, управляемая событиями.
2. **Архитектурный механизм.** Примеры: СУБД, сервер транзакций.
3. **Архитектурный шаблон.** Примеры: Model-View-Controller.

Процесс разработки, управляемый прецедентами

Прецедент – последовательность исполняемых системой действий, приводящая к значимому результату.

Актор – участник взаимодействия с системой.

Прецедент – срез поведения системы при исполнении некоторой единицы функционала.

Поток событий – отдельная ветка последовательности событий. Описывается текстуально. Основной поток, альтернативные потоки.

Класс прецедента (сценарий) – объединение основного и альтернативного потоков.

На начальном этапе взаимодействием и синхронизацией прецедентов пренебрегают.

Для описания различных аспектов прецедентов в UML используются:

- Диаграммы прецедентов - основные;
- Диаграммы классов – структуры данных, связанные с прецедентами;
- Диаграммы взаимодействий (последовательностей, коммуникационные) – реализация потока событий в различных объектах системы;
- Диаграммы состояния – изменение состояния системы при исполнении прецедентов.

Технологический процесс моделирования производства

Синоним: анализ предметной области. Модель технологического процесса, подвергаемого автоматизации. Используются те же средства, что и при управлении требованиями, только модель составляется для действующего процесса.

Артефакты:

1. **Документ видения производства (основной);**
2. **Модель производственных прецедентов (основной);**
3. **Модель объектов производства (основной);**
4. Глоссарий (словарь) производства;
5. Оценка целевой организации;
6. Правила производства;
7. Дополнительная производственная спецификация.

Сценарии, по которым осуществляется моделирование, в зависимости от того, как будет меняться схема технологического процесса в результате автоматизации:

1. **Организационная схема.** Если схема технологический процесс производства не меняется, то создается описание текущего производства: процессы (прецеденты) и требования, которая переносится 1-1 в процесс управления требованиями.
2. **Моделирование предметной области.** Статическая модель данных предметной области без описания процессов (прецедентов) – входит в модель объектов производства.
3. **Новое производство.**
4. **Реорганизация.**
5. **Общая модель производства** – усреднение моделей производства нескольких организаций при разработке общего программного проекта.

6. **Одно производство для нескольких систем** – части одного производства рассматриваются как основа для нескольких программных проектов.

Переход от модели производства к моделям системы:

- Модель производственных прецедентов – в модель прецедентов;
 - Модель объектов производства – в модель проектирования;
- Преобразование при введении автоматизации:
- Категории производства – в классы анализа;
 - Производственные исполнители дублируются (переносятся) в программные объекты производства (клерк – автоматизированный клерк).

Технологический процесс управления требованиями

Цели:

- Единое представление функционала системы;
- Определение границ системы - **масштаб**;
- Основа для планирования содержания итераций, оценки стоимости и сроков исполнения;
- Определение **пользовательского интерфейса** системы;

Требования классифицируются как **функциональные** и прочие (**нефункциональные**) и свойства. **Свойства** – обобщенные требования к поведению системы.

Нефункциональные

1. Практичность (Usability)
2. Надежность (Reliability)
3. Производительность (Performance)
4. Возможность поддержки (Supportability) – требования к поддержке системы после ее выпуска, а также требования к среде проектирования.
5. Безопасность

Параметры (характеристики) требований:

Приоритеты

1. Обязательные (О)
2. Желательные (Ж)
3. Возможные (В)
4. Перспективные (П)

Полезность

1. Критическое (К)
2. Важное (В)
3. Полезное (П)

Решаемость

1. Проблематичное (П)
2. Выполнимое (В)
3. Тривиальное (Т)

Трудоемкость

1. Высокая (В)
2. Средняя (С)
3. Малая (М)

Типы требований (по уровням детализации):

- Запросы заинтересованных сторон;
- Свойства – требования абстрактного вида;
- Программные требования – требования к поведению отдельных компонент.
- Прецеденты.

Артефакты:

- Глоссарий (словарь);
- Архив прецедентов;
- Прототип пользовательского интерфейса.

Технологический процесс. Анализ проблемы – понимание нужд заинтересованных сторон – определение системы на основе набора свойств – управление масштабом – уточнение определения системы – прецеденты и пользовательский интерфейс.

Технологический процесс анализа и проектирования ?????

Модель анализа – классы анализа и подсистемы, построенные на основе модели прецедентов и функциональных требований к системе. Единственная, т.к. описывает основные сущности системы на уровне функциональных требований.

Модель проектирования – согласование результатов анализа с нефункциональными требованиями, средой реализации, архитектурой

Артефакты систем реального времени – оболочка (шаблон проектирования, протокол, сигнал, событие).

Элементы технологического процесса

• Определение потенциальной архитектуры –

- эскиз архитектуры

- модель классов анализа из архитектурно-значимых прецедентов

- взаимодействие прецедентов и классов анализа (диаграммы устойчивости)

Уточнение архитектуры.

Анализ поведения.

Проектирование компонентов.

Проектирование компонентов реального времени.

Проектирование баз данных.

Технологический процесс реализации (конструирования)

Состоит из множества итераций, на каждой из которых интегрируются компоненты, реализованные и прошедшие блочное тестирование.

Технологический процесс сборки: Компоненты – интеграция – подсистема реализации – интеграция – система.

Цели:

- Структура кода – на основе подсистем реализации и уровней;
- Реализация (конструирование) кода классов;
- Блочное тестирование;
- Интегрирование в исполняемую систему.

Конструкция – функциональная версия системы или ее части, демонстрирующая сокращенный вариант возможностей системы. Синоним: **сборка** – вариант интеграции компонентов в подсистему реализации (подсистем – в систему), «сырой» вариант действующей системы на текущий момент.

Интеграция – поэлементная интеграция – на каждом этапе одной изменой компоненты, **позтапная** – одновременно несколько новых или измененных компонент.

Прототип – работоспособный, но функционально неполный вариант системы или ее части, предназначенный для оценки, анализа и экспериментирования:

- **Поведенческий** – моделирование поведения системы (например, образец интерфейса пользователя, эмулятор для анализа API сервера при отсутствии полной документации);
- **Структурный** – «заготовки» компонент проектируемой системы, которые эволюционируют в итоговый проект;
- **Пробные** – временные, используемые для текущих нужд;
- **Эволюционные** – в дальнейшем развиваются как части проекта.

Артефакты:

- Компонент;
- Подсистема реализации;
- План проведения интеграции.

Технологический процесс тестирования

Цели:

- Проверка взаимодействия компонентов;
- Проверка правильности интеграции;
- Проверка точности реализации всех требований;
- Выявление дефектов и их устранение до фазы развертывания.

Взаимосвязь тестирования и качества. Качество – отсутствие дефектов и соответствие **реальным** требованиям пользователей (потребительские качества).

Классификация тестов:

- Тестируемый параметр качества:
 - надежность (отсутствие багов);
 - функциональные возможности – соответствие прецедентам и поведение;
 - производительность.
- Этап тестирования:
 - блочное тестирование (модульное тестирование при конструировании компонента);
 - интегральное тестирование (составных компонент или подсистем);
 - системное тестирование;
 - приемочное тестирование (перед развертыванием).
- Тип теста:
 - аттестационный – оценка производительности в сравнении с прототипом или в виде некоторой меры;
 - конфигурационный – работоспособность системы в различных конфигурациях;
 - функциональные испытания – тестирование прецедентов;
 - установочные испытания – проверка установки в различных конфигурациях с различными ресурсными ограничениями;
 - тестирование целостности – устойчивость к run-time ошибкам;
 - испытание под нагрузкой – функционирование при изменении числа пользователей, транзакций, соединений (рабочая нагрузка);
 - испытание в жестком режиме – ограниченные ресурсы или высокая рабочая нагрузка;
 - эксплуатационные испытания – производительность в различных конфигурациях при постоянной рабочей нагрузке.

Регрессионное тестирование – выполнение всех ранее проведенных тестов на очередной версии (после внесенных изменений).

Модель тестирования - компоненты:

- **Контрольная задача** – набор тестовых данных, условий выполнения и ожидаемый результат.
- **Методика испытаний** – описание набора контрольных задач, их настройки, выполнения и оценки результатов.
- **Сценарий испытаний** – компонента автоматизации методики испытаний.
- **Классы и компоненты испытаний** – программные компоненты системы тестирования.
- **Описание программных средств тестирования** – в виде диаграмм последовательностей или диаграмм взаимодействия.

Артефакты:

- Модель тестирования;
- План тестирования;
- Результаты тестирования и данные, собранные в процессе тестирования;

- **Модель рабочей нагрузки** для эксплуатационных испытаний.
- **Дефекты** – запросы на внесение изменений (управление конфигурацией).
- **Классы и пакеты тестов, подсистемы и компоненты тестирования.**

Технологический процесс распространения

Деятельности:

- Бета-тестирование в целевой операционной среде;
- Оформление ПО;
- Распространение;
- Установка;
- Обучение конечных пользователей, продавцов;
- Замещение существующего ПО, конвертирование БД.

Типы развертывания:

- Развертывание заказных систем;
- Развертывание из архива (программа установки);
- Загрузка из Интернет;

Артефакты:

- Исполняемое ПО;
- Артефакты установки – сценарии, инструментальные средства, файлы, руководства.
- Материалы поддержки – руководство пользователя и по эксплуатации.
- Обучающие материалы.

Технологический процесс:

- Планирование распространения;
- Разработка материалов поддержки;
- Тестирование продукта в среде разработчика;
- Создание версии;
- Бета-тестирование версии;
- Тестирование продукта в целевой системе;
- Оформление продукта;
- Обеспечение доступа к узлу загрузки (Интернет).

Технологический процесс управление конфигурациями

Куб CCM:

- **Управление конфигурацией** – артефакты конфигурации (3 уровня) – компоненты, подсистемы, система. Каждый артефакт существует в виде дерева версии, зависимости между артефактами, обеспечение целостности системы артефактов и правильности сборки конфигурации из различных версии.

- **Управление запросами на внесение изменений** – исправление дефекта (тестирование), улучшение качества, внесение изменений в требования, документирование отличий между версиями. Состояния запроса: новый, зафиксированный, утвержденный, принятый, завершённый. Накапливаемые данные: затрагиваемые артефакты, влияние на архитектуру, стоимость, график. В зависимости «глубины» вносимых изменений влияет на технологические процессы следующей итерации (управление требованиями, анализ – изменение классов, структуры БД, конструирование).
- **Управление состоянием и изменениями** (управление ПП) – метрические характеристики разработки (прогресс проекта относительно изменений, количество изменений в различных состояниях, «возраст» изменений), распределение изменений по серьезности, приоритету, изменениям архитектуры, причинам появления.

Артефакты:

- План управления конфигурацией;
- Запросы на внесение изменений;
- Модель реализации;
- Метрики и отчеты.

Технологический процесс управления средой разработки (-)

Технологический процесс управления проектом

Деятельности:

- Планирование ЖЦ и планирование отдельных итераций;
- Управление рисками
- Наблюдение за прогрессом проекта и метрики

Не входят:

- Управление персоналом, наем, обучение тренировка;
- Управление контрактами;
- Управление ресурсами.

Крупнозернистый план фаз:

- Продолжительность основных фаз и итераций и их результаты;
- Схема кадрового обеспечения (подбор команды)

План итерации – текущей и следующей (исполнение и планирование), сетевой график, диаграммы Ганта.

Управление рисками

Риск – случайный фактор, могущий повлиять на успех проекта, стоимость, сроки реализации. **Прямой** – контролируемый риск, **косвенный** – неконтролируемый. **Вероятность появления, степень влияния на проект (серьезность).**

Стратегии управления рисками – уклонение от риска (реорганизация проекта), перевод риска (на других участников проекта), Принятие риска – сосуществование как с непредвиденным обстоятельством (снизить вероятность, применить запасной план).