

Процесс и инструментарий RUP

Игорь Бочкарев
НТЦ ОАО «Радиозавод»

Основные принципы RUP

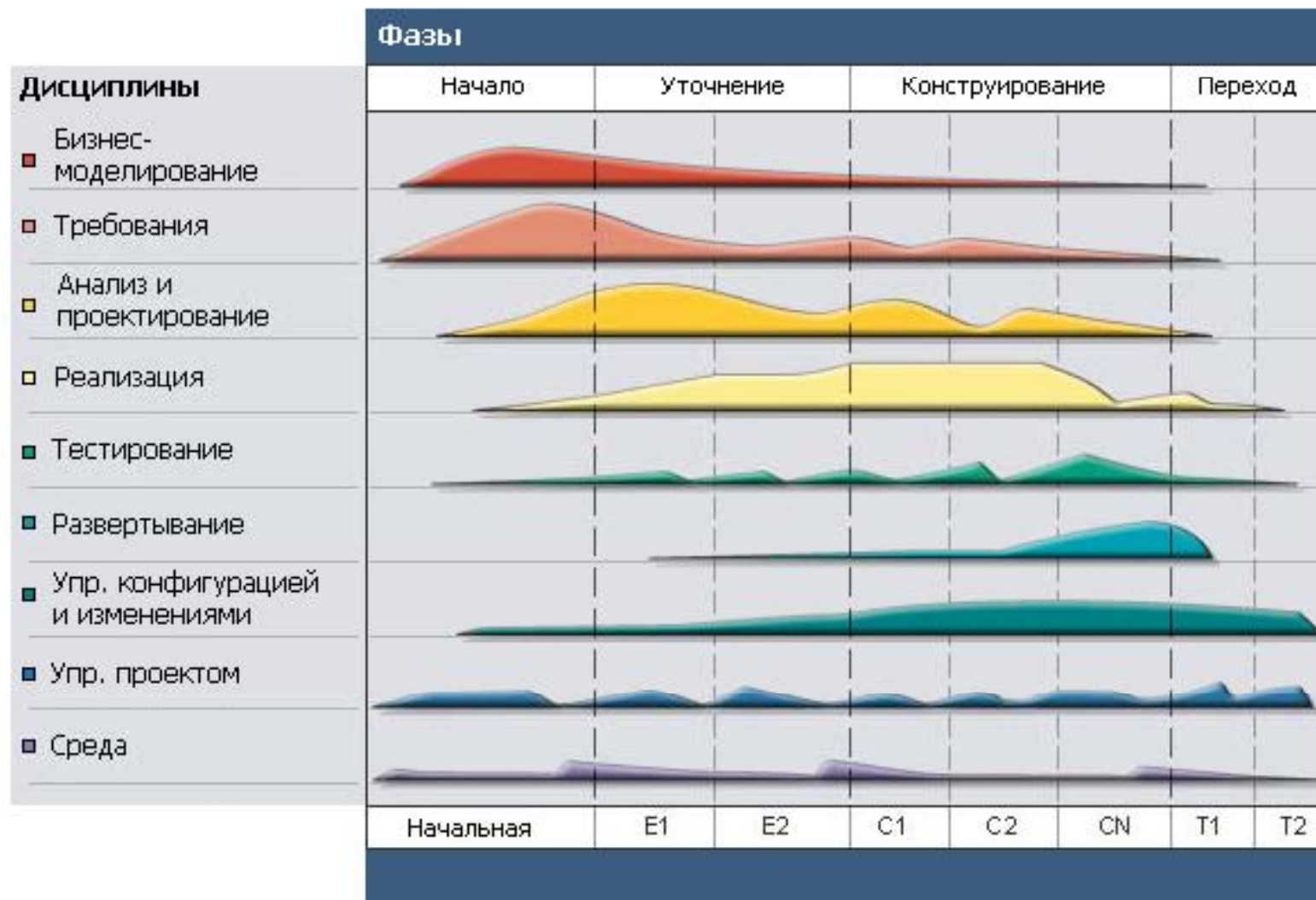
RUP – процесс разработки от Rational.

RUP – упорядоченный подход к решению задач и обязанностей, а также управлению ими.

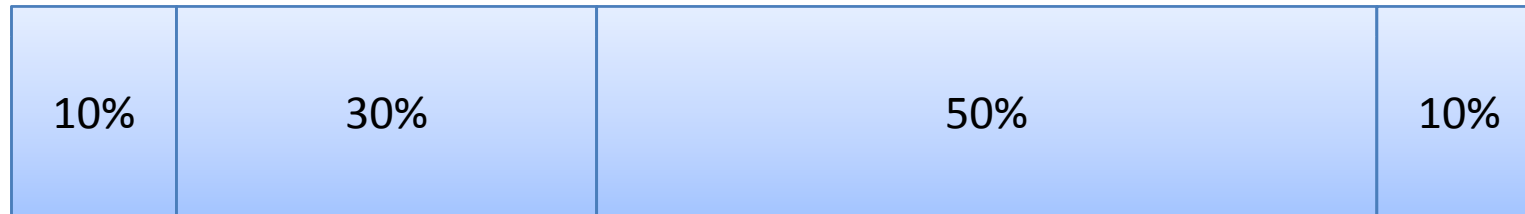
Принципы RUP:

- Разрабатывайте итеративно
- Управляйте требованиями и изменениями
- Пользуйтесь модульными архитектурами
- Используйте визуальное моделирование
- Не забывайте о проверке качества
- Распределяйте роли в команде

Технологические процессы разработки



Предлагаемое время итераций RUP



Исследование

Проектирование

Разработка

Развертывание

Исследование:

1. Понять что создавать
2. Определить ключевые функции
3. Определить хотя бы одно возможное решение
4. Оценить стоимость, сроки, риски
5. Определить процесс достижения цели

Проектирование:

1. Уточнить требования
2. Спроектировать, реализовать архитектуру
3. Минимизировать риски
4. Уточнить прецедент разработки, определить среду

Разработка:

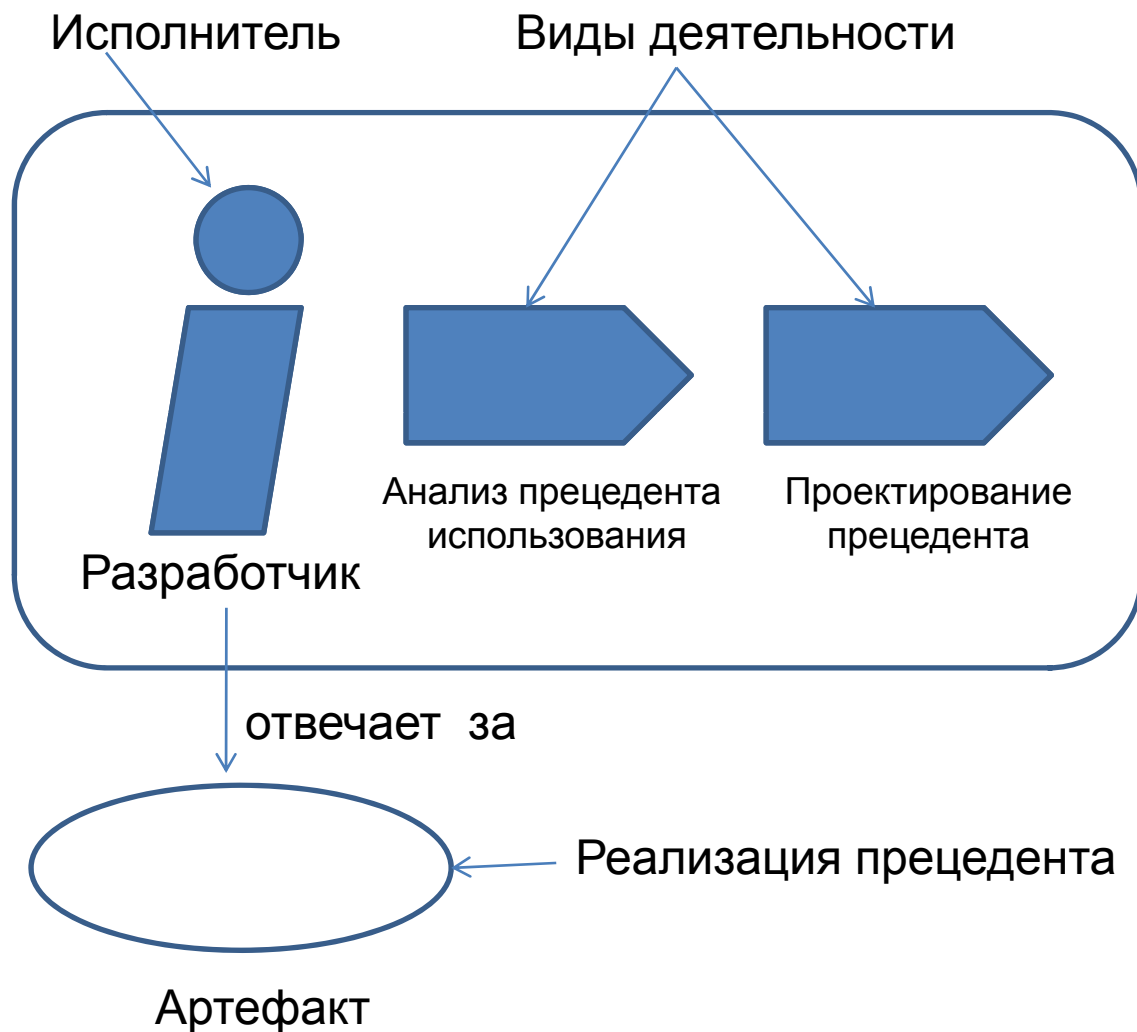
1. Снизить стоимость разработки и добиться параллелизма
2. Разработать продукт

Развертывание:

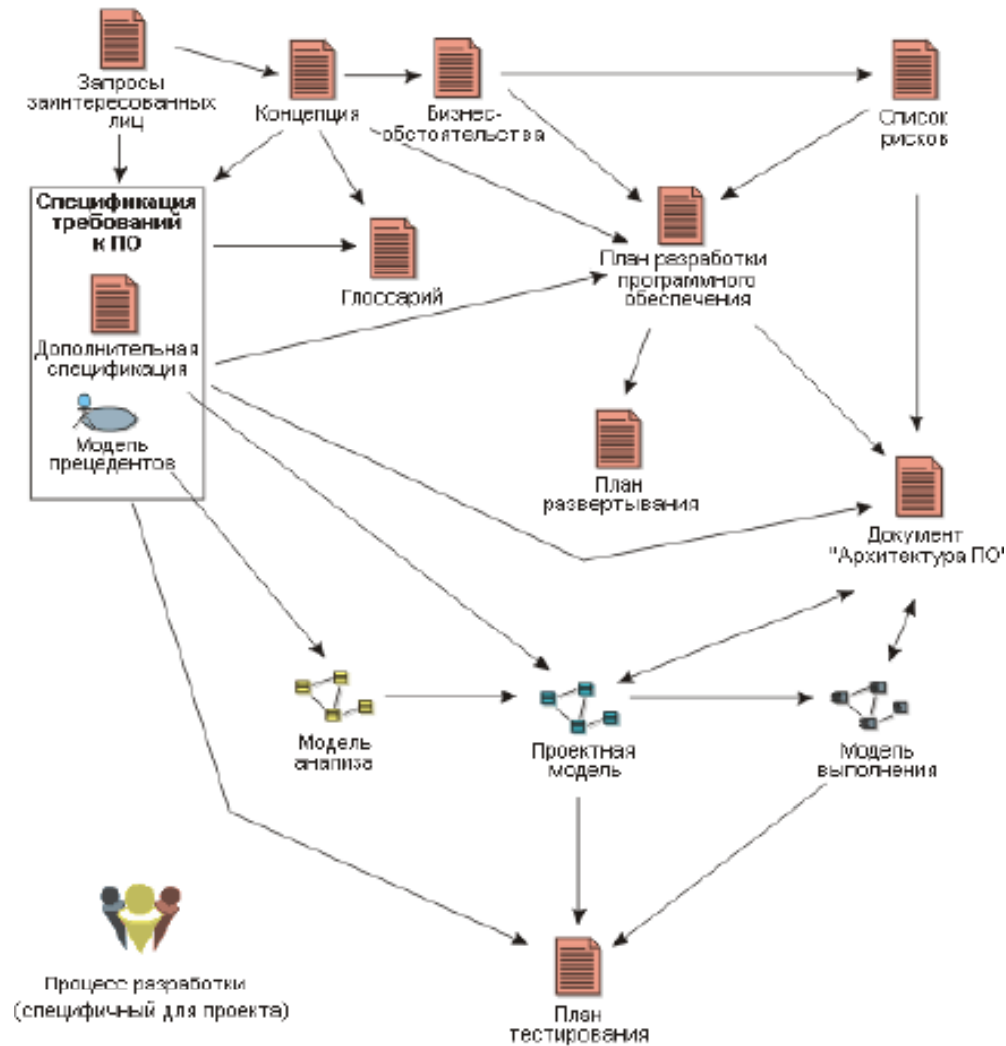
1. Бета-тестирование
2. Обучить пользователей
3. Подготовить площадку для развертывания
4. Приемочное тестирование

Ключевые понятия RUP

- **Исполнители:** *кто*
- **Виды деятельности:** *как*
- **Артефакты:** *что*
- **Технологические процессы:** *когда*



Артефакты RUP



Артефакты:

- Документы
- UML-модели
- Исходный код

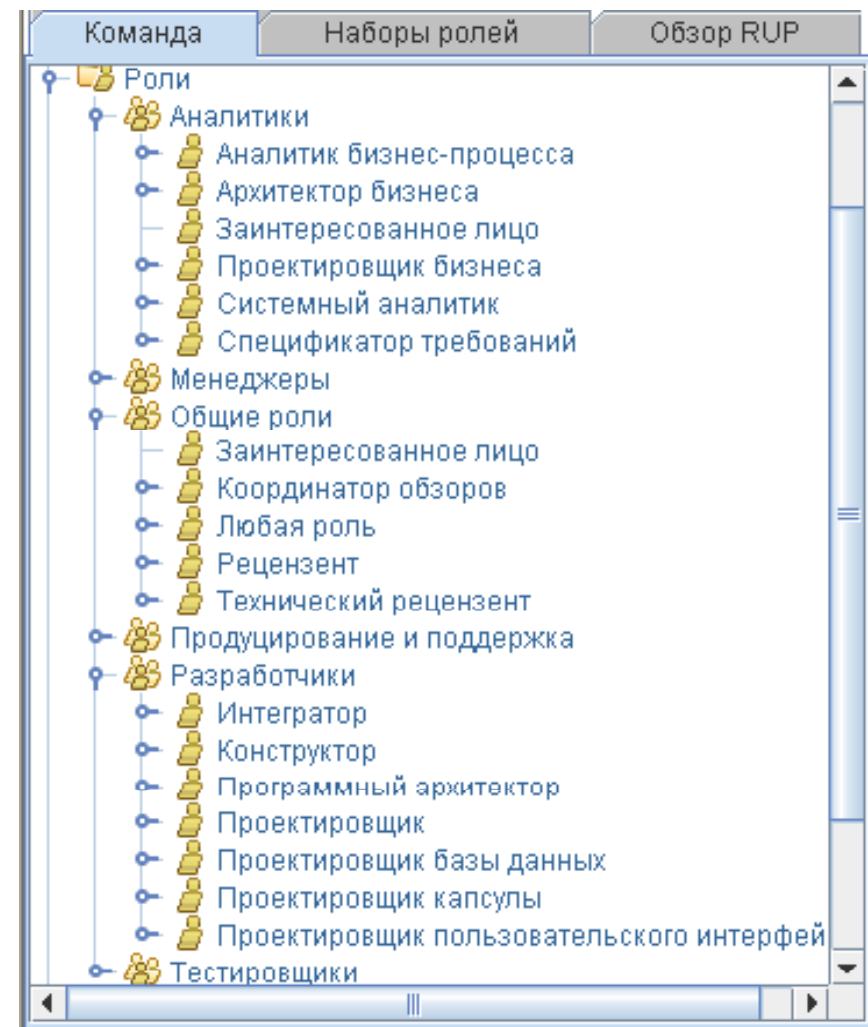
Каждый член проектной группы создает и использует свои артефакты

Роли RUP

Роль - набор ответственностей

Группы ролей:

- Аналитики
- Менеджеры
- Разработчики
- Тестировщики
- Поддержка
- Общие роли



Контур «своего» процесса

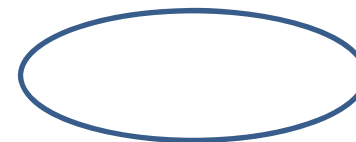
Необходимо выбрать:



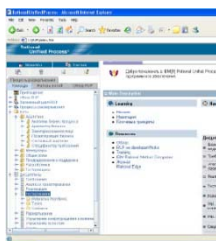
Роли



Деятельности

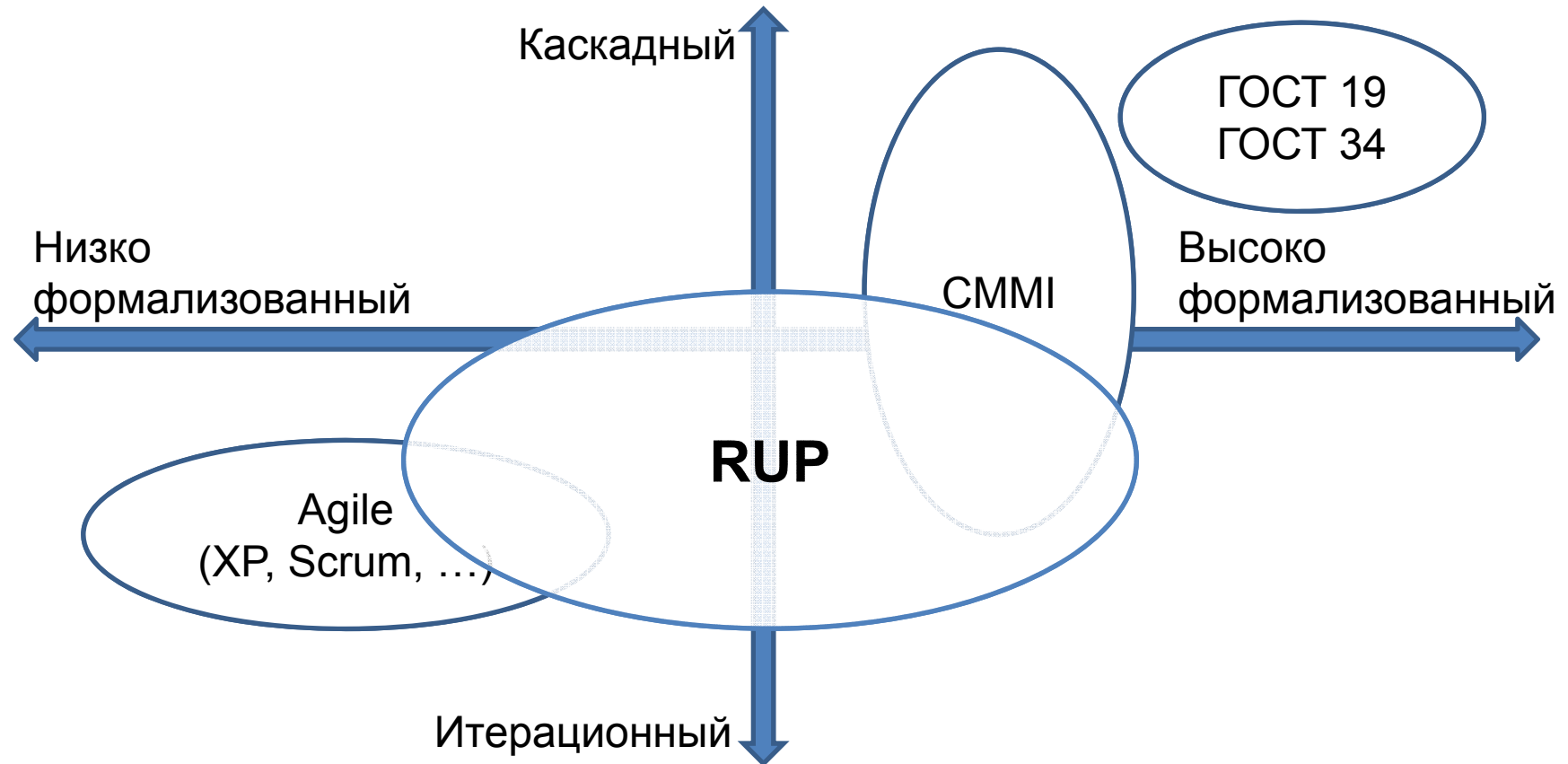


Артефакты



Инструментарий

Сравнение RUP с другими методологиями



Основа сравнения – различный уровень формализма и требования к итерациям

Сравнение RUP и Agile

Agile	Общее	RUP
Паттерны проектирования, стандарты кода	Активное общение между участниками	Стандарты моделирования (UML)
Экстремальное программирование	Итерации	Артефакты, модели
Взаимодействие лицом к лицу и минимум документации (в т.ч. с заказчиком)	Быстрая реакция на запросы заказчика, изменения в требованиях принимаются на всех этапах проекта	Распределение ролей
TDD	Непрерывная интеграция, частый выпуск релизов	Четкие метрики оценки качества результата
Самоорганизующиеся команды	Объектно-ориентированное проектирование	
В каждой итерации выпускается рабочий продукт	Рефакторинг	

Результат сравнения RUP и Agile

- Agile – это не методология, а набор практик
- Agile проще, чем RUP:
 - RUP требует более глубокой проработки задач перед кодированием
 - RUP требует большей формализации
- Проект Agile дешевле
- Проект RUP менее рискован
- Работающий код – в RUP не единственный результат
- в RUP больше метрик для оценки качества результата

А сколько формализма нужно?

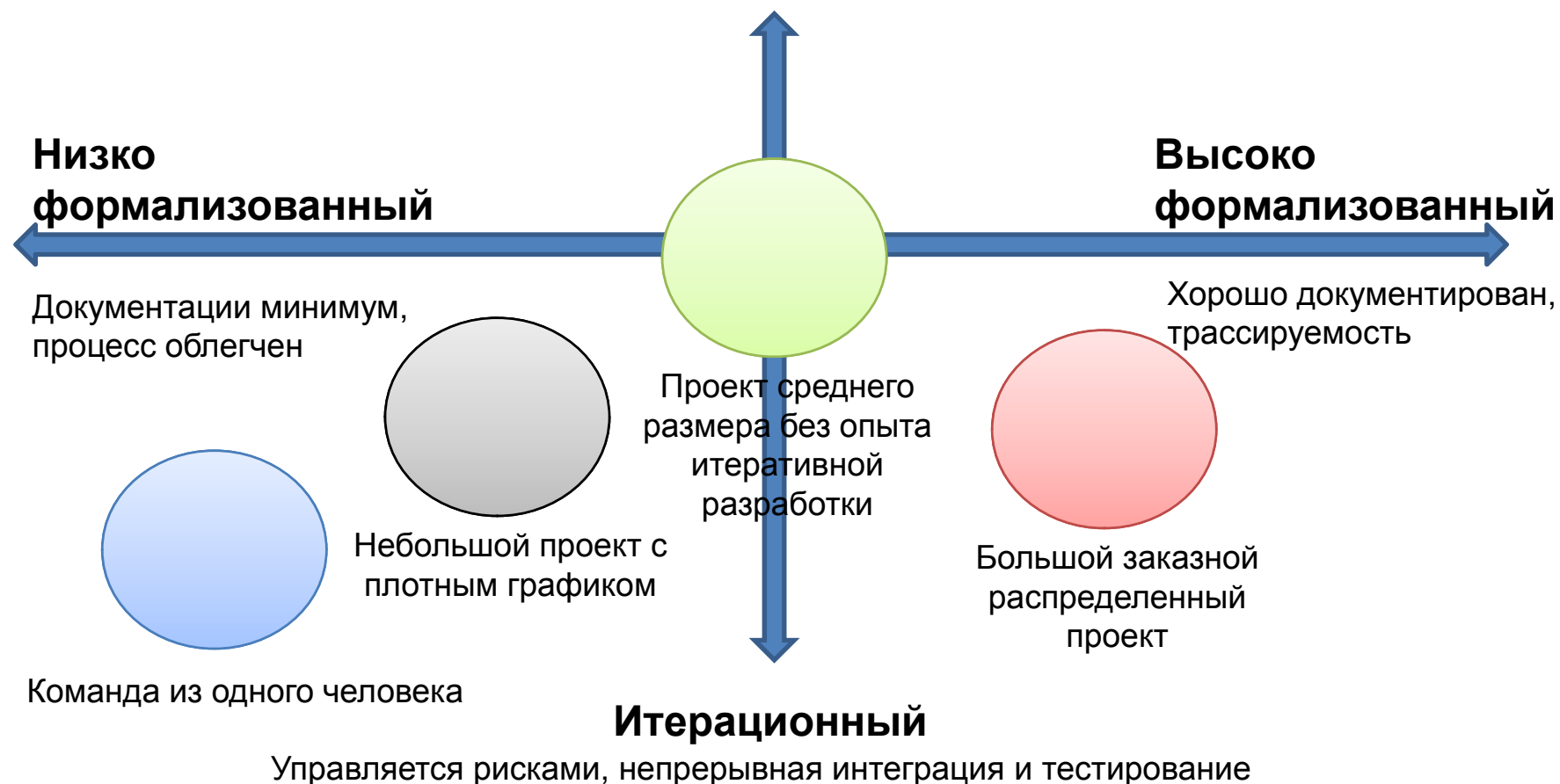
Основные факторы, влияющие на оптимальный уровень формализации процесса:

- Масштаб проекта
- Критичность проекта
- Распределение участников
- Новизна проекта
- Требования заказчика (заказной / свободный проект)
- Ожидаемая долговечность проекта

Пример выбора методологии

Мало внимания рискам, последовательная разработка, поздняя интеграция и тестирование

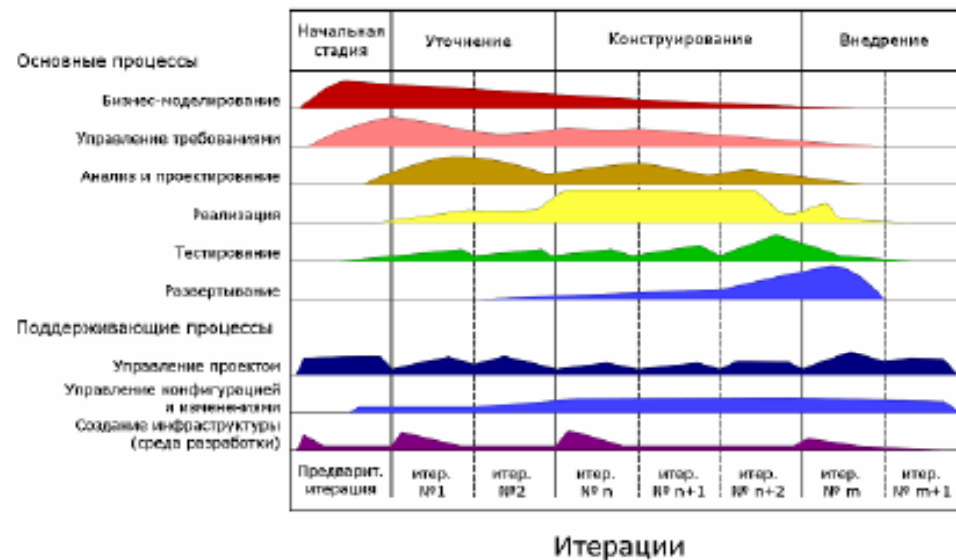
Каскадный



Инструментарий RUP

Бизнес-моделирование Rational Software Architect
Управление требованиями Rational RequisitePro, ClearQuest
Анализ и проектирование Rational Software Architect, ClearQuest
Реализация Rational ClearCase, SoDA
Тестирование Rational TestManager, Robot, PurifyPlus, Functional Tester, Performance Tester
Развертывание
Управление проектом MS Project
Управление конфигурацией Rational ClearCase

Рабочие процессы



Основа – база знаний RUP

Пример применения не-Rational инструментария в RUP

JIRA,
Track

Формирование требований,
Управление запросами

MS Project

Управление проектом

Ant, Maven, ...

Управление сборкой

Subversion

Управление версиями

Wiki,
Doxygen,
Шаблоны MS
Word

Подготовка документации

Enterprise
Architect,
MS Visio,
Visual Paradigm

Проектирование

Практика применения

Шесть основных этапов перехода к RUP:

- Оценить текущее состояние организации
- Выбрать методологию, проанализировать риски
- Сформировать рабочую группу, обучить сотрудников
- Организовать «свой» процесс (сконфигурировать RUP)
- Выбрать, настроить инструментарий
- Реализовать процесс на небольшом несложном проекте

Еще раз о RUP

1. RUP – методология разработки программного обеспечения + база знаний, но не набор программных средств
2. RUP отличается от других методологий (Agile, XP, FDD) уровнем формализованности процесса разработки – процессы, роли, артефакты
3. RUP использует итеративную модель разработки
4. RUP позволяет достичь пятого уровня зрелости по CMM
5. RUP поддерживается UML (используется для создания и формализации большинства артефактов)
6. Артефакты RUP позволяют достичь единства понимания основных автоматизируемых процессов
7. RUP позволяет организовать «свой» процесс

