

Дисциплина

СПЕЦИФИКАЦИЯ, ПРОЕКТИРОВАНИЕ И АРХИТЕКТУРА ПРОГРАММНЫХ СИСТЕМ

Тема 2

РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ НА ОСНОВЕ МОДЕЛЕЙ

Преподаватель

к.т.н. Романенко Сергей Александрович

Особенности программного проекта

1. От неформального к формальному
2. Творческий характер
3. Стремление к повторному использованию

Причины неудачных проектов

1. Недостаточно адекватное управление требованиями
2. Необоснованность и недостаточная документированность проектных решений
3. Несогласованность требований, дизайна и реализации
4. Неточные и противоречивые коммуникации
5. Недостаточное тестирование
6. Игнорирование рисков и отсутствие процедур управления рисками
7. Бесконтрольное внесение изменений в артефакты проекта
8. Недостаточное использование CASE-средств и средств поддержки отдельных этапов проекта

Model Driven Development (MDD)

Модель - это описание системы с конкретной точки зрения, которое допускает пропуск несущественных деталей, чтобы интересующие характеристики были видны наиболее отчетливо.

Процесс разработки ИС на основе моделей - процесс построения и последовательного преобразования ряда согласованных моделей на всех этапах жизненного цикла ИС, связанный с принятием решений.

Особенности MDD

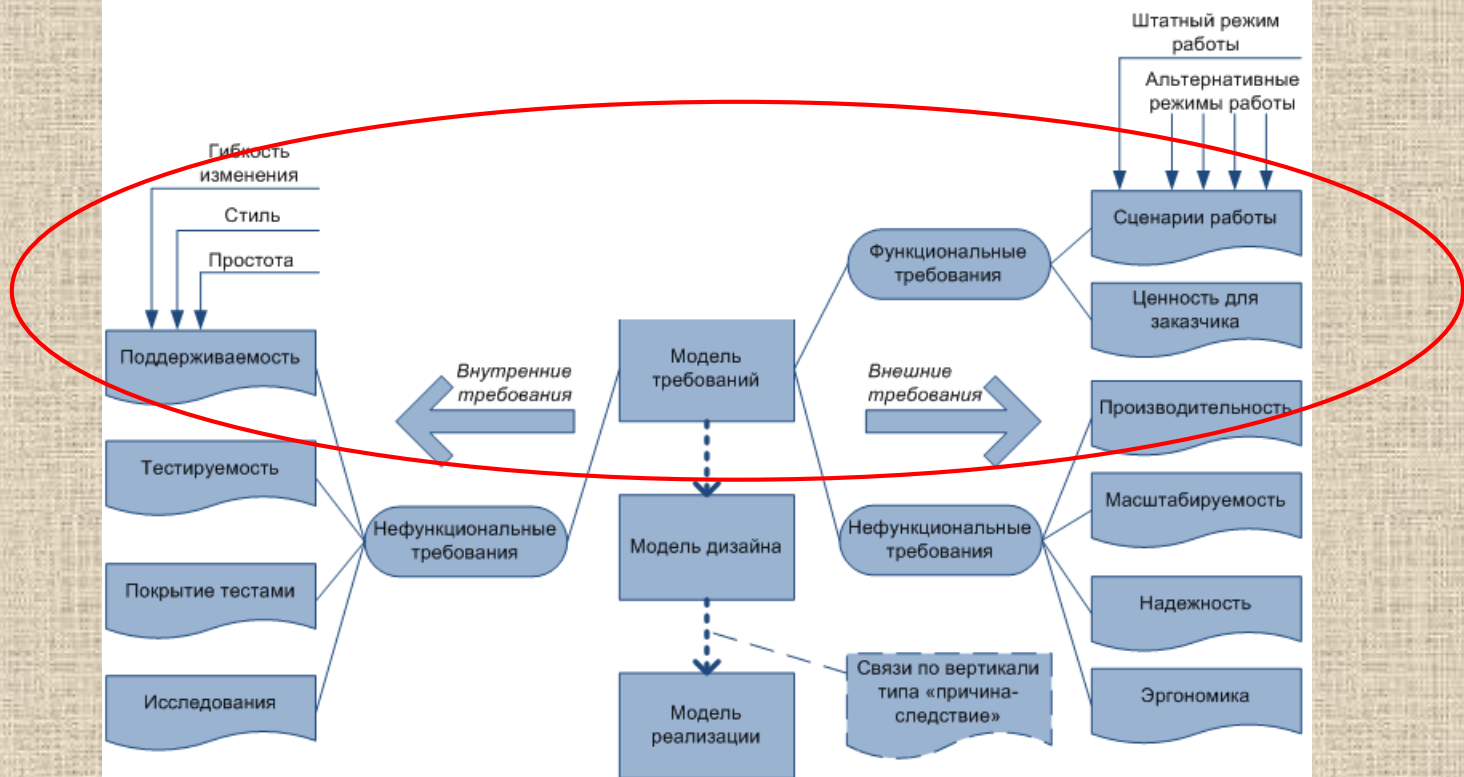
Разработка, управляемая моделями, — это такой стиль разработки программ, когда главными артефактами являются модели, а по ним генерируется код и другие прикладные артефакты.

1. Модели как наброски и чертежи (обычно так, но надо больше).
2. Точные модели позволяют осуществлять автоматизацию.
3. Определяющая роль шаблонов в процессе разработки.
4. Не только код.

Преимущества MDD

- ✓ Повышение производительности команды.
- ✓ Удобство обслуживания системы.
- ✓ Повторное использование унаследованных компонент.
- ✓ Адаптируемость.
- ✓ Согласованность решений.
- ✓ Фиксация и повторяемость успешного опыта.
- ✓ Совершенствование общения.
- ✓ Модели как долгоживущие активы.
- ✓ Возможность отсрочки технологических решений.

Принятие решений на основе моделей

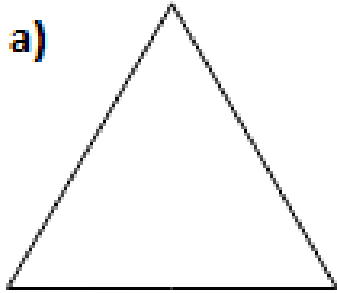


ВАЖНО!

- ✓ Все, что не определено требованиями, становится выбором проектировщика.
- ✓ Метафора системы определяет способ описания требований.
- ✓ Большинство (99,9%) проблем в программной инженерии связано с неопределенностью.
- ✓ Решения на любом уровне моделей должны быть обоснованы требованиями предыдущего уровня.
- ✓ Требования противоречивы

**Проектирование (design) – это разрешение конфликтов, т.е. нахождение компромиссов.
В любом случае за каждое решение приходится чем-то платить.**

Фрактальная природа проекта



Разработчикам необходимо ограничивать количество вариантов, которые они реализуют, но при этом необходимо иметь ввиду, что попадание в сценарий, который не был реализован, называется сбоем или отказом программы.

а) – штатный сценарий работы системы

б) – альтернативный сценарий работы

в) - альтернативы альтернатив

....

При сохранении ограниченного пространства
периметр стремиться к бесконечности

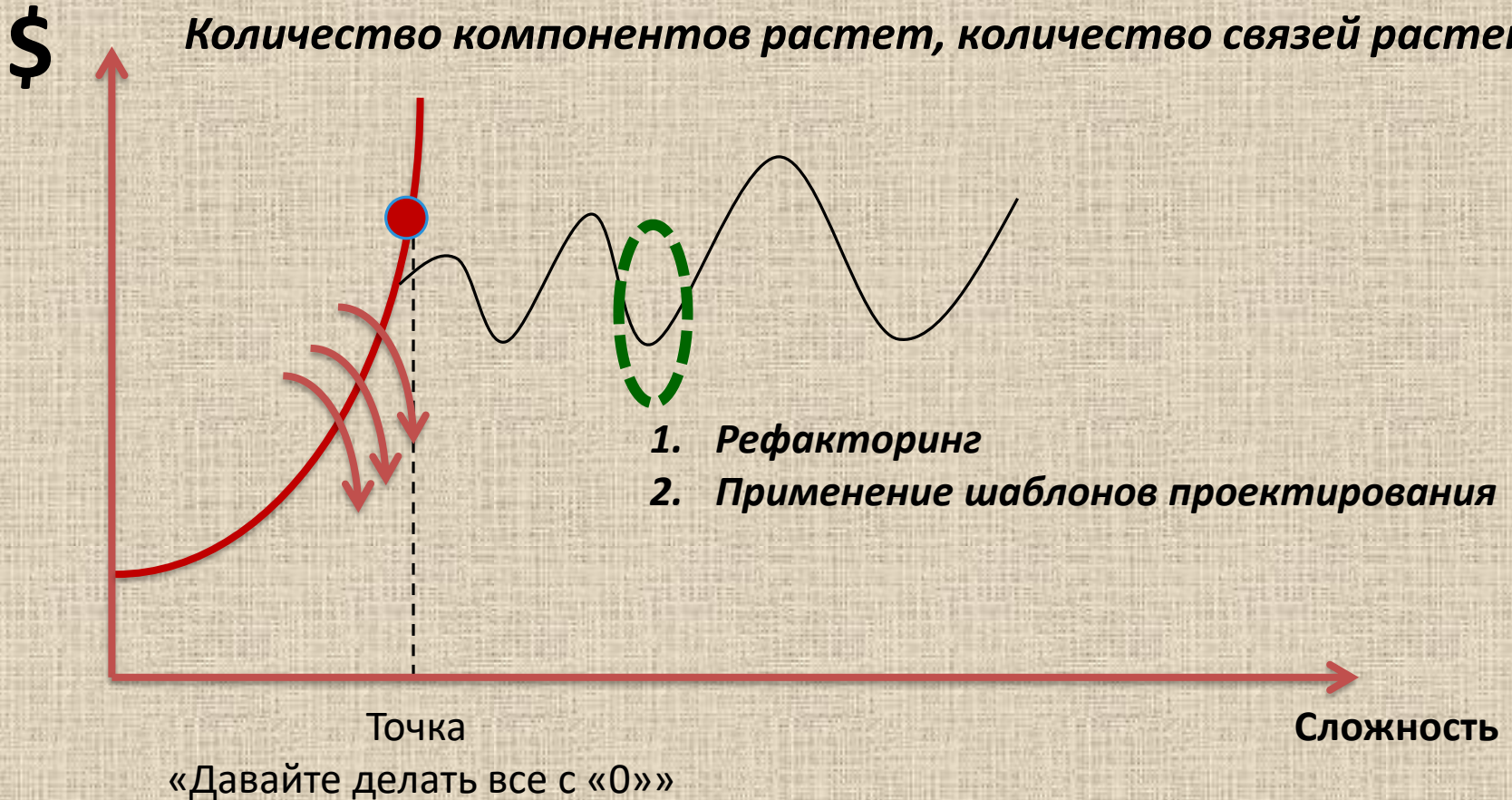
Конфликты и взаимосвязи требований

Пример ключевого конфликта: Простота vs Гибкость

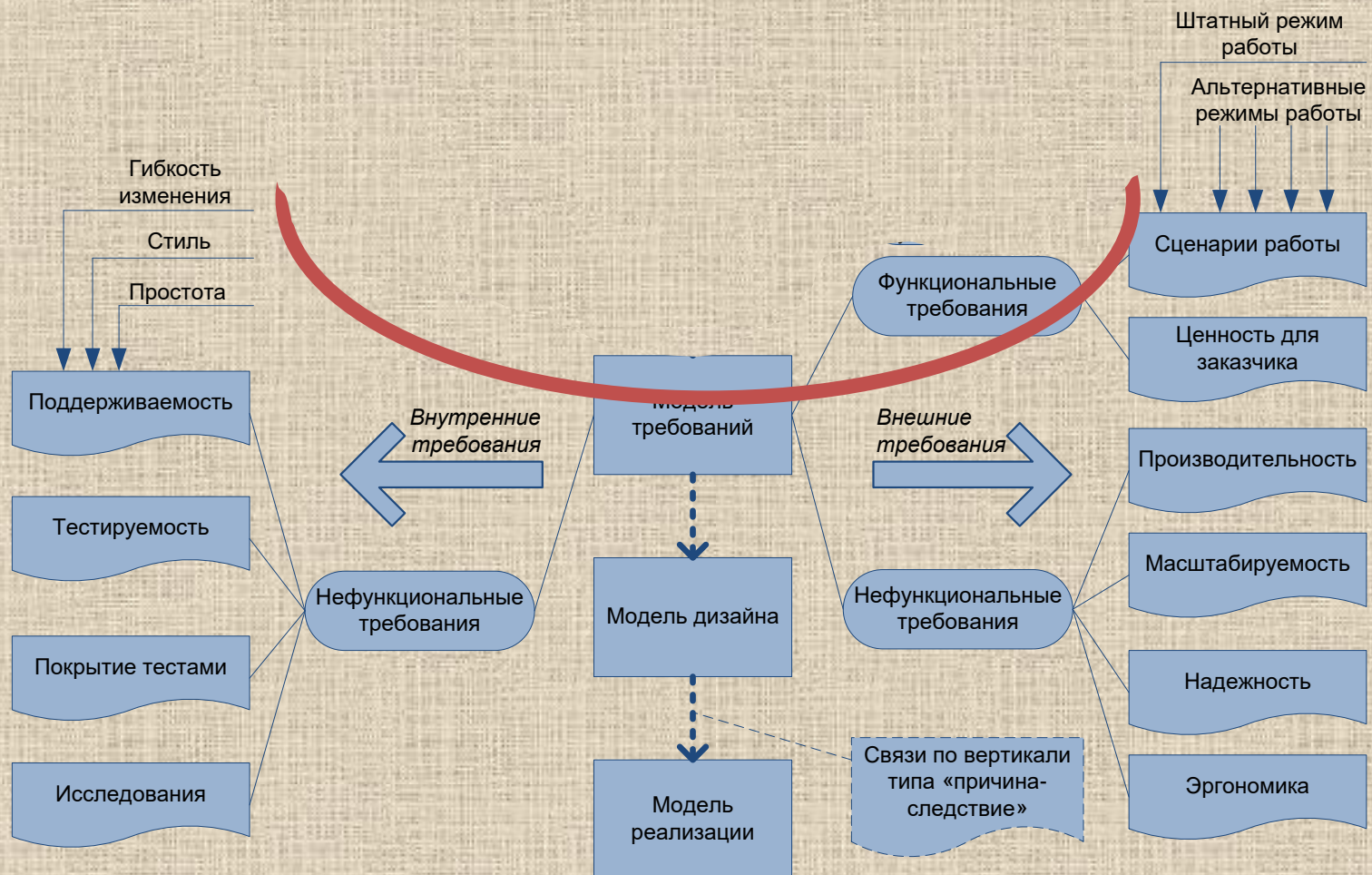
Для разрешения конфликтов требований необходимо определить **«ось вариативности»**, т.е. будет или нет изменяться конкретное требование и использовать либо гибкую модель дизайна, либо простую.

Взаимосвязь сложности программной системы и стоимости владения

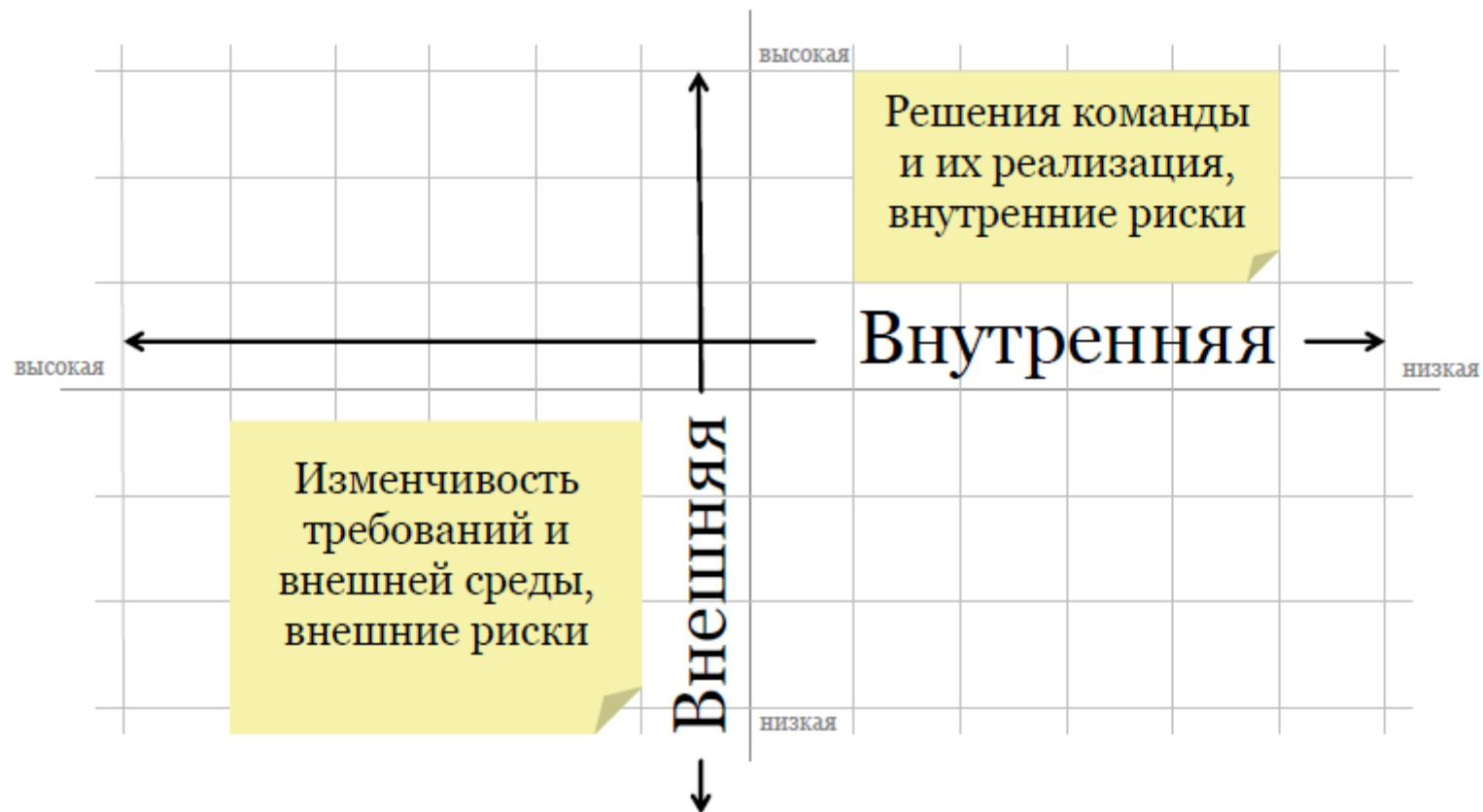
**Программная система – это компоненты и связи.
Количество компонентов растет, количество связей растет.**



Что делать, если требования не полны?

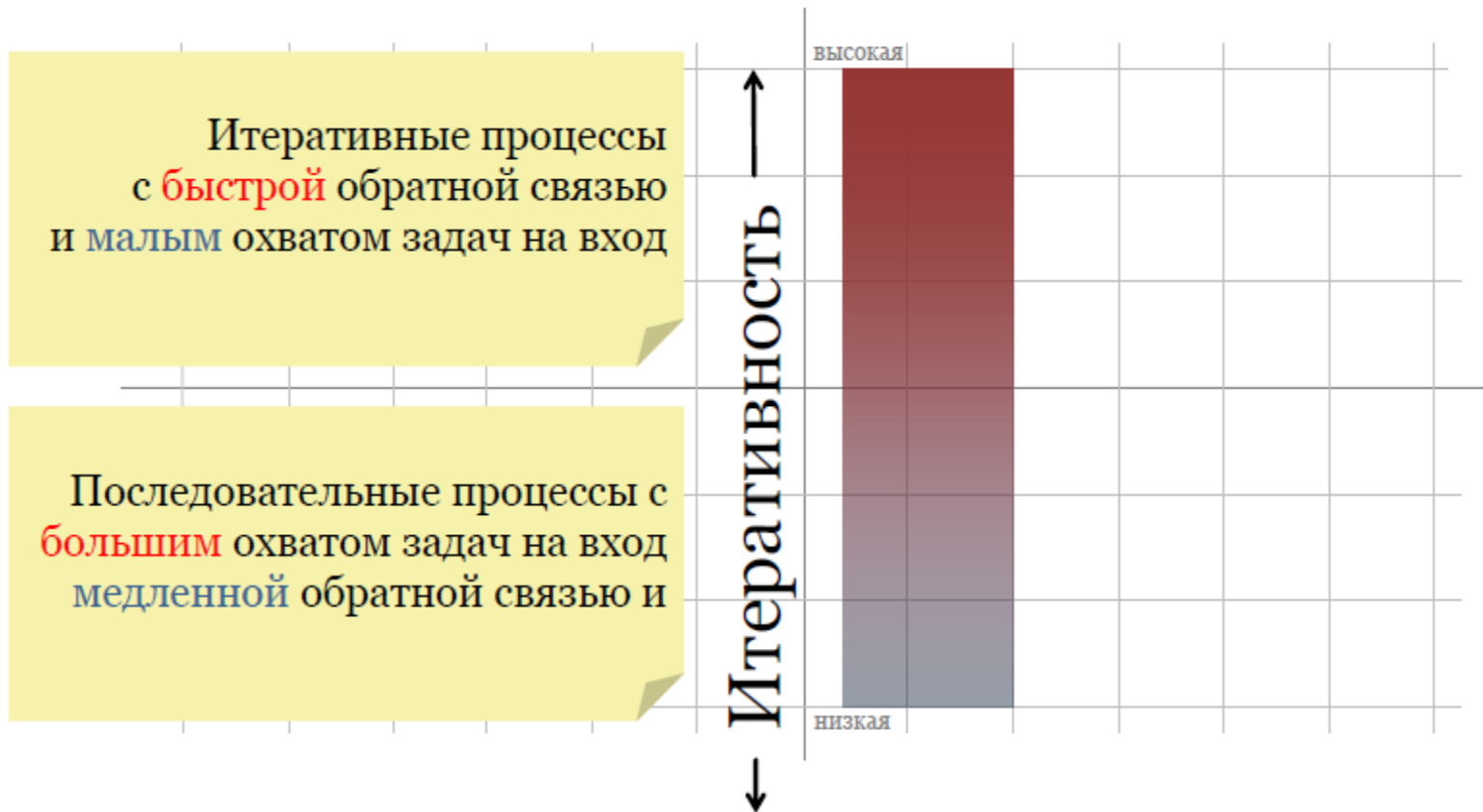


Характер неопределенности



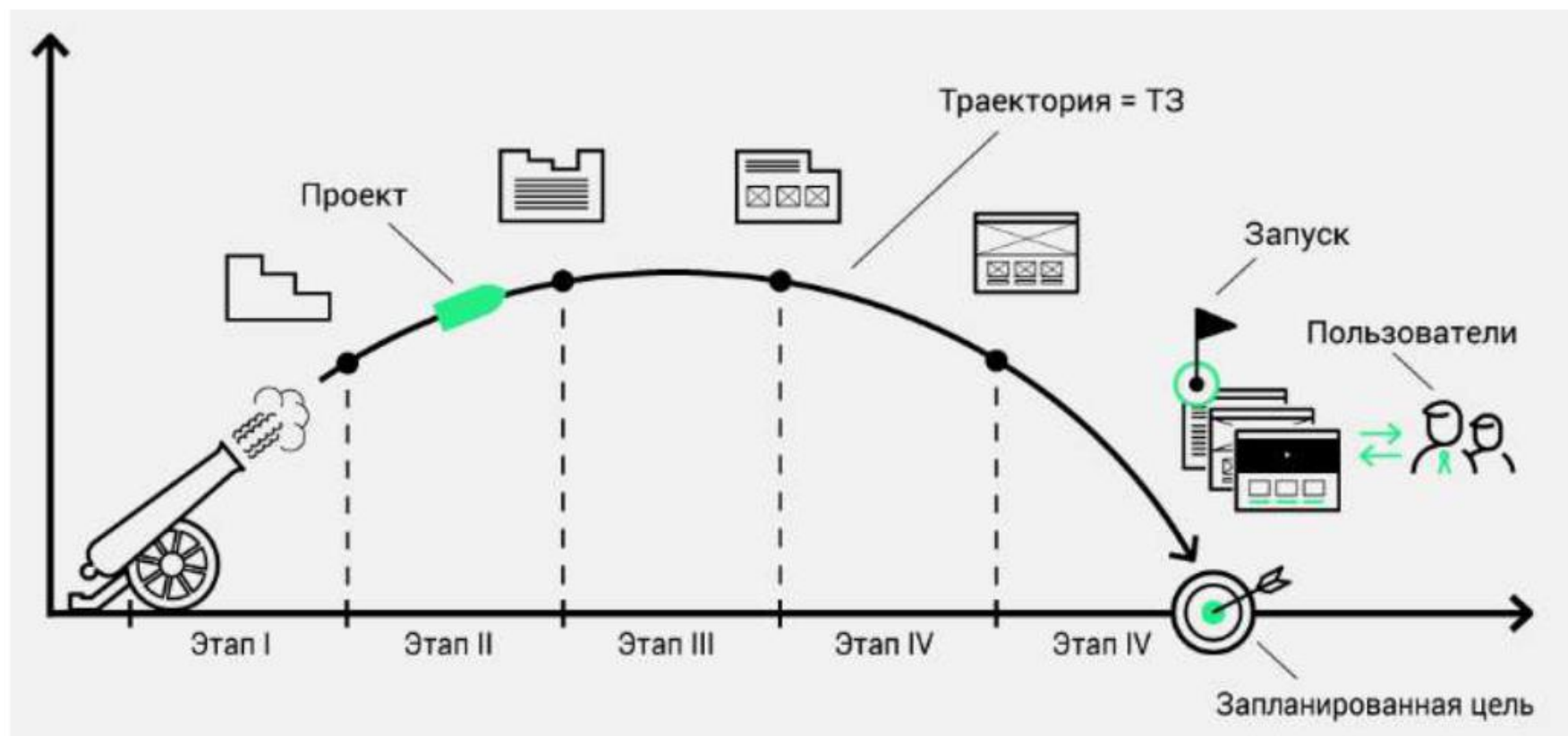
Выбор модели процесса разработки

БОРЬБА С ВНЕШНЕЙ НЕОПРЕДЕЛЕННОСТЬЮ



«Баллистическая» система

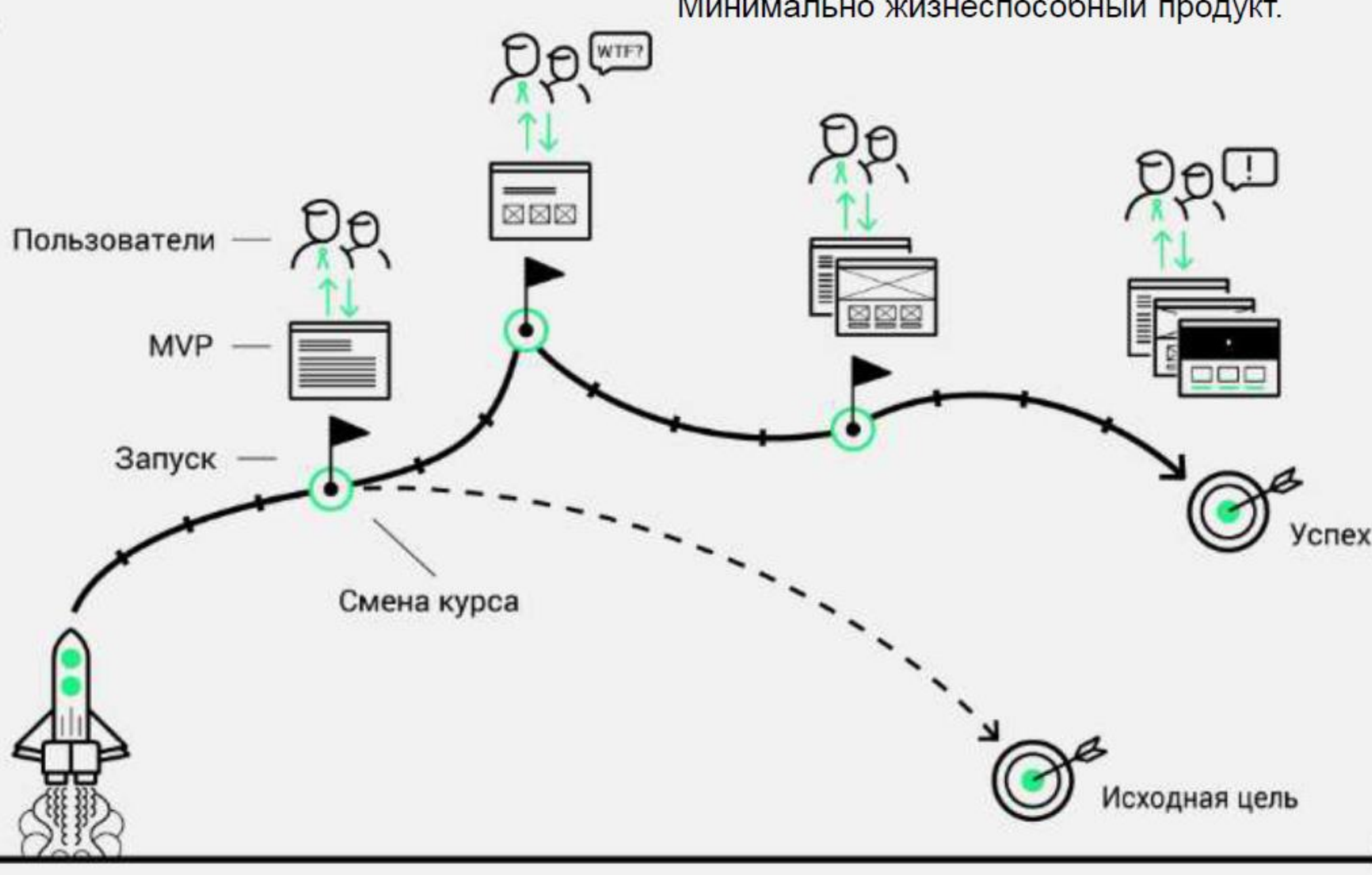
Водопад



Адаптивные циклы

MVP - Minimum Viable Product

Минимально жизнеспособный продукт.



Основные типы жизненных циклов

Предиктивный

ТЗ

Создание

Тестирование

Инкрементный



AGILE

Итеративный



Классический Scrum

Роли

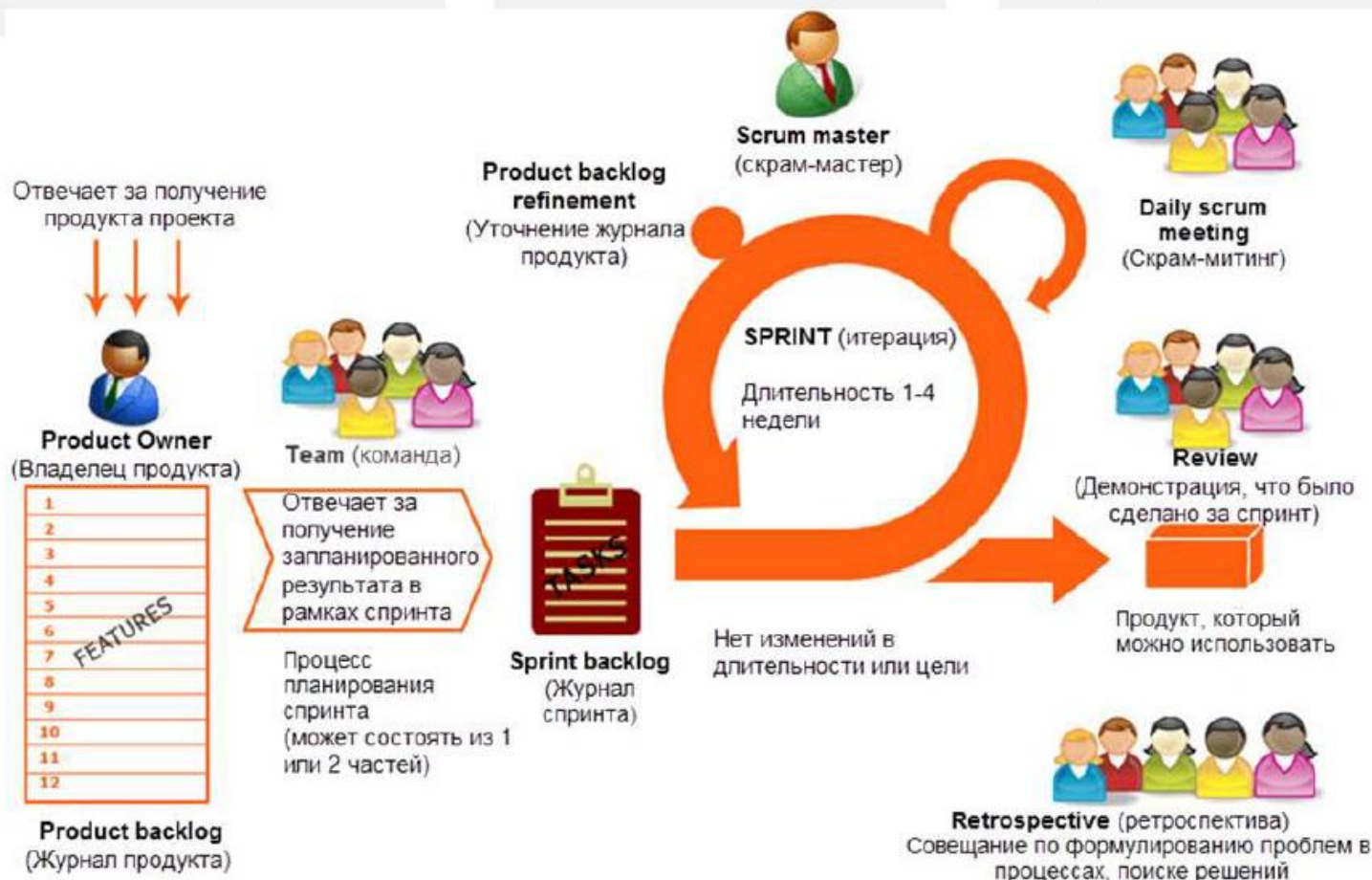
- Владелец продукта.
- Скрам-мастер.
- Команда разработки

Артефакты

- Журнал пожеланий продукта.
- Журнал пожеланий спринта.
- Инкремент продукта

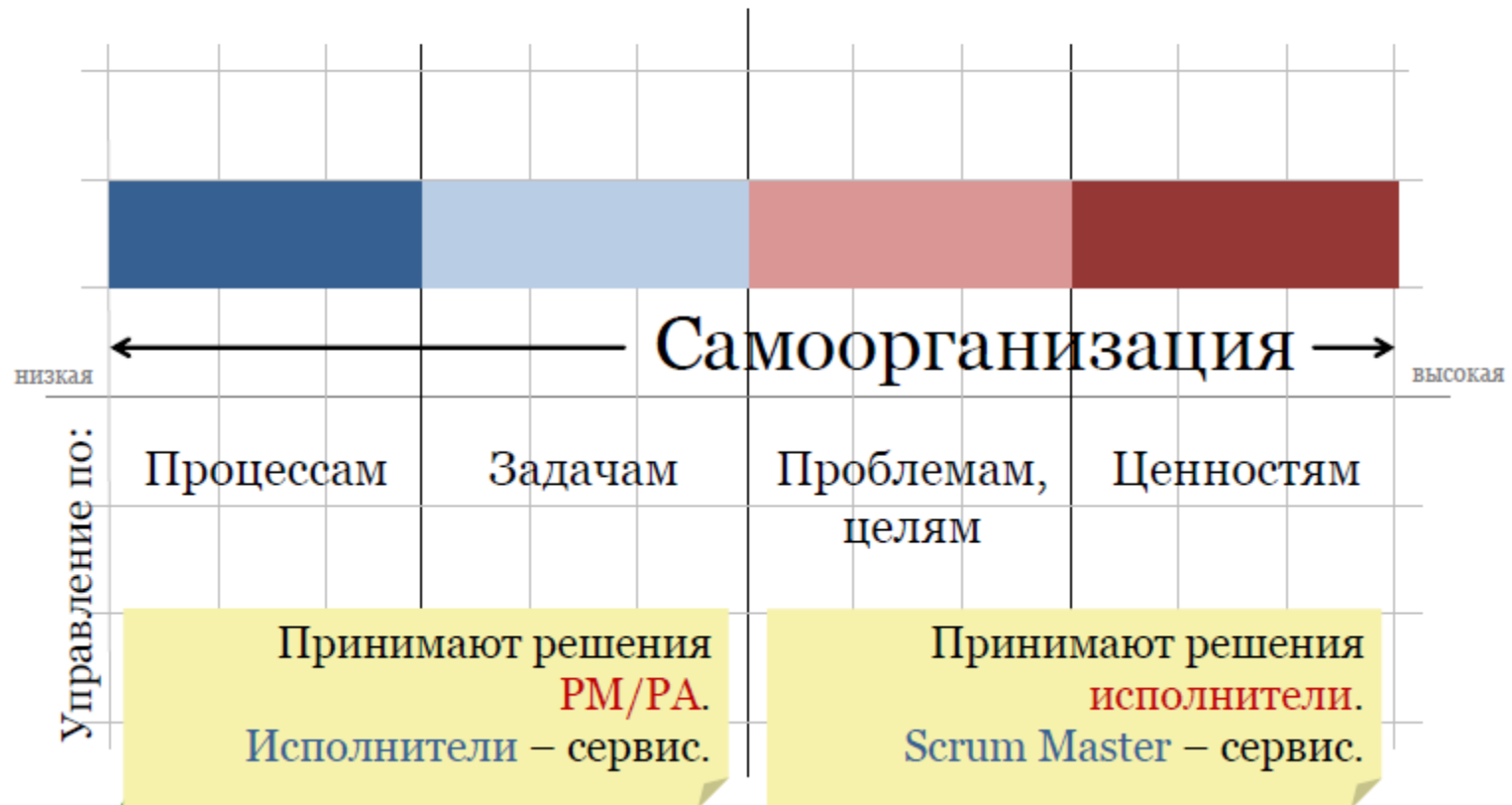
Процессы

- Планирование спринта.
- Обзор спринта.
- Ретроспектива.
- Скрам-митинг.
- Спринт

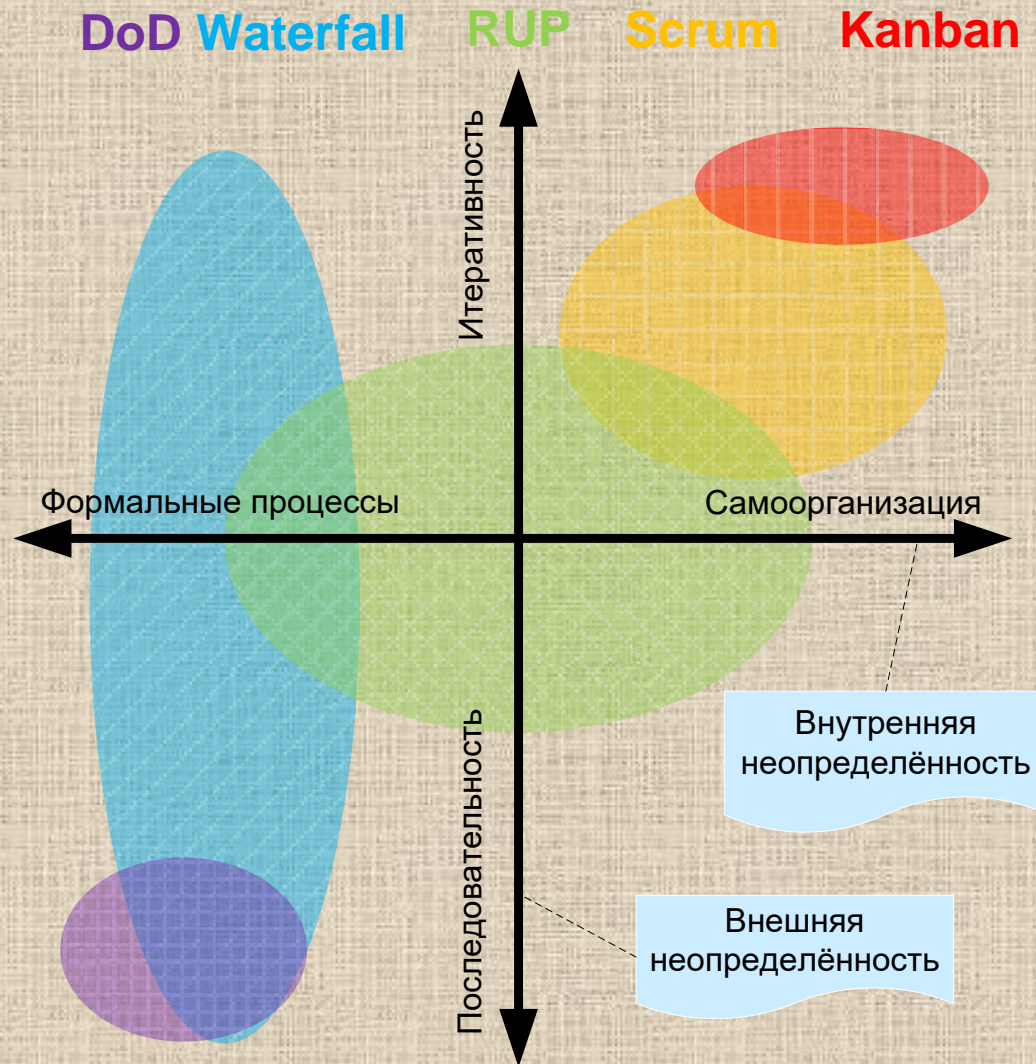


Выбор модели проектной команды

БОРЬБА С ВНУТРЕННЕЙ НЕОПРЕДЕЛЕННОСТЬЮ



Типовые методологии



Модели разработки

BDUF - Big Design Up Front
(проектирование всего до разработки)

Последовательные
процессы

- 1. Большой охват задач
 2. Долгая обратная связь
 3. Возрастает цена ошибки на ранних стадиях

YAGNI - You didn't gonna need it
(вам это не понадобится, делаем проще и меняем, рефакторинг)

Итеративные
процессы

- 1. Нарушение целостности
 2. Большие накладные расходы
 3. Накопление технического долга

Бизнес-модели компаний разработчиков

Вид разработки

Продуктовая

Заказная

Услуги

Вид заказчика

B2B

B2C

B2G

Уровень зрелости

Start-Up

Зрелая компания

Контекст продукта

Для внутреннего
использования

Оффшорная разработка

Интеграционная
разработка

Заказная разработка

Валидация и верификация моделей

Важно!

На всех уровнях должна быть **модель документирования**, которая определяет, как конкретно осуществляется описание модели, а также **модель тестирования**. Последняя должна позволять обосновать модель с учётом модели более высокого уровня.

Верификация – проверка на соответствие внутренней системе аксиом, проверка на соответствие нашим ожиданиям.

Валидация – проверка на соответствие внешней системе аксиом, проверка на соответствие ожиданиям заказчика

Заключение

1. **Проектирование** – разрешение конфликтов требований, поиск компромиссов.
2. **Решение** – это всегда **компромисс**, задача многокритериальной оптимизации. За каждое решение приходится чем-то платить.
3. Невозможно решить проблему на том же уровне абстракции, на котором она возникла.
4. Необходимо обеспечивать полноту и непротиворечивость моделей.
5. Любое решение должно быть обосновано в системе критериев, заданных системой моделей. **Модель** каждого уровня **должна быть обоснована** требованиями **предыдущего уровня**.
6. **Шаблон** – принятое ранее решение, готовое к повторному использованию.
7. Современная разработка программного обеспечения – групповая борьба с **неопределенностью** (внешней и внутренней).
8. Типовые процессы определяют степени **итеративности** и **самоуправления** проектной команды.

Принятие решений на основе моделей

