

# Автоматизация тестирования в жизненном цикле разработки ПО

# Введение

## Цель данного семинара

Обсуждение теории и практики использования автоматизации в тестировании в рамках различных методологий разработки ПО.

## Дисклеймер

В рамках семинара не будет углубления в технологии и готовых решений, лишь обзор и систематизация ряда подходов к тестированию в общем и автоматизации в частности.

# Структура семинара

---

- Роль автоматизации в тестировании
- Классическое понимание автоматизации тестирования (RUP / Waterfall)
- Гибкие подходы: разработка автотестов как инструмент для тестировщика
- План на автоматизацию
- Поддержка тестового фреймворка: культура и соглашения
- Поддержка процесса тестирования с использованием средств автоматизации

# **Роль автоматизации в тестировании**

# Проблемы тестирования

- Рутина
  - Скорость
  - Затраты
  - Управление рисками
  - Человеческий фактор?
- 
- Предоставление отчётности и *своевременной* обратной связи
  - Оценка качества и сбор статистики
  - Частые и внезапные изменения требований
  - Внезапное изменение кода
  - ...

# Цели автоматизации тестирования

- **Оптимизация затрат**
- **Ускорение тестирования**
- **Повышение качества ПО**
- **Повышение производительности команды**
- ***Уменьшение влияния человеческого фактора?***

# Ресурсы для автоматизации

- **Специалист**
- **Время**
- **Инфраструктура**
- **Процесс**

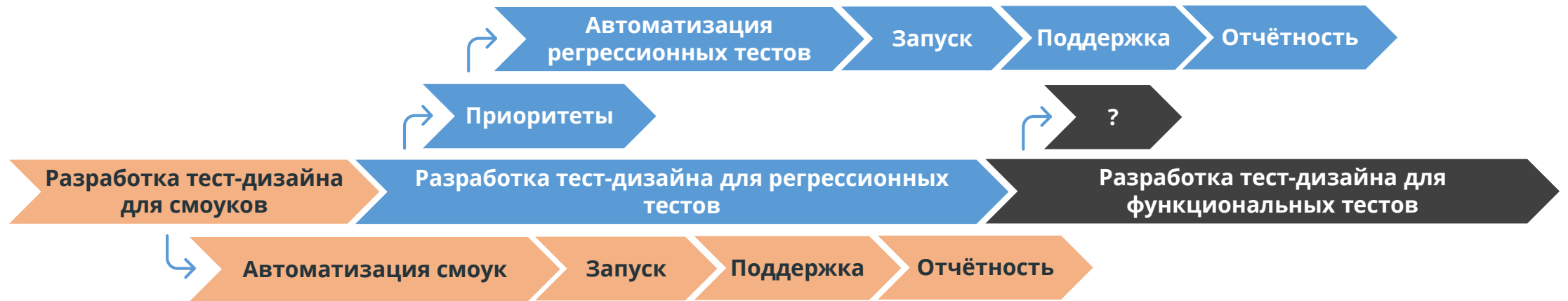
# Критерии применения автоматизации к различным видам и операциям тестирования

- Ясность постановки задачи
- Возможность повторного использования теста
- Возможность выполнения операции вручную
- Оценка трудозатрат на *разработку* автотеста
- Оценка трудозатрат на *поддержку* автотеста



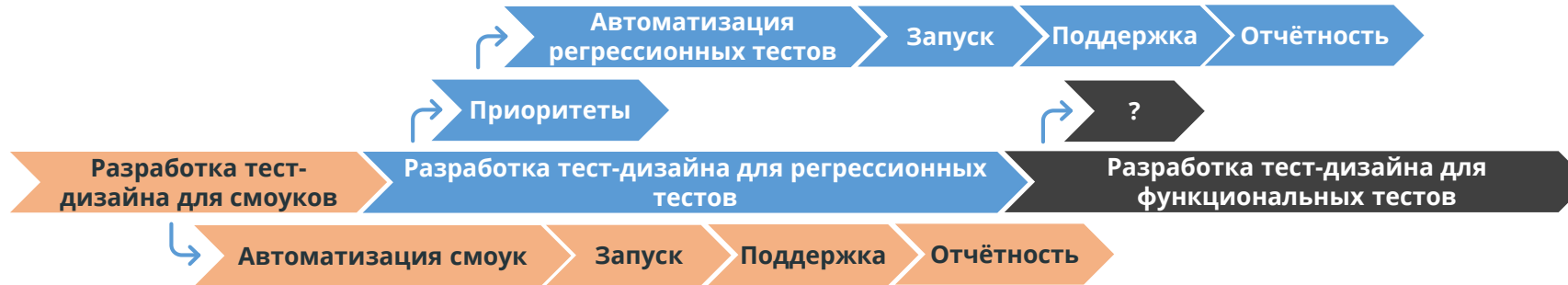
# **Классическое понимание автоматизации тестирования (RUP / Waterfall)**

# Автоматизация готового тест-дизайна



1. Смоук тестирование – первый и главный приоритет
2. Автоматизация регрессионных тестов для того, чтобы снять рутину и нагрузку с команды тестирования
3. Что делать с функциональными тестами?

# Ресурсы и проблемы



**Что нужно для использования автоматизации при таком подходе к тестированию?**

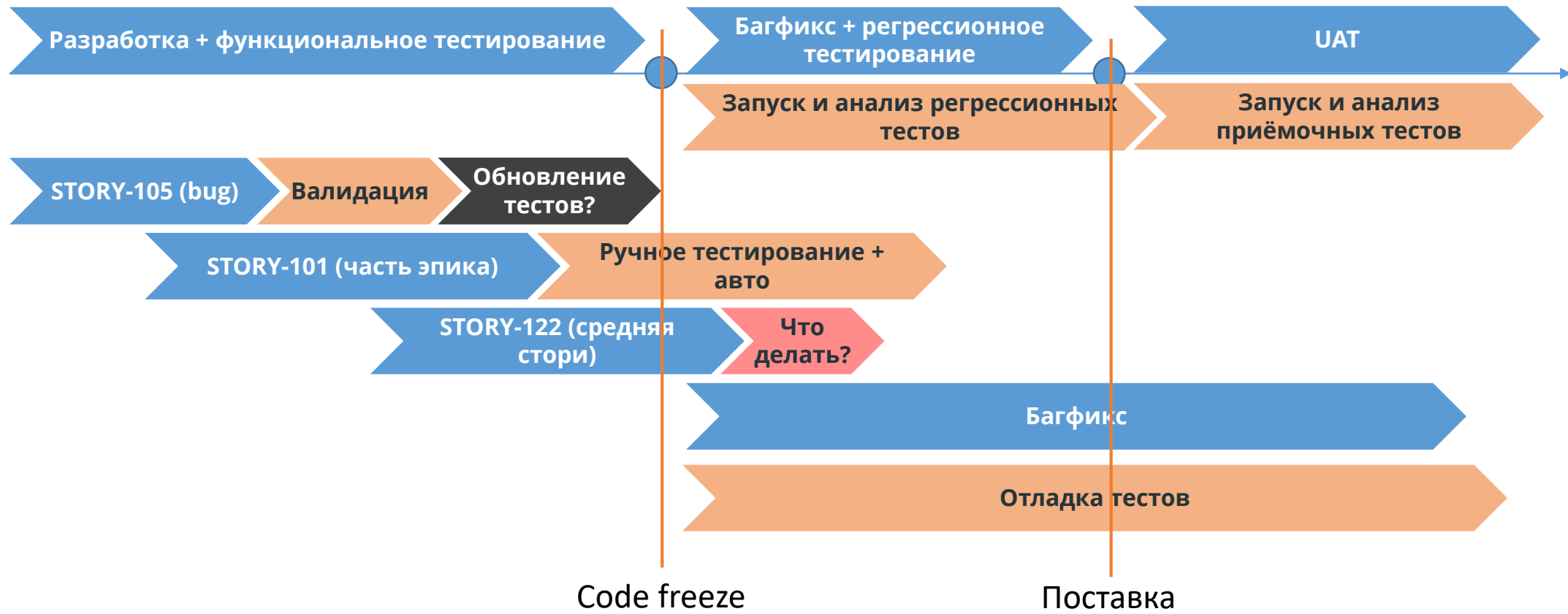
1. Тест-дизайн
2. Рабочие окружения
3. Понимание членами команды достоинств и недостатков автоматизации
4. Специалист

**С какими проблемами можно столкнуться?**

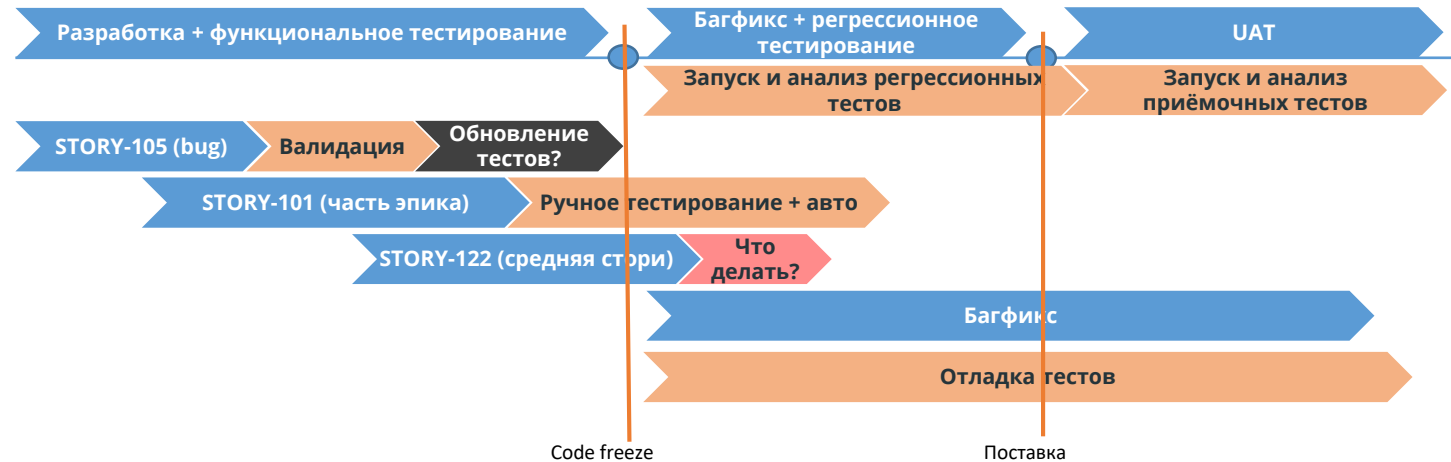
- Неактуальный или неформализованный тест-дизайн
- Расхождение в понимании смысла автотестов
- Простой специалиста

## **Гибкие подходы: разработка автотестов как инструмент для тестировщика**

# Автоматизация как инструмент тестирования в рамках гибких подходов к разработке



# Ресурсы и проблемы



**Что нужно для использования автоматизации при таком подходе к тестированию?**

1. Специалист
2. Рабочие окружения
3. Правильно составленные задачи на текущую итерацию и бэклог

**С какими проблемами можно столкнуться?**

- Приоритеты задач
- Критерии приёмки и начала/окончания тестирования
- Приоритет ручного тестирования над автоматизированным

**Что делать?**

## **План на автоматизацию**

# План на автоматизацию

План на автоматизацию = приоритезированный список задач тестирования, которые она (автоматизация) решает

## Waterfall

1. Оптимизация затрат
2. Ускорение тестирования
3. Повышение качества ПО
4. Повышение производительности команды

## Agile

1. Ускорение тестирования
2. Повышение качества ПО
3. Оптимизация затрат
4. Повышение производительности команды



# Критерии применимости автоматизации

## Цели автоматизации

**Оптимизация затрат**

**Ускорение тестирования**

**Повышение качества ПО**

**Повышение производительности команды**

## Критерии применимости автоматизации

1. Возможность повторного использования теста  
(Как часто мы будем его запускать?)
2. Оценка трудозатрат на разработку автотеста  
(За какое время можно написать тест?)
3. Возможность выполнения операции вручную  
(Можно / не проще ли руками?)
4. Ясность постановки задачи  
(Что нужно сделать? Когда нужно сделать?)
5. Оценка трудозатрат на поддержку автотеста  
(Что может меняться?)

# Принятие решения

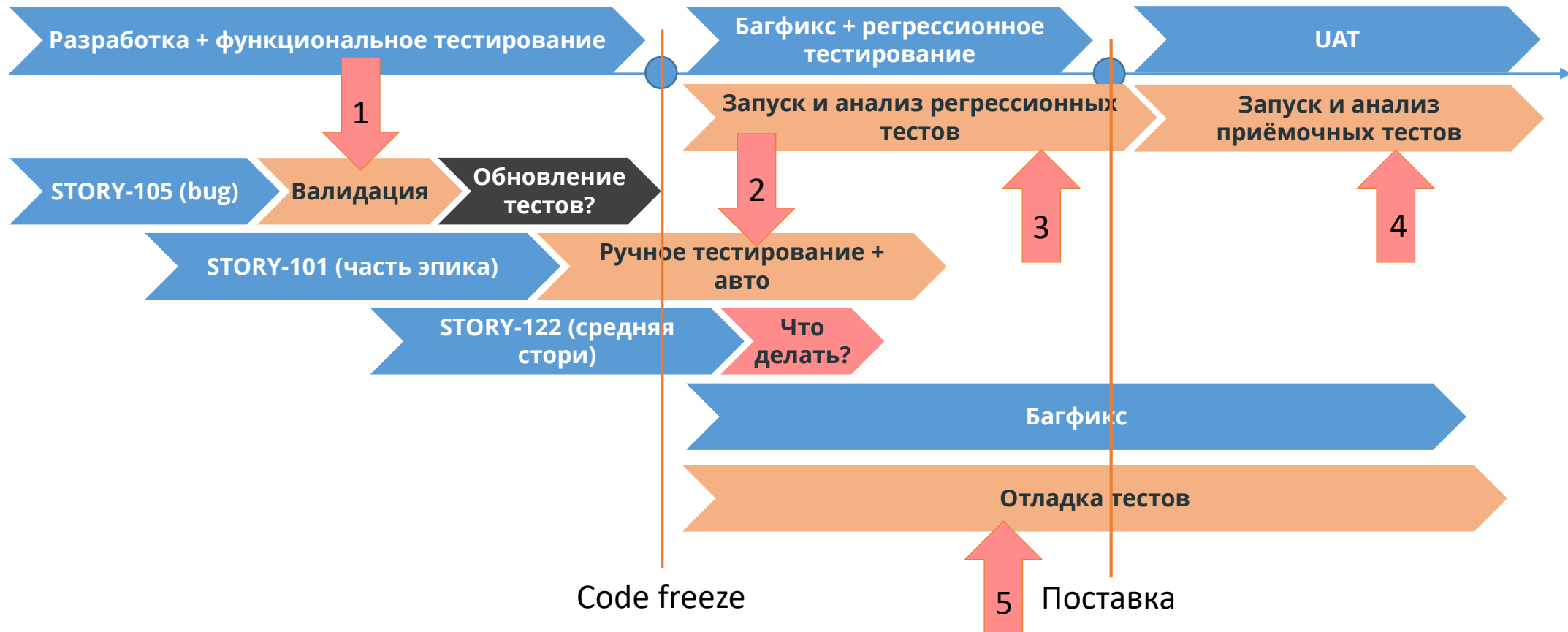
Разработчик: «Слушай, мне бы заавтоматизировать создание сущности с определённым набором параметров, хочу кой-чего посмотреть»

1. Ускорит ли это тестирование задачи вообще? (если да, то п.2)
  1. Сможем ли мы запускать этот тест в рамках регрессии?
  2. Сможет ли этот тест находить дефекты в будущем?
2. Сколько времени займёт написать тест? (если разумно, то п.3)
3. Не проще ли руками? (если не проще, то п.4)
4. Всё ли понятно что нужно сделать (если более-менее, то п.5)
5. Сложно ли будет поддерживать тест в будущем?

# Работа без плана по автоматизации в Waterfall



# Работа без плана по автоматизации в Agile



# Тезис

# Будь строгим

## **Поддержка тестового фреймворка: культура и соглашения**

# Поддержка тестового фреймворка: культура и соглашения

## Почему не работают автотесты?

- Изменение селекторов на UI
- Изменение поведения элементов на странице (например перестал работать `.clear()`)
- Изменение подписей элементов
- Jenkins
- Git
- Repository
- Дефекты фреймворка

## Что делать?

- Чинить
- Обходные пути / костыли
- Ждать

# Поддерживаемый код

**Что такое поддерживаемый код? Как это реализуется и зачем нужно?**

**1. Хороший стиль программирования**

- Внятное именование переменных
- Одна функция = одно действие
- Комментирование
- Обработка ошибок
- Форматирование текста

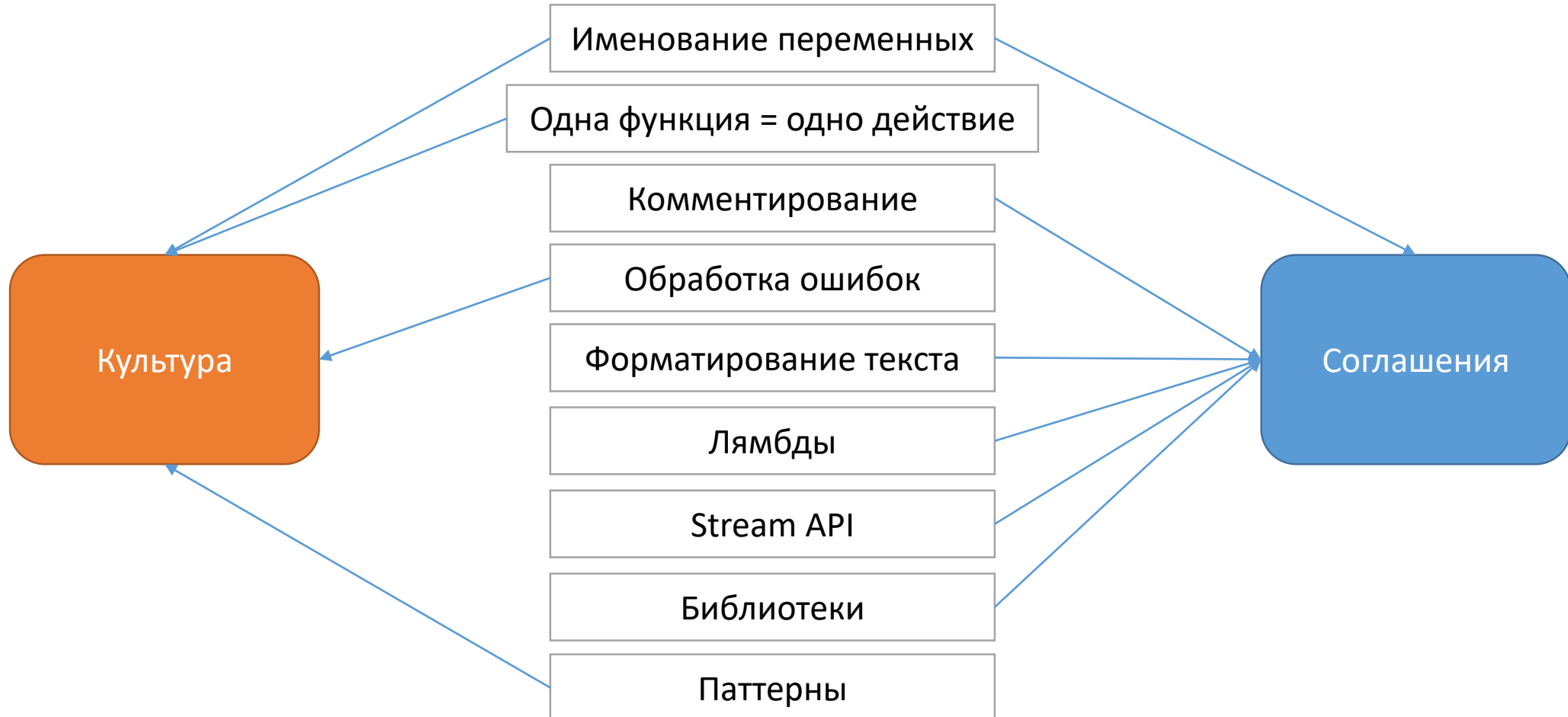
**2. Использование паттернов проектирования**

**2. Независимость модулей фреймворка**



# Культура и соглашения

Где проходит грань между действительно плохим кодом и просто недостаточно понятным?

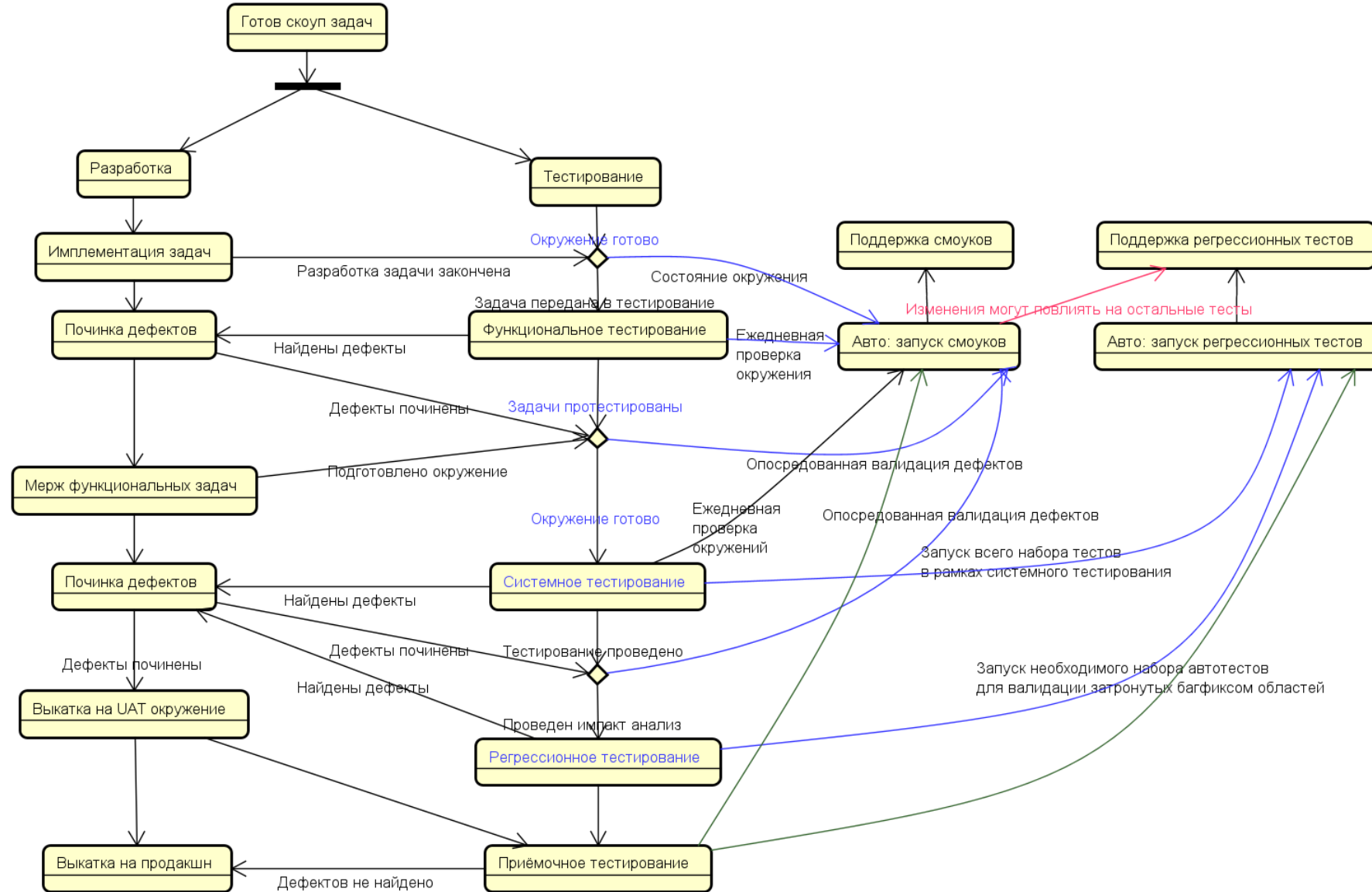


# Тезис

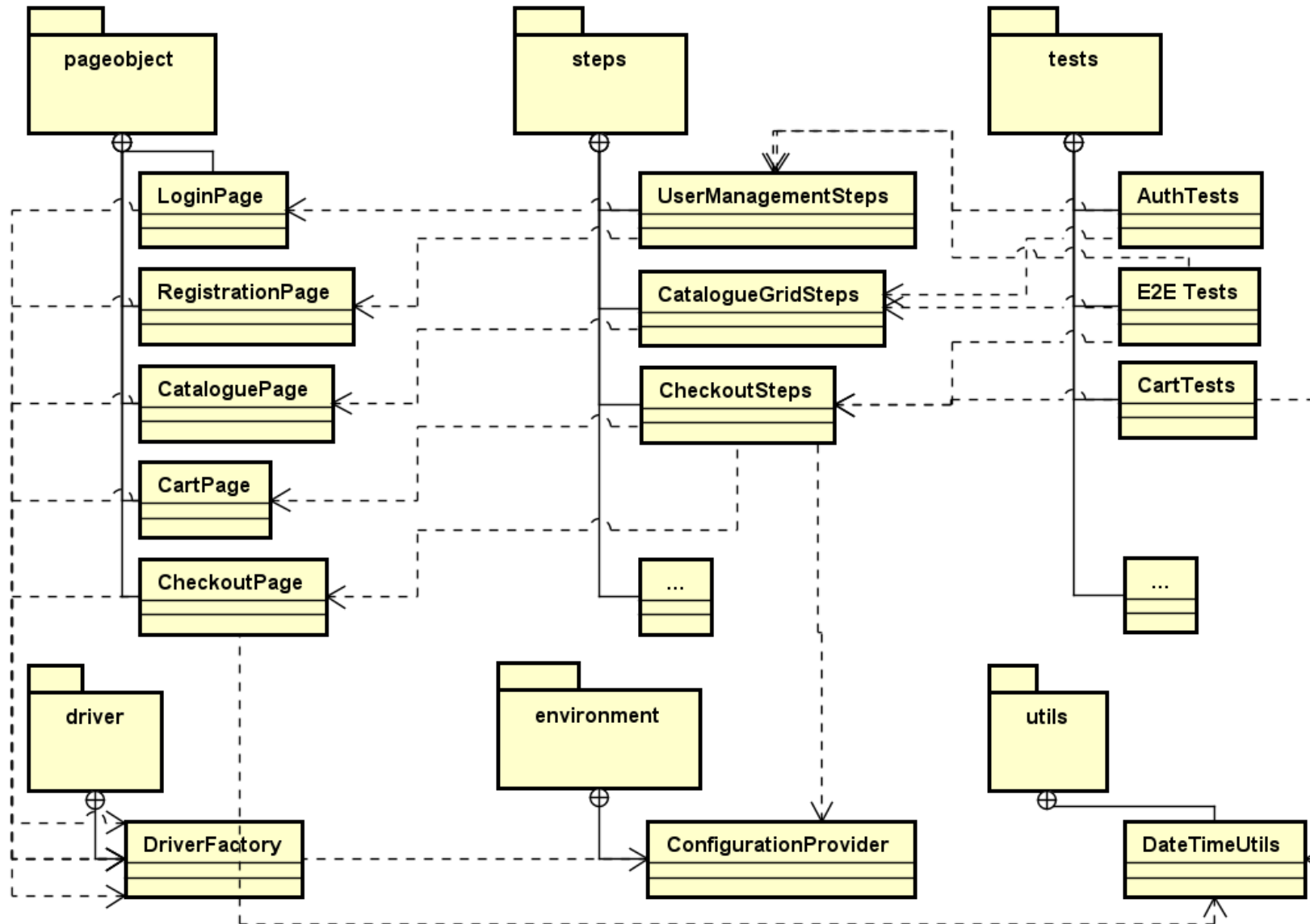
Будь гибким

# **Поддержка процесса тестирования с использованием средств автоматизации**

# Место автоматизации в тестировании

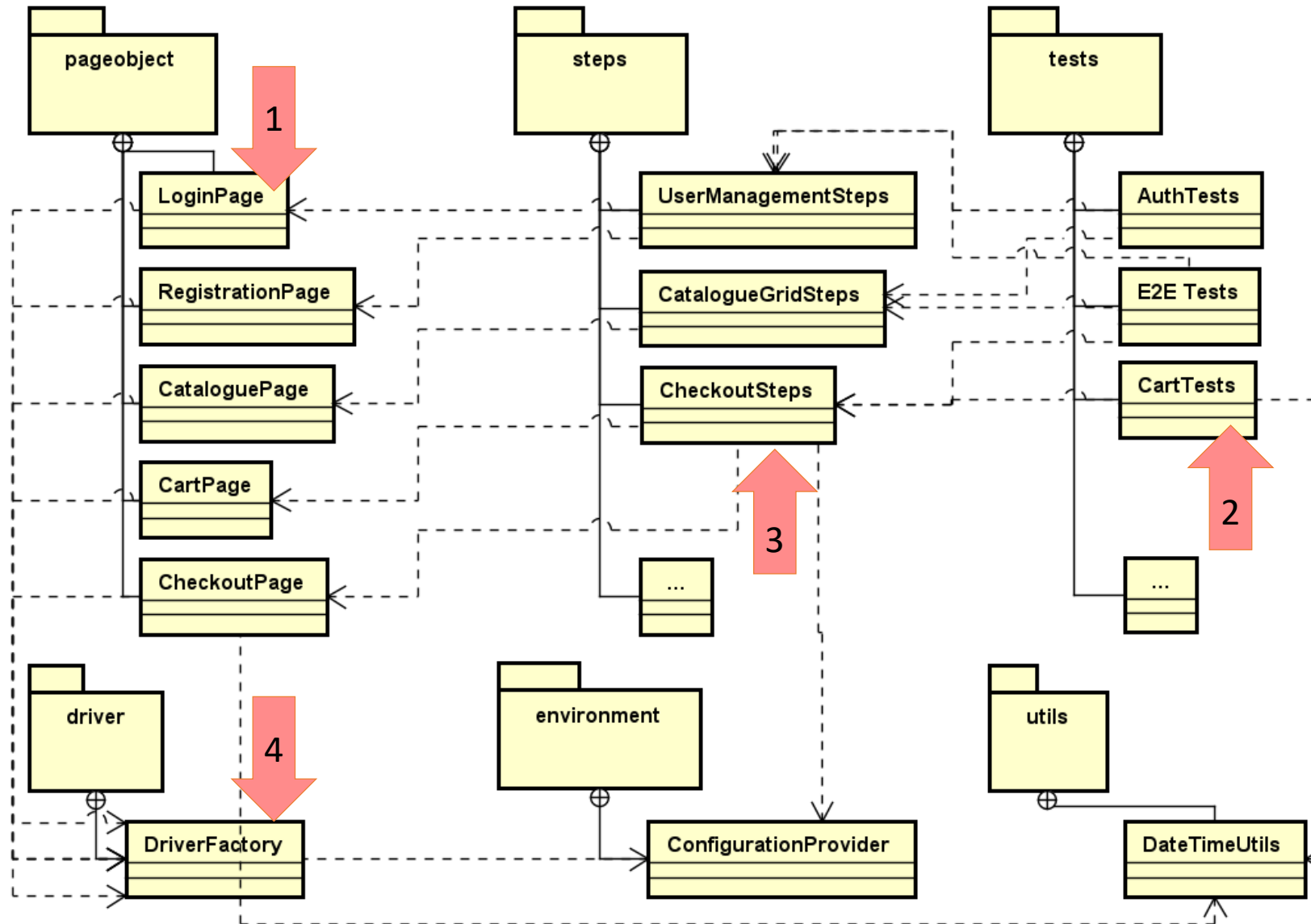


# Подходы к разработке фреймворка



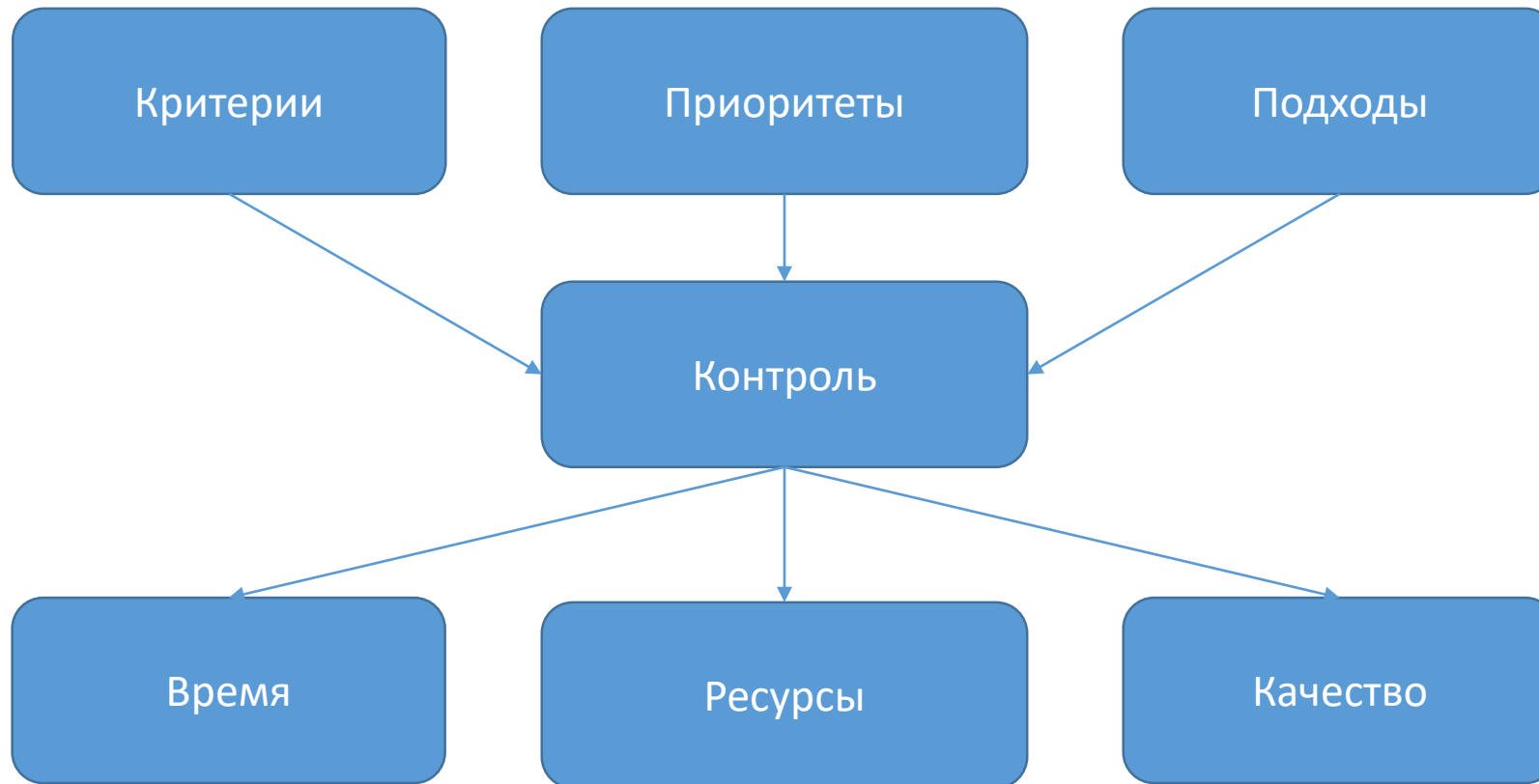
1. PageObject – только взаимодействие с элементами страницы
2. Steps – обёртка методов из PageObject в законченные операции с проверкой результата
3. Tests – набор шагов собранный из методов классов Steps

# Поддержка фреймворка



1. Изменение локаторов
2. Изменение логики UI
3. Изменение логики теста
4. Изменение ожидаемого результата
5. Изменение конфигурации
6. Проблемы с обработкой JavaScript
7. Проблемы неустойчивости сети или окружения

# Резюме



# Thank You!

think.  
create.  
accelerate.

## Contacts:

Ilya Demchenko  
[idemchenko@luxoft.com](mailto:idemchenko@luxoft.com)  
Skype: sooospinoy

Поделись мнением -  
**пройди опрос!**

[https://ru.surveymonkey.com/r/Seminar\\_28\\_May](https://ru.surveymonkey.com/r/Seminar_28_May)

