

# Технологии программирования

Лекция 12

Методологии разработки ПО

Старичков Н.Ю.

# ЛИКБЕЗ: МЕТОДОЛОГИЯ РАЗРАБОТКИ ПО

- Подход к организации работы (проектирования, разработки, тестирования, ...)
- Выбор методологии зависит от многих параметров (как внутренних, так и внешних)
  - Особенности проекта
    - Сроки
    - Бюджет
    - Вариант поставки
    - Требования к качеству
    - ...
  - Особенности команды
  - Особенности заказчика

# Внешние факторы

# ВНЕШНИЕ ФАКТОРЫ

← проблема в том, что на эти факторы мы не можем влиять!

- Форс-мажоры (природные явления, пандемия и т.п.)
- Государственные органы и их распоряжения
  - Регламентирующие
  - Запрещающие
  - Сертификация, ее необходимость и ее правила
- Действия конкурентов
  - Экономические
  - Управленческие / «бизнесовые»
  - Содержательные

# Внутренние факторы

# ВНУТРЕННИЕ ФАКТОРЫ (ПРОЕКТА)

- Особенности команды

- Роли

- Квалификация


- Размер команды

- Человеческий фактор

- Особенности организации управления и принятых регламентов

- Особенности принятых правил взаимодействия с заказчиком

*этими можно управлять  
в некотором объеме*



# ВНУТРЕННИЕ ФАКТОРЫ (ПРОЕКТА)

- Особенности команды
- Особенности организации управления и принятых регламентов
  - Организация работы внутри структурных единиц ← *тоже в нашей власти*
  - Организация взаимодействия с руководством
  - Организация горизонтального взаимодействия ) ← *зависит от нашей роли в компании*
- Особенности принятых правил взаимодействия с заказчиком

# ВНУТРЕННИЕ ФАКТОРЫ (ПРОЕКТА)

- Особенности команды
- Особенности организации управления и принятых регламентов
- Особенности принятых правил взаимодействия с заказчиком
  - Принятое отношение с заказчиком ← *зависит от нашей роли в компании*
  - Выбранные приоритеты при взаимодействии с заказчиком ← *частично*
  - Уровень взаимодействий и их степень регламентированности
  - Человеческий фактор со стороны заказчика ← *практически все нашей власти*



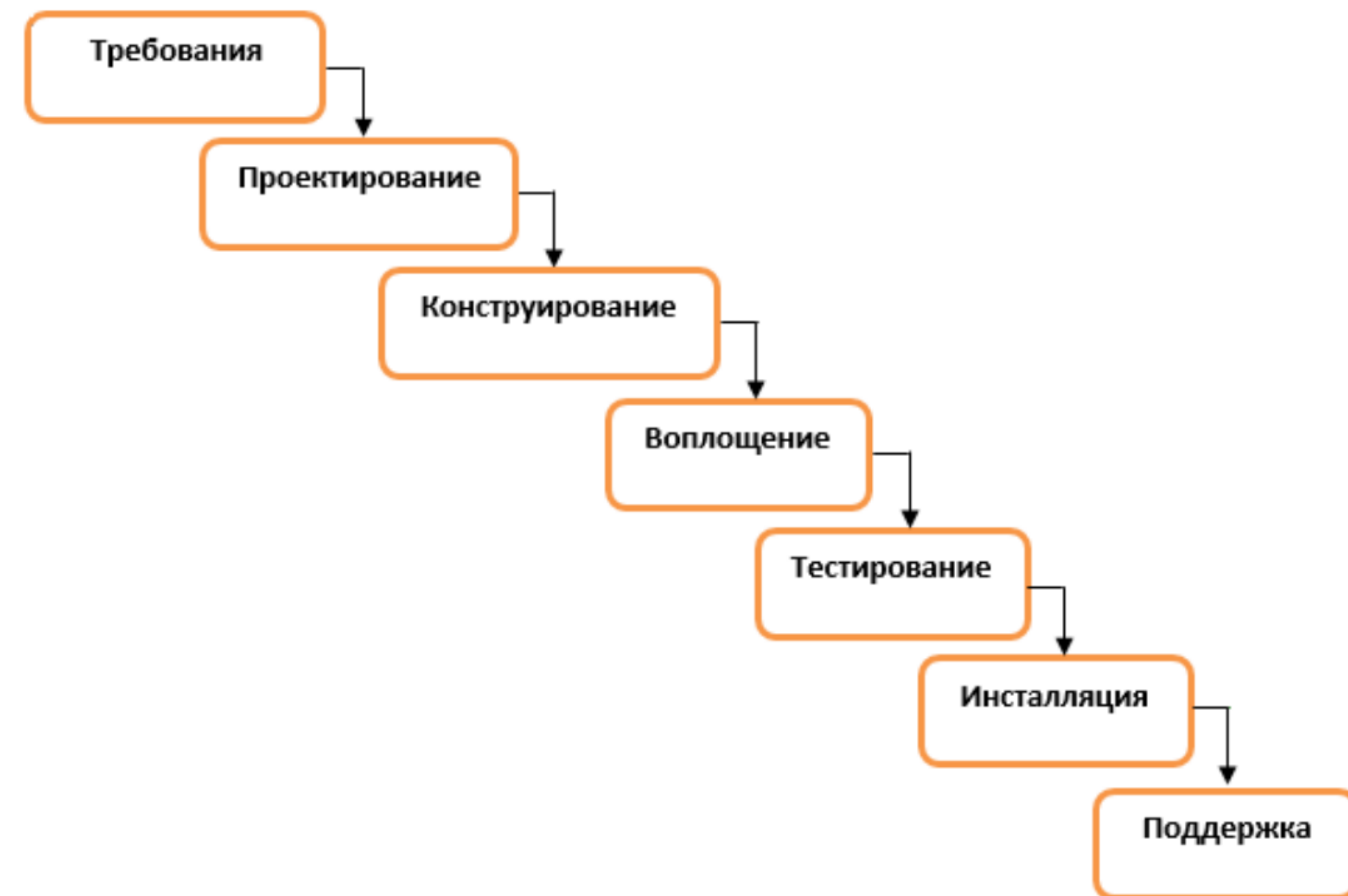
# ПРИМЕРЫ МЕТОДОЛОГИЙ РАЗРАБОТКИ ПО

- Каскадная (*она же Waterfall*)
- V-модель
- Инкрементная модель
- Итерационная модель
- Спиральная модель
- RAD-модель
- Семейство гибких методологий
  - Scrum
  - Kanban
  - Lean
  - ...

# КАСКАДНАЯ МОДЕЛЬ

# КАСКАДНАЯ МОДЕЛЬ

- Последовательное прохождение стадий
- Стадия не начинается, пока не закончена предыдущая
- **Четкие:**
  - Сроки
  - Стоимость
  - Результат
- Но есть и **минусы**



# V-МОДЕЛЬ

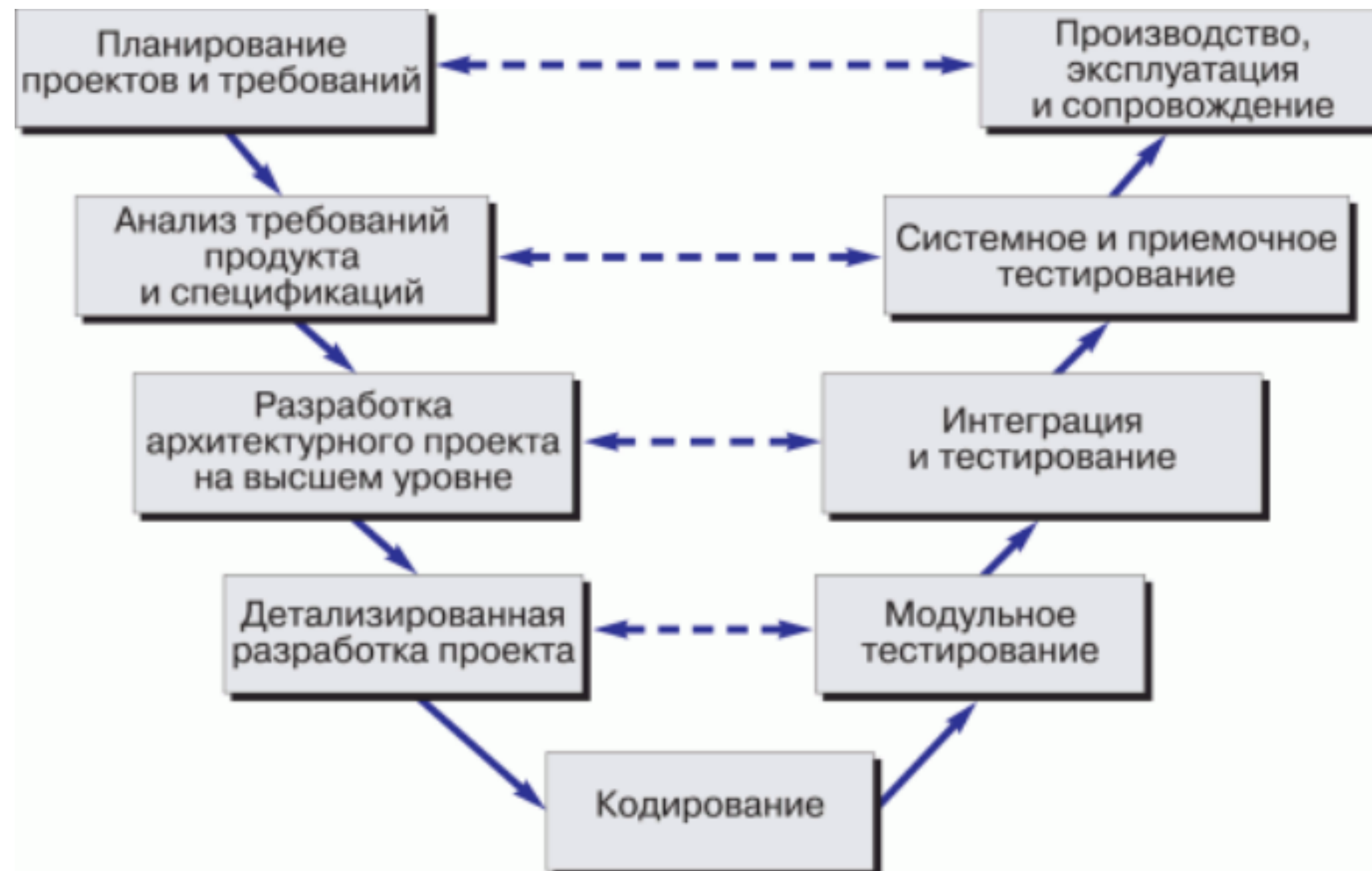
# V-МОДЕЛЬ

## ■ Верификация

- Общая концепция
- Требования бизнеса
- Функциональные требования
- Архитектура
- Реализация

## ■ Валидация

- Приемо-сдаточное тестирование
- Функциональное тестирование
- Интеграционное тестирование
- Модульное тестирование



# ИНКРЕМЕНТНАЯ МОДЕЛЬ

# ИНКРЕМЕНТНАЯ МОДЕЛЬ

- Первая версия - базовая
- Далее - дополнительные возможности
- На каждом этапе:
  - Определение требований
  - Проектирование
  - Реализация
  - Внедрение
  - Тестирование



# ИТЕРАЦИОННАЯ МОДЕЛЬ



# ИТЕРАЦИОННАЯ МОДЕЛЬ

- Каждый этап - база для следующего (определение требований)
- Важный момент - каждая версия полностью работоспособна
- Проводится анализ (опционально - сбор обратной связи) по каждой выпущенной версии для формирования требований и планов для следующей версии

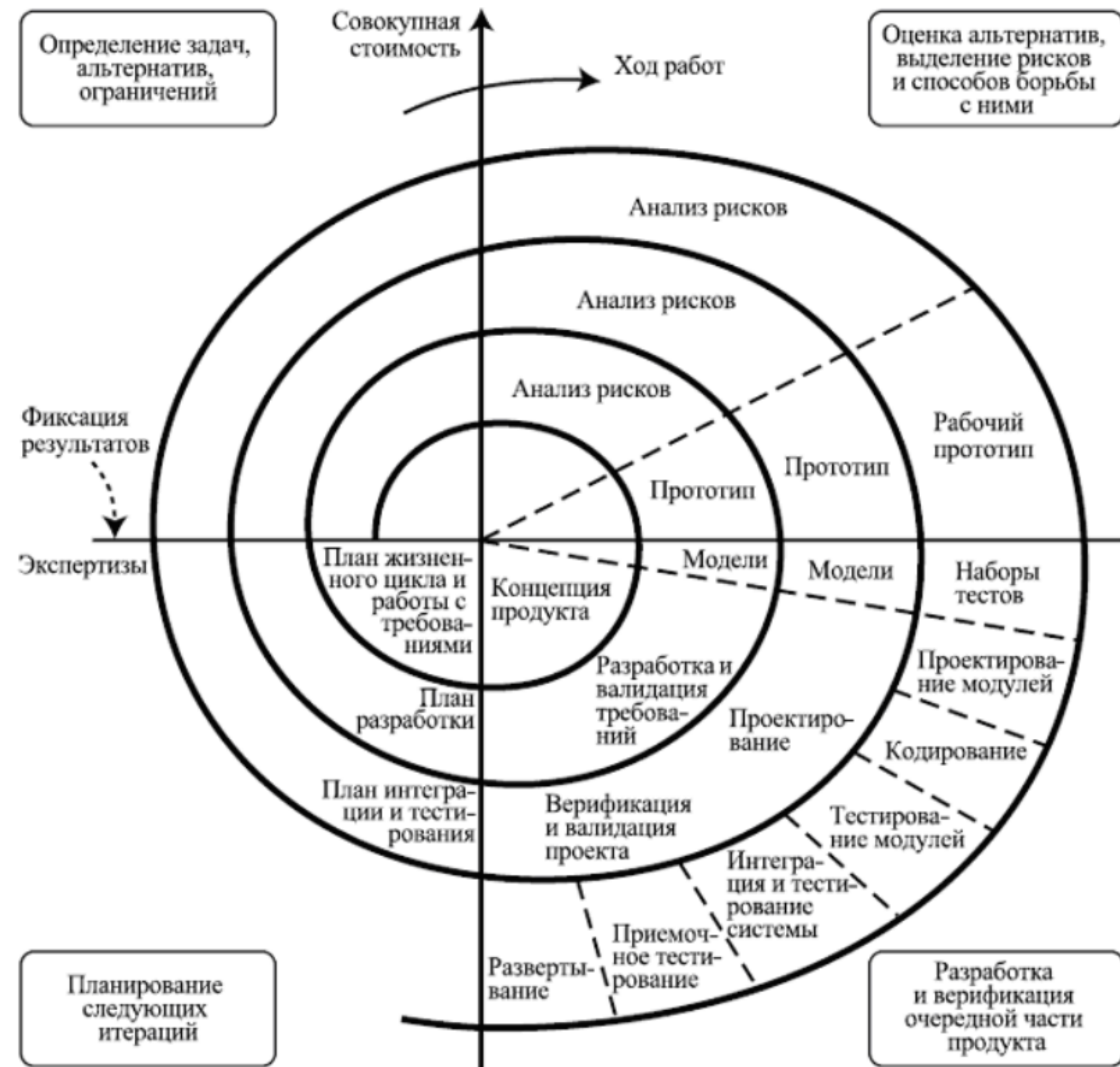


# СПИРАЛЬНАЯ МОДЕЛЬ

# СПИРАЛЬНАЯ МОДЕЛЬ

## Этапы:

- Планирование
- Анализ рисков
- Конструирование
- Оценка результатов



**RAD-МОДЕЛЬ**

# RAD-МОДЕЛЬ

## **Rapid Application Development Model**

- Различные модули разрабатываются различными командами
- Жестко ограниченное время
- Интеграция отдельных модулей в один проект
- Использование инструментов автоматической сборки и генерации кода

### **Этапы:**

- Бизнес-моделирование
- Анализ и создание модели данных
- Анализ и создание моделей процессов
- Автоматическая сборка приложения и тестирование

**СЕМЕЙСТВО ГИБКИХ МЕТОДОЛОГИЙ**

# AGILE

- Семейство гибких методологий разработки
- Короткие итерации
- Разные метрики качества работы
- Много разных конкретных подходов

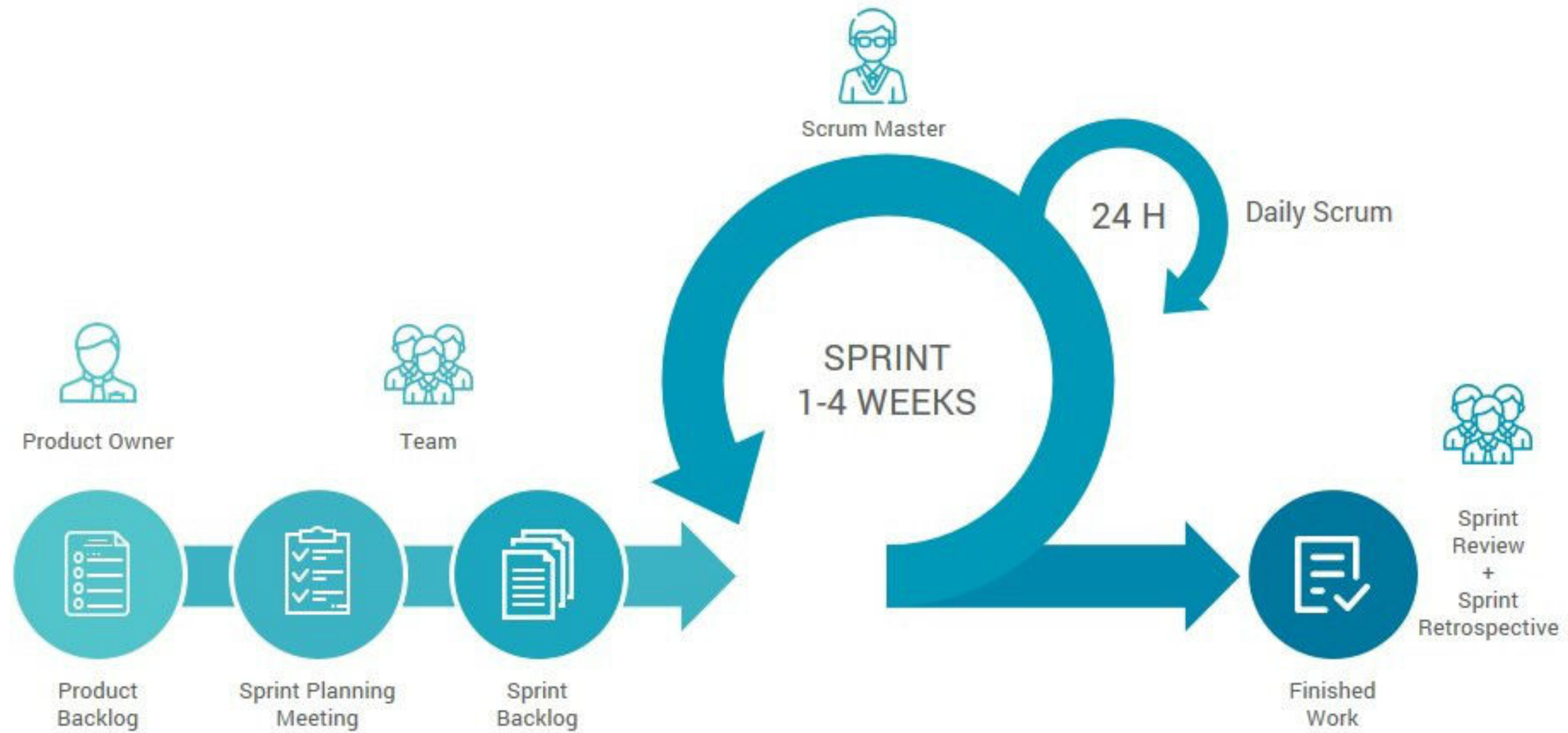
# AGILE-МАНИФЕСТ

- ЛЮДИ и ВЗАИМОДЕЙСТВИЕ важнее процессов и инструментов
- РАБОТАЮЩИЙ ПРОДУКТ важнее документации
- СОТРУДНИЧЕСТВО С ЗАКАЗЧИКОМ важнее условий контракта
- ГОТОВНОСТЬ К ИЗМЕНЕНИЯМ важнее первоначального плана



**SCRUM**

# SCRUM



**KANBAN**

# KANBAN

- Контролируем каждую задачу
- Общая идея - минимизация времени реализации каждой задачи
- Отлично работает, когда команда гетерогенная

	В работе 6/6	Анализ	Дизайн	Разработка 3/3	Тестирование
Команда 1	<div><div></div><div></div></div> <div><div></div><div></div></div> <div><div></div><div></div></div>	<div><div></div><div></div></div> <div><div></div><div></div></div>	<div><div></div><div></div></div> <div><div></div><div></div></div> <div><div></div><div></div></div>	<div><div></div><div></div></div>	<div><div></div><div></div></div> <div><div></div><div></div></div>
Команда 2	<div><div></div><div></div></div> <div><div></div><div></div></div> <div><div></div><div></div></div>			<div><div></div><div></div></div> <div><div></div><div></div></div>	<div><div></div><div></div></div> <div><div></div><div></div></div>

>>: tbc...