



Тестирование веб-проектов в Agile

Асхат Уразбаев, ScrumTrek

Содержание

- Специфика веб-проектов
- Коротко об Agile
- Тестирование в Agile
- Виды качества
- Практики тестирования

Специфика веб-проектов

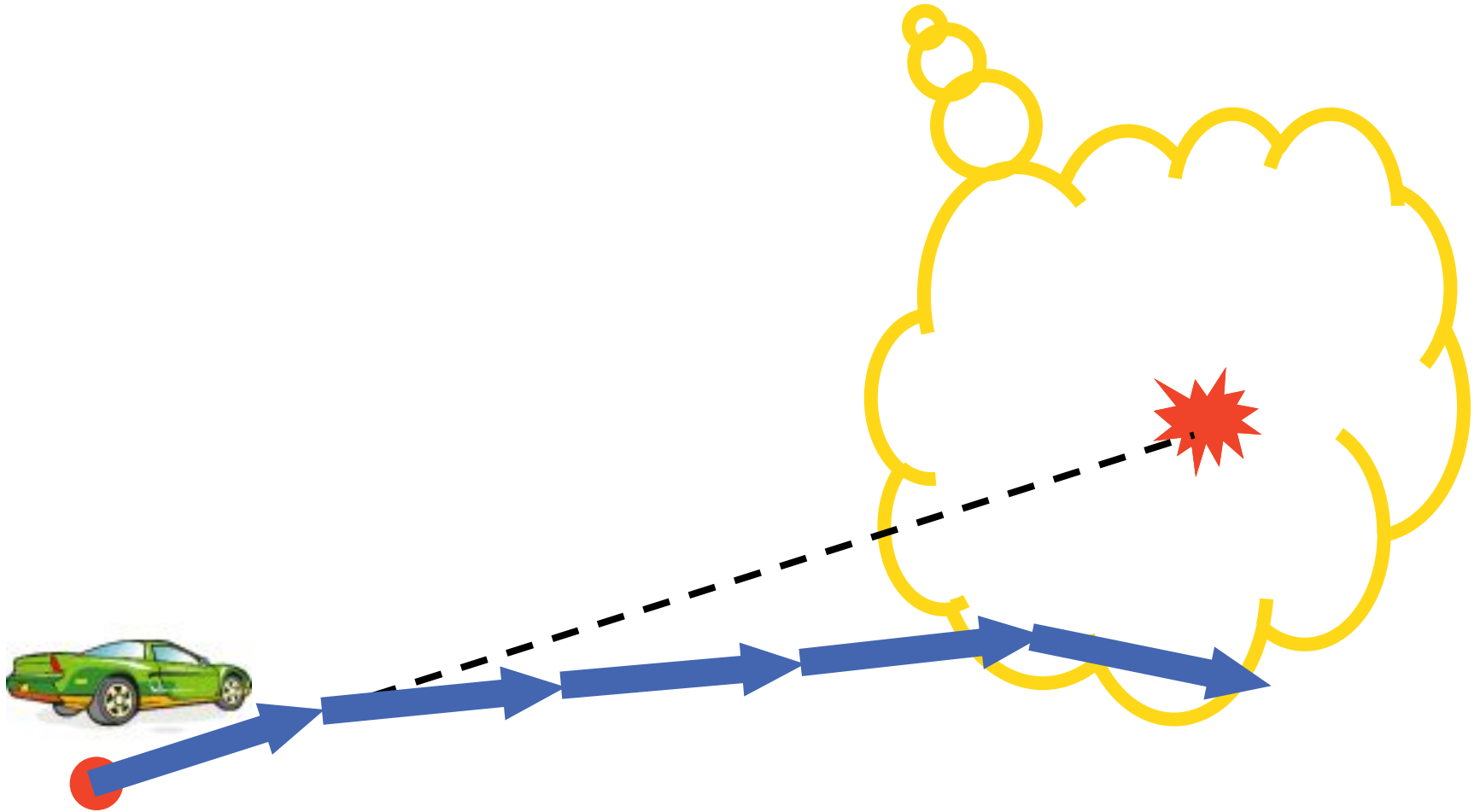
- Короткий цикл выкладки
- Деплой на сервера (без обновления клиентских машин)
- Легкие технологии, относительно простая архитектура
- Относительно невысокая связанность кода

Специфика тестирования

- Низкий уровень документирования
- Тестирование на разных броузерах
- Testing Environment < > Production Environment

Немного об Agile и Scrum

Итеративная разработка



Скр

SCRUM
каждые
24 часа

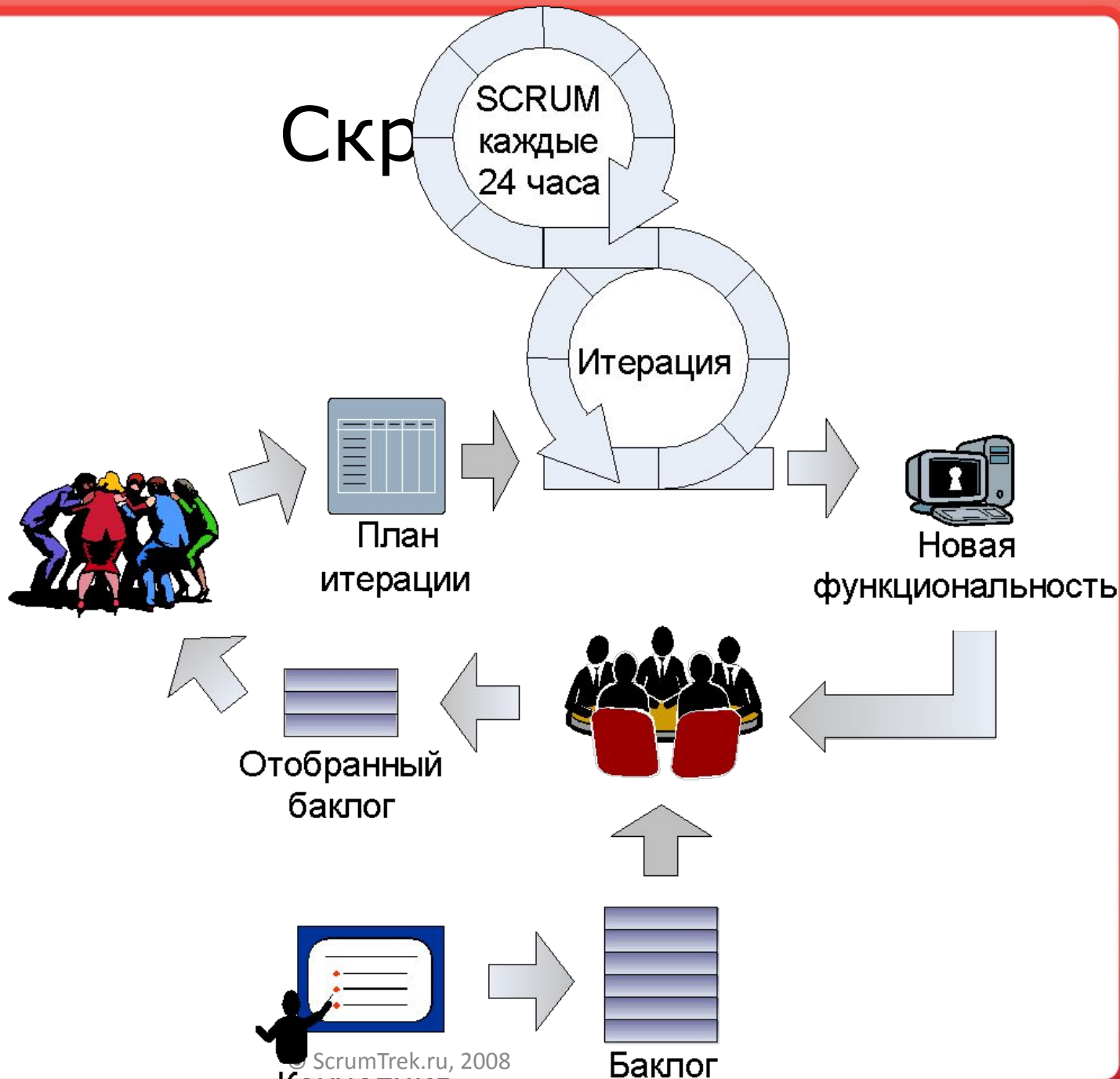
Итерация

План
итерации

Новая
функциональность

Отобранный
баклог

Баклог



Тестирование в Agile

Цикл тестирования

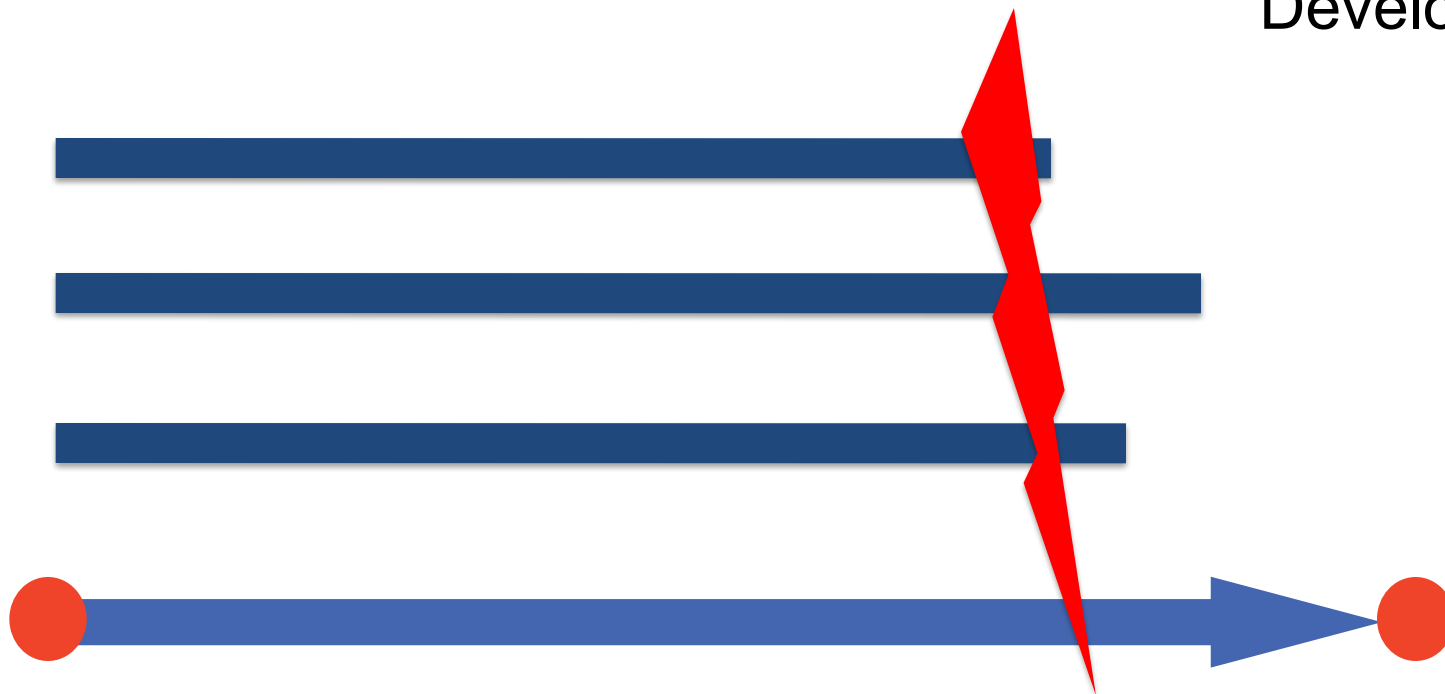
Development



- Каждый разработчик делает свою фичу

Цикл тестирования

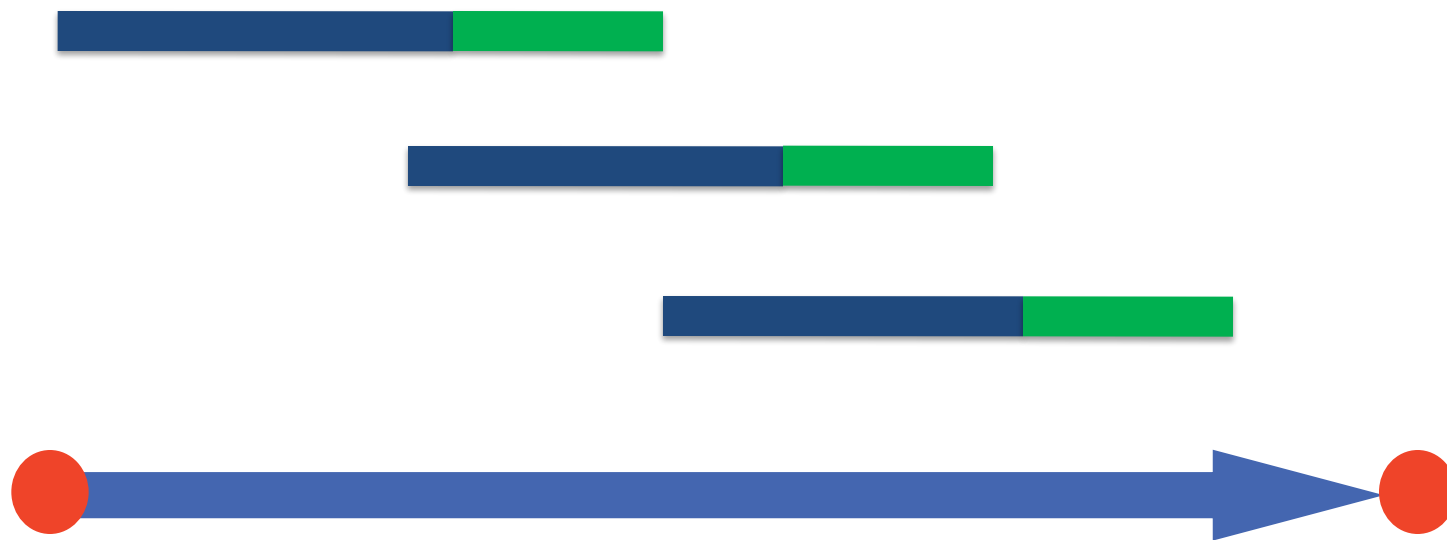
Development



- Если не успевают – не успевают все три!

Цикл тестирования

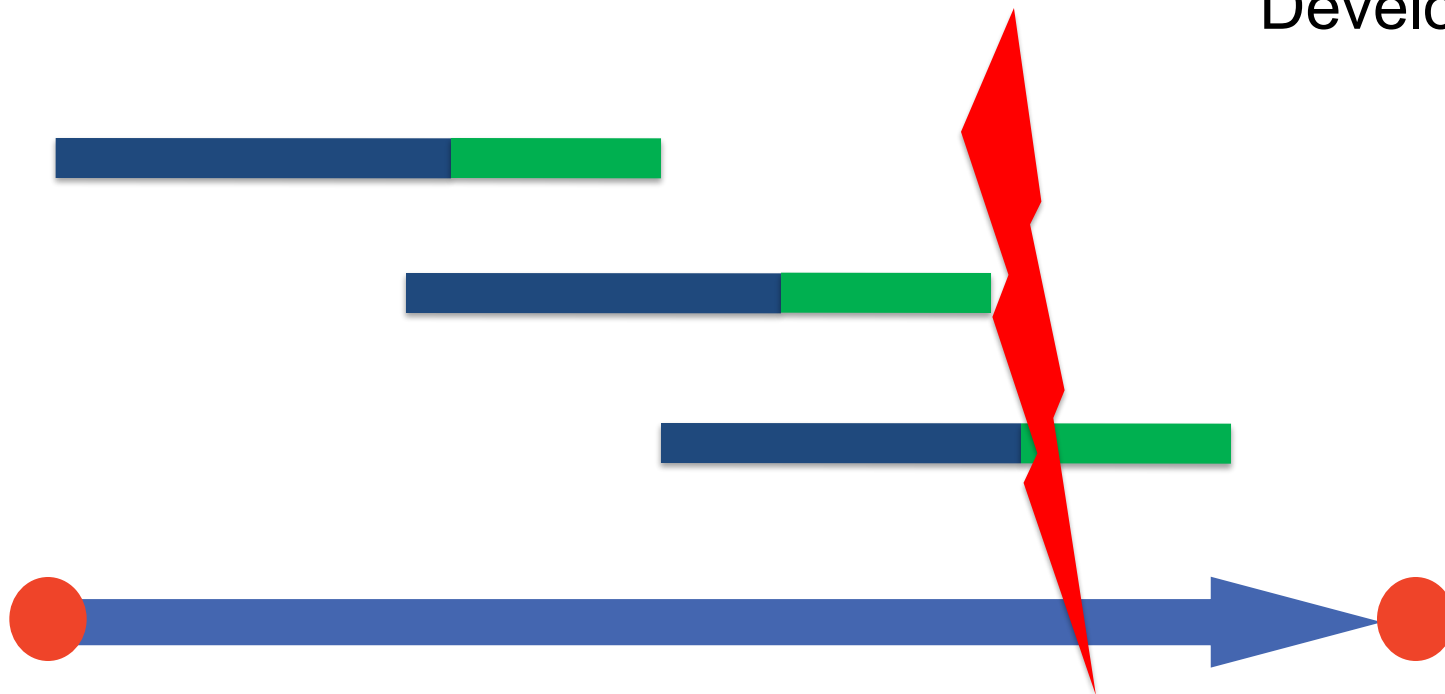
Development



- Фичи делаются в порядке приоритета
- Три разработчика делают одну фичу
- Фичи тестируются сразу

Цикл тестирования

Development



- Не успеваем только низкоприоритетные!



«Все в порядке, сейчас пофиксим!»

Программисты против Тестировщиков

- Программисты слишком полагаются на тестировщиков
- Снижается ответственность за код

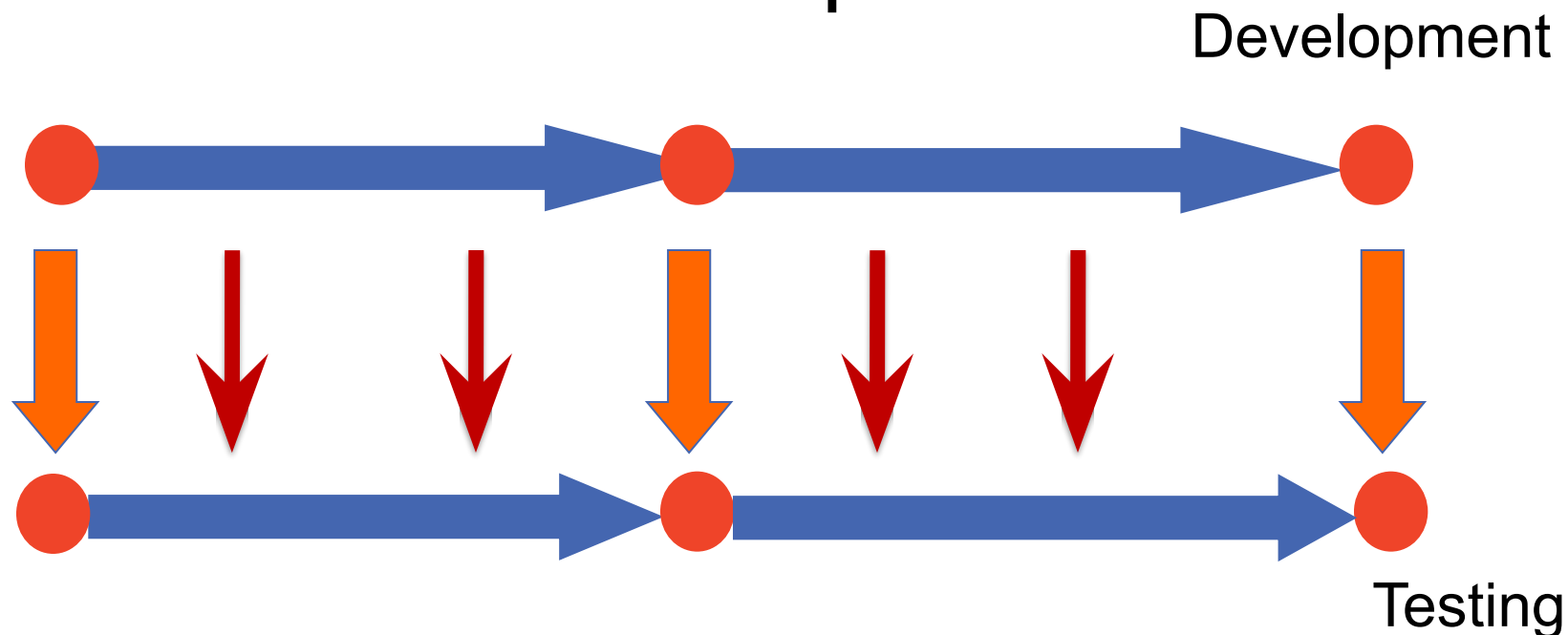
Структура команды

- Команда состоит из разработчиков и тестировщиков
- За качество отвечает вся команда
- Значит и за объем работ отвечает вся команда

Тестировщик

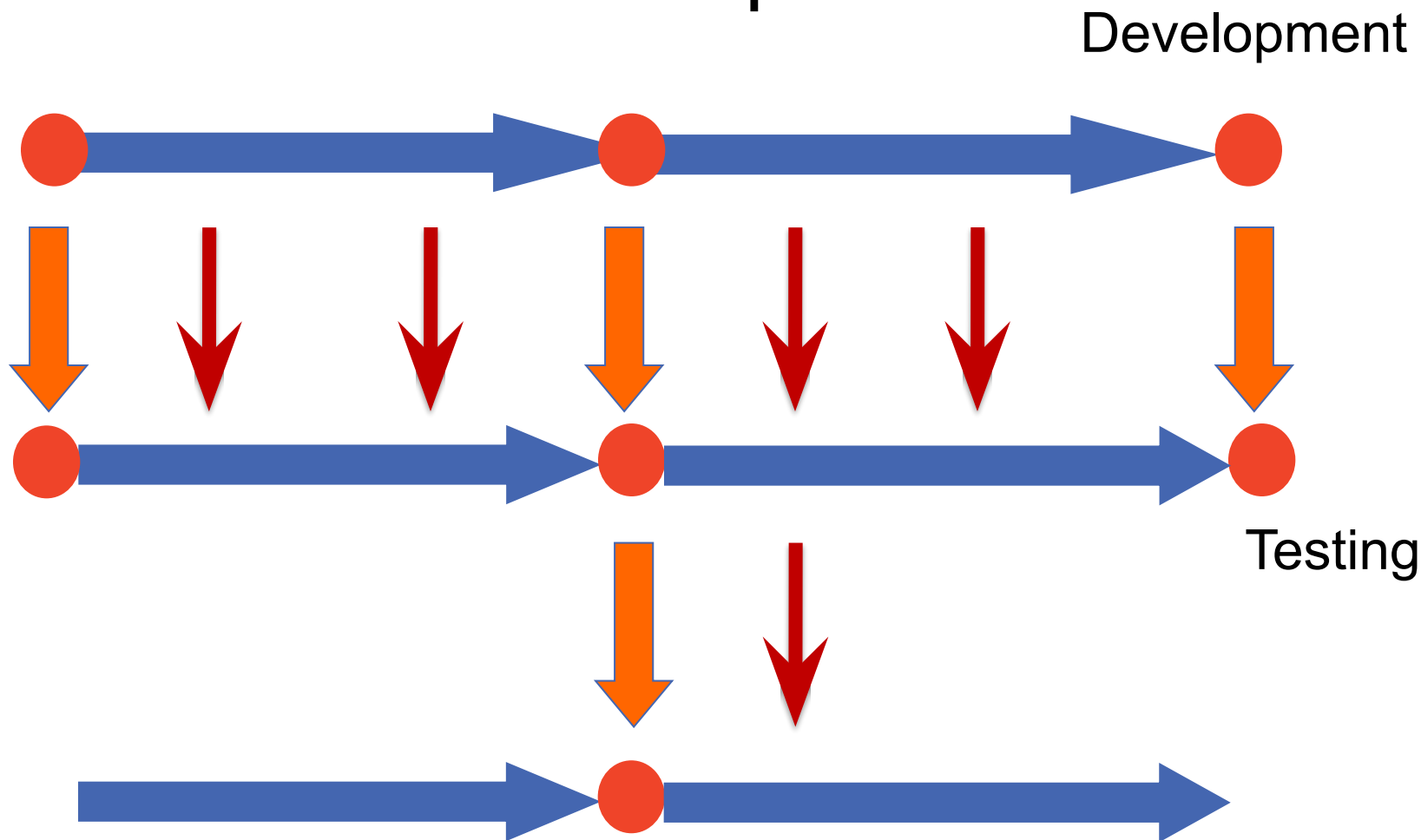
- Тестировщик – член команды
- Участвует в планировании итераций
- Тесно общается с заказчиками
- Помогает документировать требования к коду
- Частично берет на себя функции аналитика

Цикл регрессионного тестирования



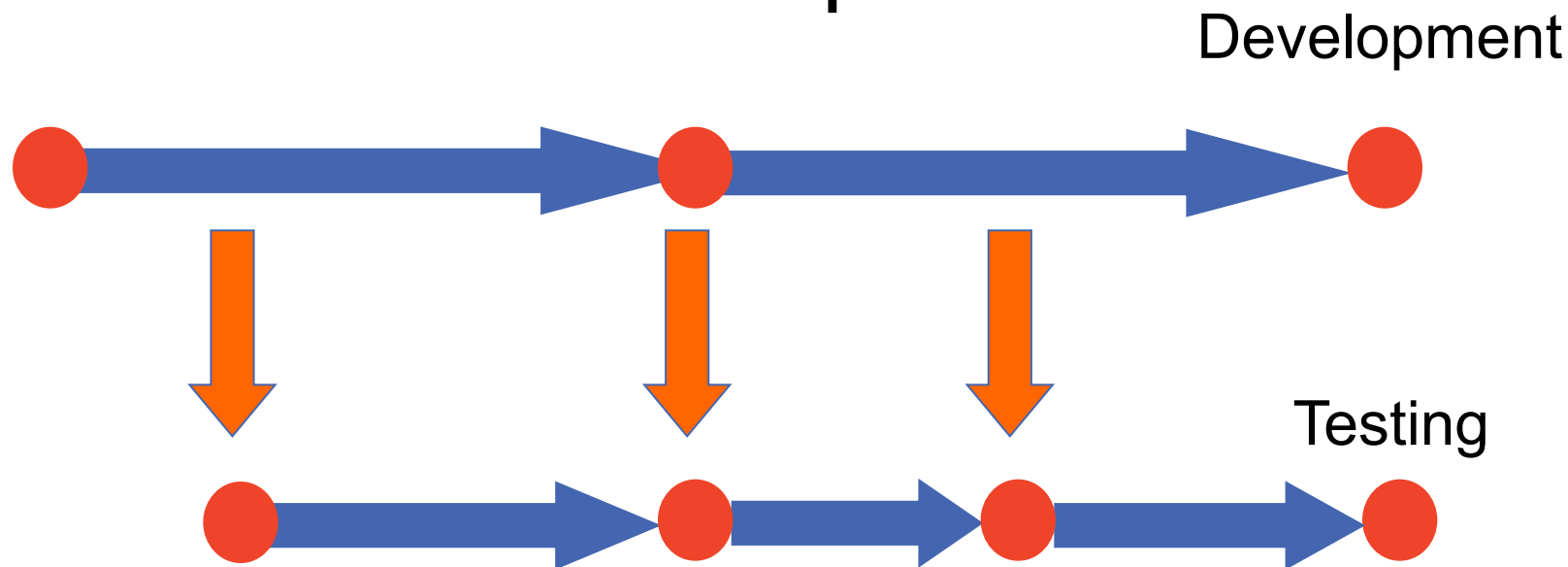
- Пронес на Test Environment в конце итерации
- Создается ветка (branch) «Test»
- Тестирование
- Баги исправляются в основной ветке и делается merge в Test (или наоборот)

Цикл регрессионного тестирования



- А если баги в Production?

Цикл регрессионного тестирования



- Тестирование не прекращается никогда
- Выберите удобный ритм тестирования
- Работать в одной ветке

Что такое качество?

- Качество заказчика
- Внутреннее качество

Качество с точки зрения заказчика

- **Качеством управляет заказчик!**
 - В этой итерации сделайте самую простую версию
 - Будем исходить из 10000 пользователей в день
 - Пока просто текстбокс, AJAX потом добавим

Качество заказчика

- Заказчик осуществляет приемку каждую итерацию
- Постоянная обратная связь от рынка

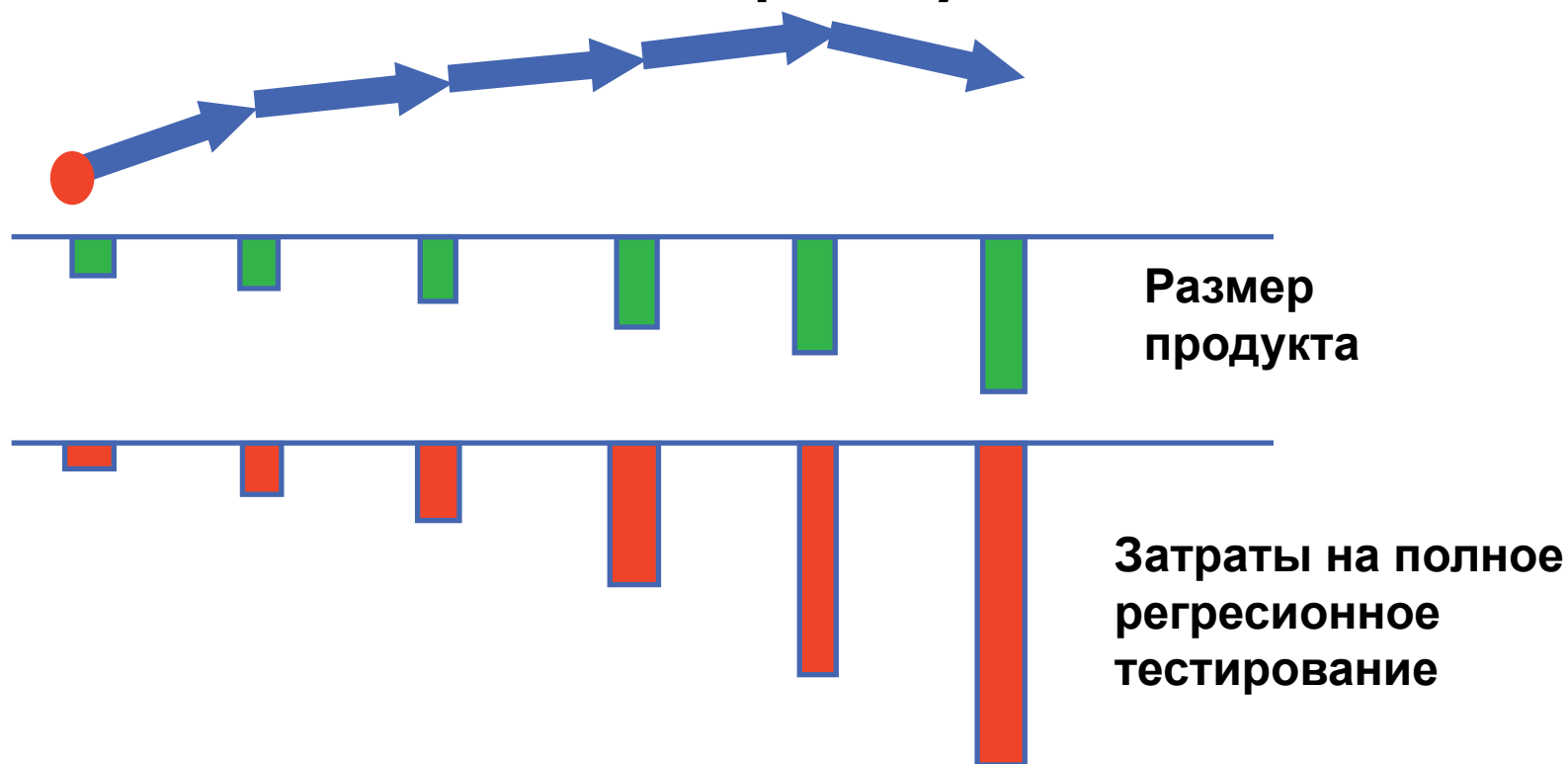
Баги должны приоритезироваться!

- Разработчики НЕ должны исправлять **все** баги, найденные тестерами
- Баги должны быть приоритезированы Product Owner
 - Реальные ошибки
 - Улучшения – фичи
 - Не баги

Внутреннее качество

- Качество кода
 - Простота, читаемость
- Качество архитектуры и дизайна
- Agile:
 - Внутренним качеством управляет команда!
 - Внутреннее качество должно быть **высоким**

Затраты на качество продукта



- Автоматизация тестирования
 - Снизить стоимость изменений
 - Сократить цикл тестирования



Какие препятствия вы можете преодолеть не снижая скорости?

Тестирование

- Continuous Integration
 - Сборка после каждого коммита, прогон Smoke Unit Tests (=Build Acceptance Tests)
- Nightly Builds
 - Ночная сборка для всей системы, прогон всех Unit Tests и интеграционных тестов
- Regression Testing
 - Частое регрессионное ручное тестирование всей системы

Виды автоматизации тестирования

- Unit Tests
 - Дешевые в создании, быстрые при прогоне
 - Не используют базу данных и сеть
- Интеграционные тесты
 - Более дорогие в создании, медленнее при прогоне
 - Проходят весь цикл, включая работу с БД и сетью
- Тесты пользовательского интерфейса
 - Проходят через пользовательский интерфейс
 - Неустойчивые, медленные

Стадии развития проекта

- Без тестеров
- Тестеры внутри команды
- Тестеры в отдельной команде

Независимый отдел тестирования

- Слабая связь разработчиков и тестеров
 - Низкий уровень документирования
- =>
- Низкое качество и производительность тестирования
 - (например, тестеры находят в основном баги верстки)

СПАСИБО!