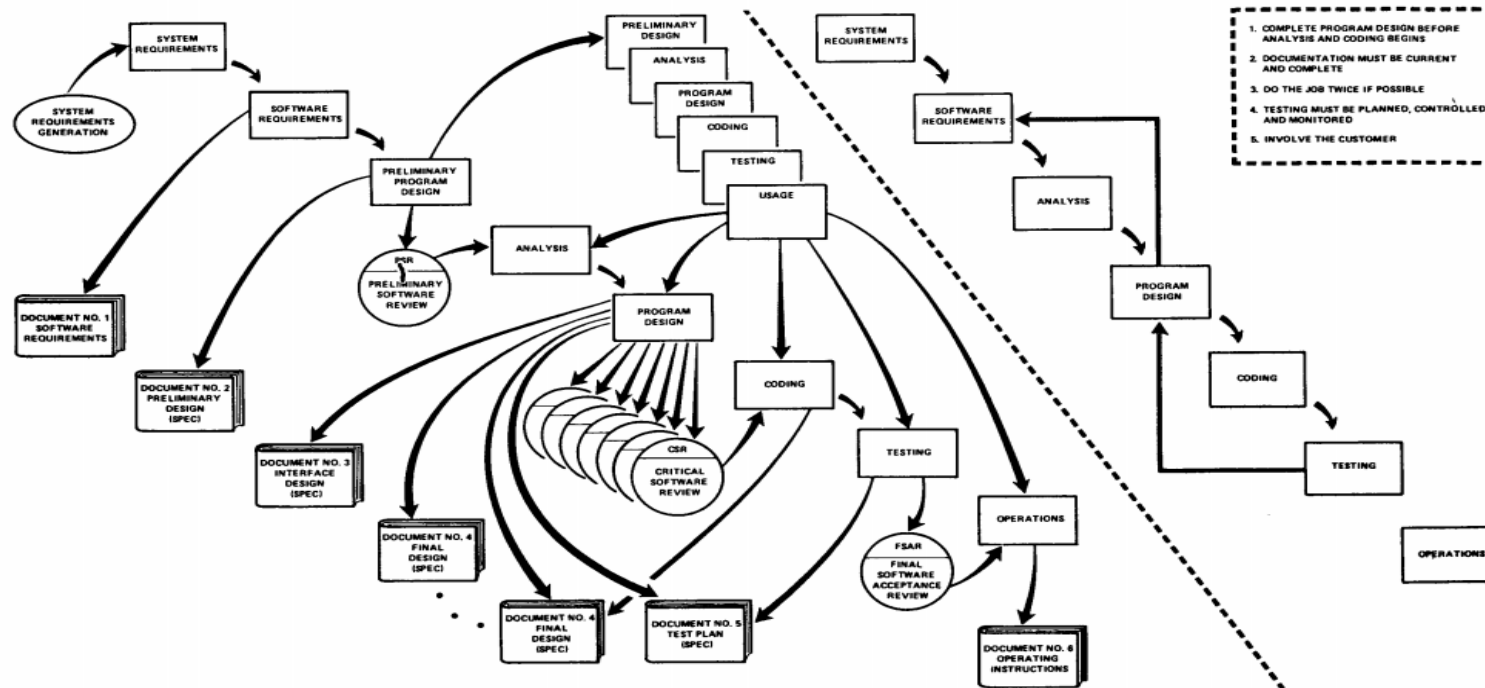


## **Лекция 2**

# **AGILE - МЕТОДОЛОГИИ**

# Водопадная модель: высокая стоимость исправления дефектов



Dr. Winston W. Rovce Managing the development of large software systems

# СТОИМОСТЬ ИСПРАВЛЕНИЯ ДЕФЕКТОВ (ТЫС.ДОЛЛ.)

	ЭТАП ОБНАРУЖЕНИЯ ДЕФЕКТА				
Этап внесения дефекта	Выработка требований	Проектирование архитектуры	Тестирование	Конструирование системы	После выпуска ПО
Выработка требований	1	3	5-10	10	10-100
Проектирование архитектуры	—	1	10	15	25-100
Конструирование -	—	—	1	10	10-25

# НАЗНАЧЕНИЕ AGILE

- ✓ Ускорение вывода продукта на рынок.
- ✓ Управление изменениями (изменение требований в процессе разработки).
- ✓ Улучшение взаимодействия между IT и бизнесом (точная реализация бизнес-требований).

# Базовые ценности AGILE

- Люди и их взаимодействие важнее процессов и инструментов;
- Готовый продукт важнее документации по нему;
- Сотрудничество с заказчиком важнее жестких контрактных ограничений;
- Реакция на изменения важнее следования плану.

# СРАВНЕНИЕ AGILE/WATERFALL



# Принципы Agile

1. Наш высший приоритет – это удовлетворение заказчика с помощью частых и непрерывных поставок продукта, ценного для него.
2. Мы принимаем изменения в требования, даже на поздних этапах реализации проекта.
3. Гибкие процессы приветствуют изменения, что является конкурентным преимуществом для заказчика.
4. Поставлять полностью рабочее программное обеспечение каждые несколько недель, в крайнем случае, каждые несколько месяцев. Чем чаще, тем лучше.
5. Представители бизнеса и команда разработки должны работать вместе над проектом.
6. Успешные проекты строятся мотивированными людьми. Дайте им подходящую окружающую среду, снабдите всем необходимым и доверьте сделать свою работу.
7. Самый эффективный метод взаимодействия и обмена информацией – это личная беседа.

..

8. Рабочее программное обеспечение – главная мера прогресса проекта
9. Гибкие процессы способствуют непрерывному развитию. Все участники проекта должны уметь выдерживать такой постоянный темп.
10. Постоянное внимание к техническому совершенству и качественной архитектуре способствуют гибкости.
11. Простота необходима, как искусство максимизации работы, которую не следует делать.
12. Лучшая архитектура, требования, дизайн создается в самоорганизующихся командах.
13. Команда постоянно ищет способы стать более эффективной, путем настройки и адаптации своих процессов.

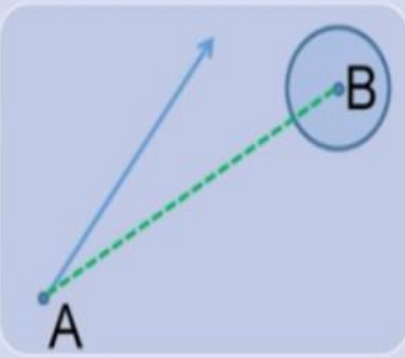
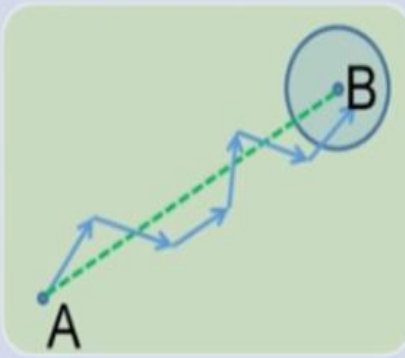
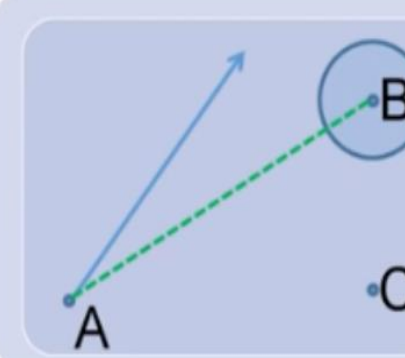
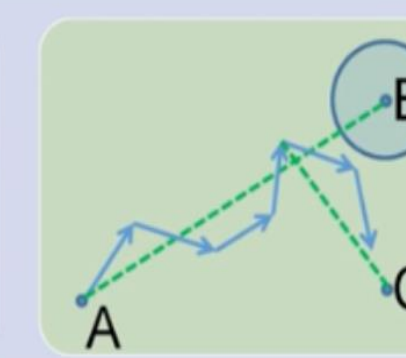


# AGILE

*AGILE* — подходы к созданию продуктов, путем непрерывной быстрой поставки ценного рабочего функционала самоорганизованной командой профессионалов в сотрудничестве с заказчиком.

# AGILE/WATERFALL

## достижение целей/уточнение требований

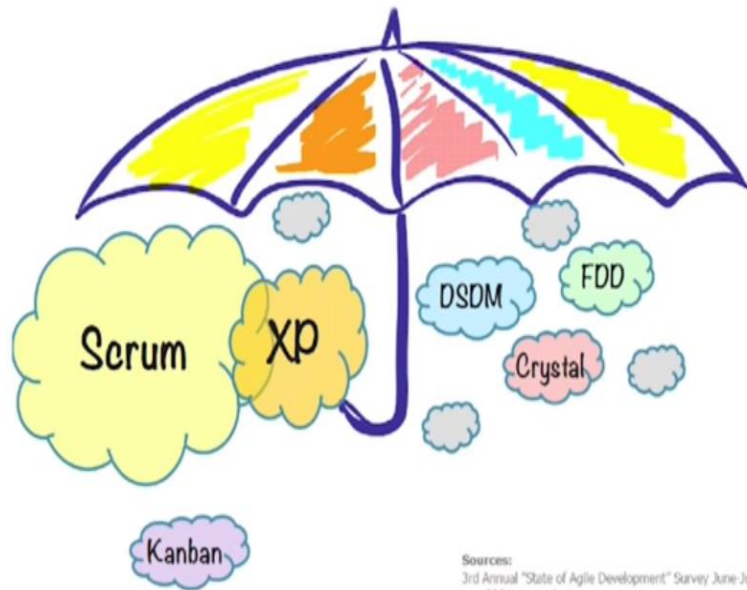
Итеративность	Изменение требований
<div data-bbox="136 483 1093 1216"><div data-bbox="174 523 577 882"><p data-bbox="219 954 533 1018">Водопадные</p></div><div data-bbox="622 523 1025 882"><p data-bbox="667 954 981 1018">Итеративные</p></div></div>	<div data-bbox="1093 483 2103 1216"><div data-bbox="1131 523 1534 882"><p data-bbox="1198 954 1512 1018">Водопадные</p></div><div data-bbox="1579 523 1982 882"><p data-bbox="1646 954 1960 1018">Итеративные</p></div></div>

# AGILE/WATERFALL

Водопад	Agile
<b>Зафиксировано:</b> (ограничения) содержание	<b>Зафиксировано:</b> (ограничения) время, ресурсы
<b>Планируется:</b> время, ресурсы	<b>Планируется:</b> содержание

# AGILE-МЕТОДОЛОГИИ

Agile-зонтик

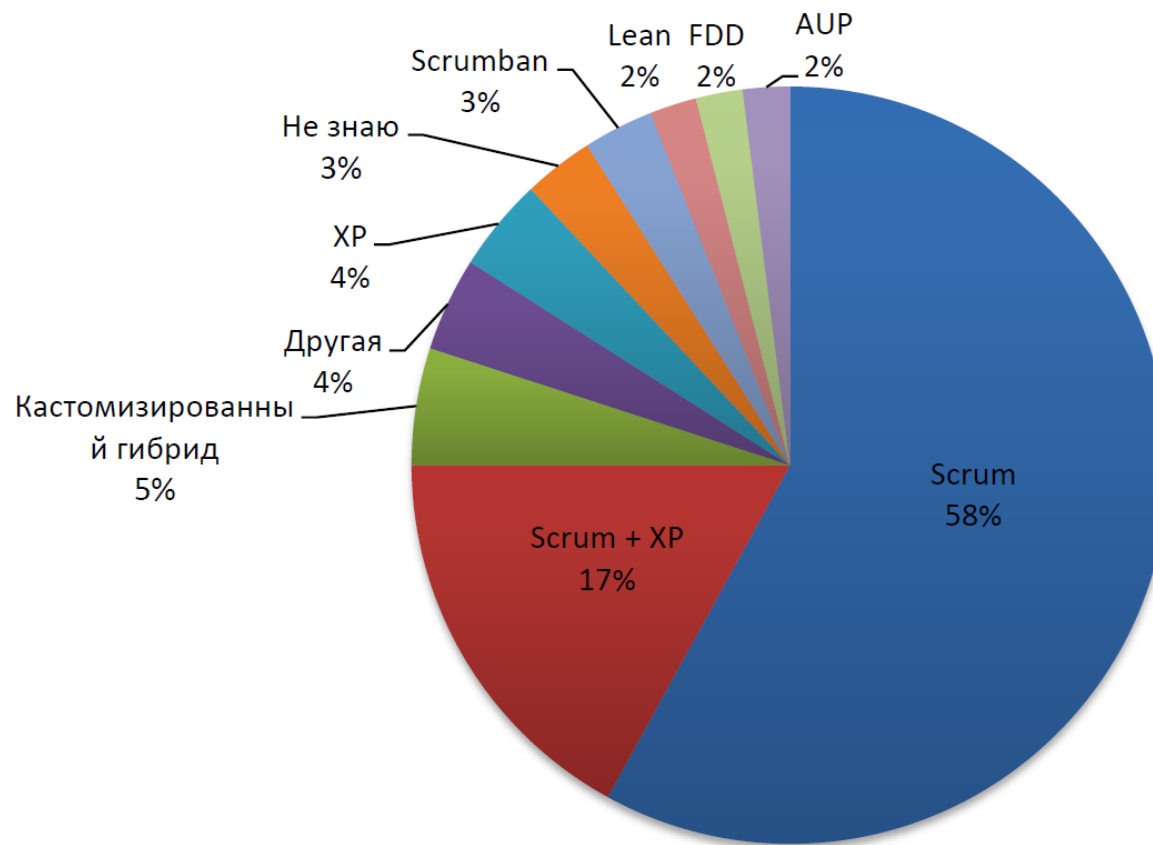


Sources:  
3rd Annual "State of Agile Development" Survey June-July 2  
• 3061 respondents

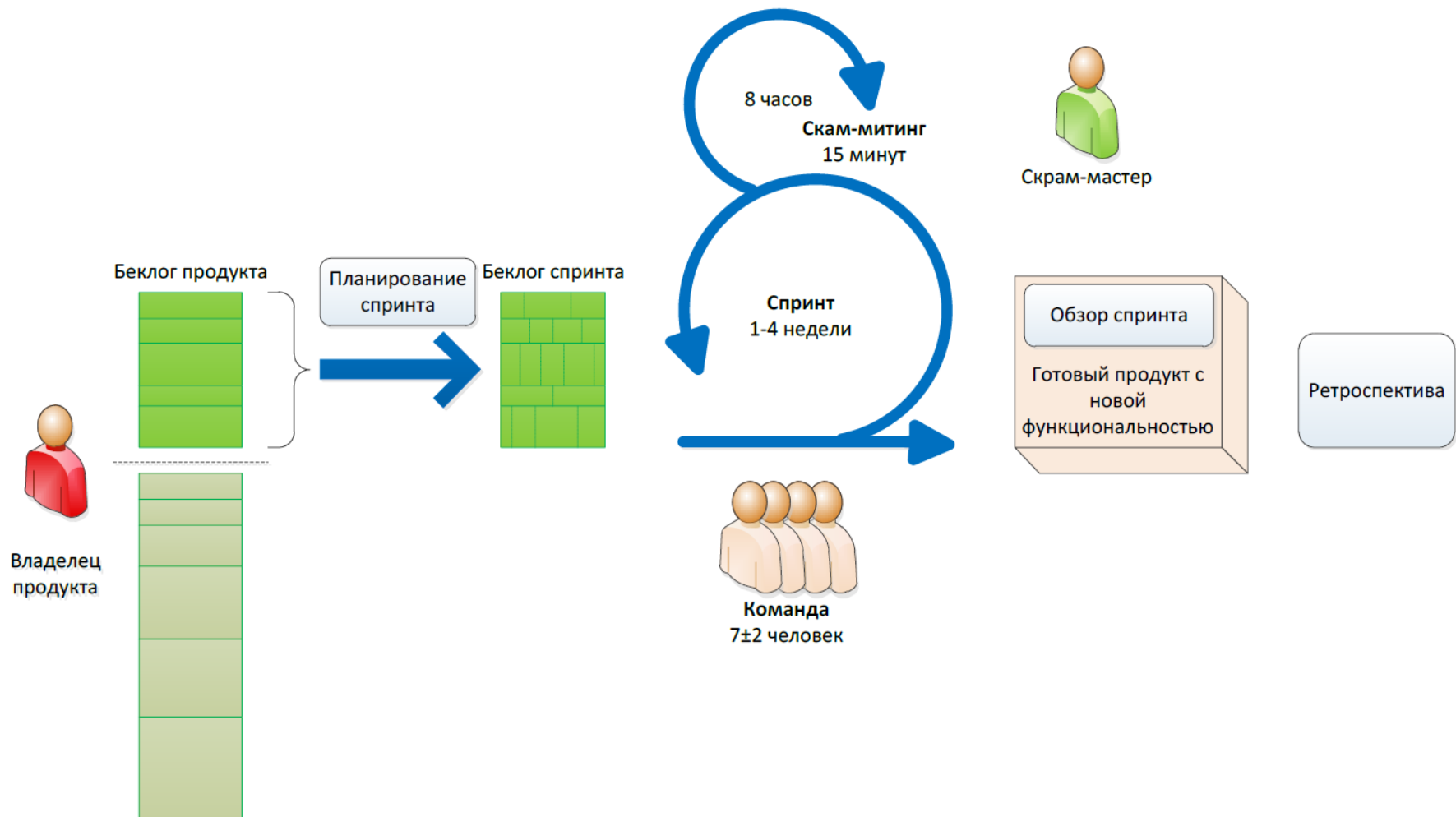
- **XP** —экстремальное программирование;
- **Crystal Clear**;
- **DSDM** - Dynamic Systems Development Method;
- **Kanban**;
- **Iconix**;
- **FDD** - Feature-driven development.

# SCRUM

## наиболее популярная методология



# Общая схема работы Scrum



# Особенности Scrum

- ✓ небольшие кроссфункциональные команды
- ✓ координирует работу команды скрам-мастер
- ✓ разработка состоит из фиксированных итераций от 1 до 4 недель - спринтов
- ✓ владелец продукта отвечает за требования и за приоритеты
- ✓ задачи на спринт набираются согласно скорости работы команды
- ✓ отбор задач — по приоритетам

# ЭЛЕМЕНТЫ SCRUM

## Роли

- Владелец продукта
- Скрам-мастер
- Команда

## Артефакты

- Беклог продукта
- Беклог спринта
- Инкремент продукта

## Процессы

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



# РОЛИ

## 1. Скрам- команда

- Владелец продукта + команда разработки, которую координирует скрам - мастер

## 2. Владелец Продукта

- **Product Owner** - ответственный за максимизацию ценности продукта

# РОЛИ

3.

Команда  
Разработки

- **Development Team** – группа специалистов, которая создает инкремент продукта

4.

Скрам  
мастер

- **Scrum Master** отвечает за процессы разработки: координацию работы и поддержание социальной атмосферы в команде.

# ВЛАДЕЛЕЦ ПРОДУКТА

**Главной целью** деятельности владельца продукта является максимизация ценности продукта.

## Основные задачи:

- ✓ создание и обработка перечня историй пользователя;
- ✓ установка приоритетов для историй пользователя;
- ✓ выработка понимания историй пользователя у команды.

# ВЛАДЕЛЕЦ ПРОДУКТА

## *Основные задачи:*

- Часть этих задач владелец продукта может делегировать членам команды, но он остается ответственным за них.
- Владелец продукта – это всегда один человек, а не группы или комитет, и все требования в виде элементов журнала пожеланий поступают команде через него.

# КОМАНДА РАЗРАБОТКИ

- ✓ Команда разработки в конце каждого спринта предоставляет потенциально готовый к релизу продукт.
- ✓ Ответственность за поставку лежит на команде в целом.
- ✓ Самоорганизация команды - она сама определяет способ, которым сделает из элементов журнала историй инкремент продукта.
- ✓ Размер команды разработки 3-9 человек (не считая владельца продукта и скрам-мастера).

# ОБЯЗАННОСТИ СКРАМ-МАСТЕРА

## Владелец продукта

- Помогает в планировании и оценке элементов журнала пожеланий.
- По необходимости помогает в организации процессов

## Команда разработки

- Обучает команду самоорганизации и многофункциональности.
- Устраняет внешние препятствия.
- По необходимости помогает в организации процессов

## Организация в целом

- Адаптирует Scrum в соответствии с потребностями организации.
- Помогает понять внешним к команде людям Scrum.
- Обменивается лучшими практиками с другими скрам-мастерами

# АРТЕФАКТЫ

## Бэклог Продукта

- **Product Backlog** - упорядоченный список требований к разрабатываемому продукту.
- Обычно он состоит из бизнес-требований, которые приносят конкретную бизнес-пользу и называются историями пользователя.

## Бэклог Спринта

- **Sprint Backlog** - часть журнала историй пользователя с самой высокой важностью и суммарной оценкой, не превышающей скорости команды, отобранная для спринта.

## Инкремент

- новая функциональность продукта, созданная во время спринта.

# ПРОЦЕССЫ

## Спринт

- **Спринт** – итерация с фиксированным сроком, длительностью 1- 4 недели, в течение которого создается инкремент продукта.

## Планирование спринта

- Планирование работы на предстоящий спринт.

## Ежедневный Скрам

- **(Daily Scrum)** 15 - минутные мероприятия для команды.



# ПРОЦЕССЫ

## Обзор Спринта

- Оценка результатов, проводится в конце спринта

## Ретроспек- тива Спринта

- Оценка процесса разработки, проводится после нескольких спринтов

# СКРАМ-МИТИНГ

Цель - синхронизации действий и создания плана работы на ближайшие 24 часа.

Каждый член команды отвечает на три вопроса:

- ✓ Что было сделано с предыдущего скрам-митинга?
- ✓ Какие возникли проблемы?
- ✓ Что будет сделано к следующему скрам-митингу?

# ПЛАНИРОВАНИЕ СПРИНТА

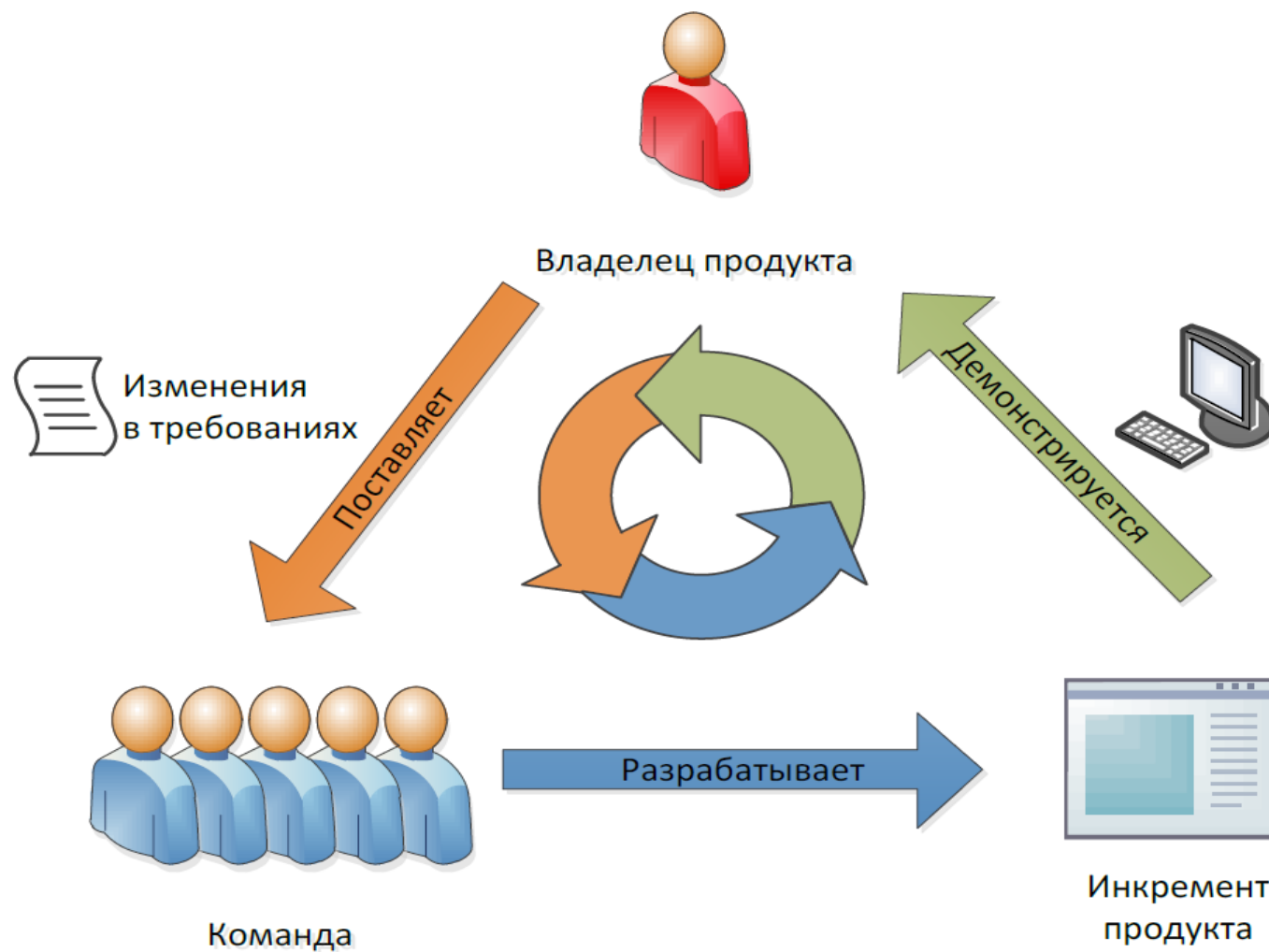
- ✓ Важность каждого элемента бэклога оценивается в числовой форме;
- ✓ Элементы бэклога (истории пользователя) уточняются так, чтобы были понятны всей команде и владельцу продукта;
- ✓ Владелец продукта определяет функциональность (перечень историй пользователя), которые будут реализованы на каждом спринте.

Результат - список задач, которые команда планирует реализовать в рамках спринта.

# ОБЗОР СПРИНТА

- ✓ Демонстрация владельцу продукта и заинтересованным лицам работающего функционала продукта, сделанного за спринт.
- ✓ Обсуждение выполненной во время спринта работы.

# ОБРАТНАЯ СВЯЗЬ В SCRUM



# РЕТРОСПЕКТИВА

- ✓ Длительность - от 30 минут до четырех часов
- ✓ Факторы, влияющие на проведение ретроспективы:
  - **длина спринта**: чем длиннее спринт, тем больше команда успевает сделать и тем больше материала для обсуждения;
  - **размер команды**: чем команда больше, тем больше надо времени, чтобы у каждого ее члена была возможность высказаться и тем больше функционала команда успевает сделать;
  - **наличие проблем**: со временем команда решает проблемы, и ретроспективы сокращаются по времени.

# СТРУКТУРА РЕТРОСПЕКТИВЫ

Открытие – 5%

Сбор данных – 30%-50%

Проникновение в суть – 20%-30%

Принятие решение – 10%

Закрытие – 5%-10%



# РЕТРОСПЕКТИВА: СБОР ДАННЫХ

Заключается в ответах каждого участника на три вопроса:

- ✓ 1. Что было сделано хорошо?
- ✓ 2. Что можно улучшить?
- ✓ 3. Какие улучшения будем делать?



# ТЕМЫ ДЛЯ ОБСУЖДЕНИЯ НА РЕТРОСПЕКТИВЕ:

- ✓ скорость работы команды и причины ее изменения по сравнению с предыдущими спринтами;
- ✓ нереализованные истории пользователей и причины отклонения от графика;
- ✓ дефекты и их причины;
- ✓ качество процесса разработки (нарушения, отклонения).