



Александр Федоров
software-testing.ru

Что нас ждет

- Особенности тест-дизайн при итерационной разработке
- Польза и спорная эффективность автоматизации тестирования
- Наследование тест-кейсов новыми и «родственными» версиями продукта
- Поддержание тест-кейсов в актуальном состоянии на разных этапах жизненного цикла продукта

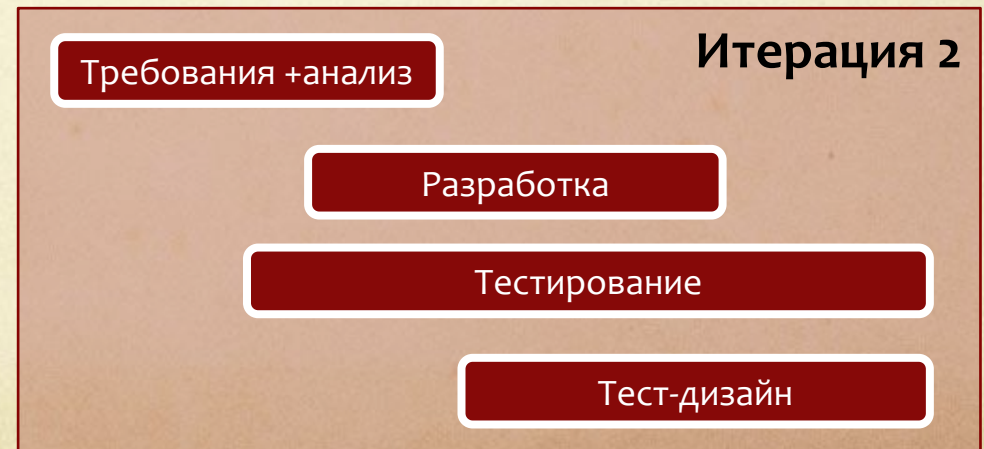
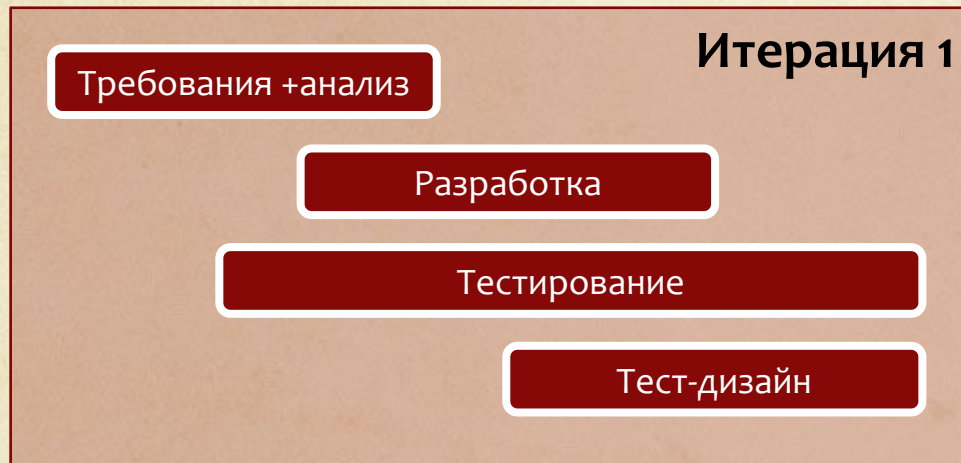
Продолжительность 1 астрономический час
Вопросы задаем по ходу



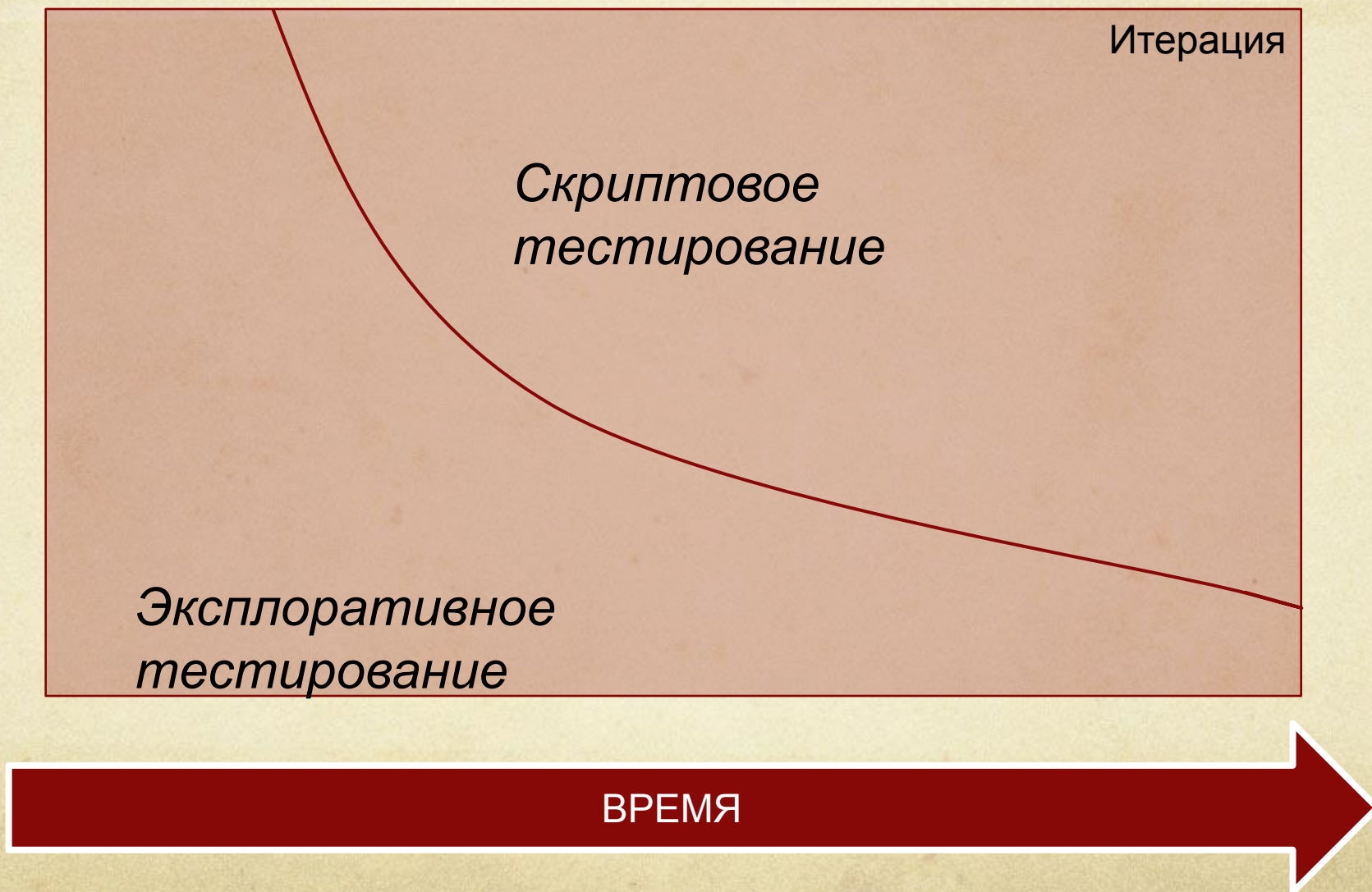
Итерационная разработка



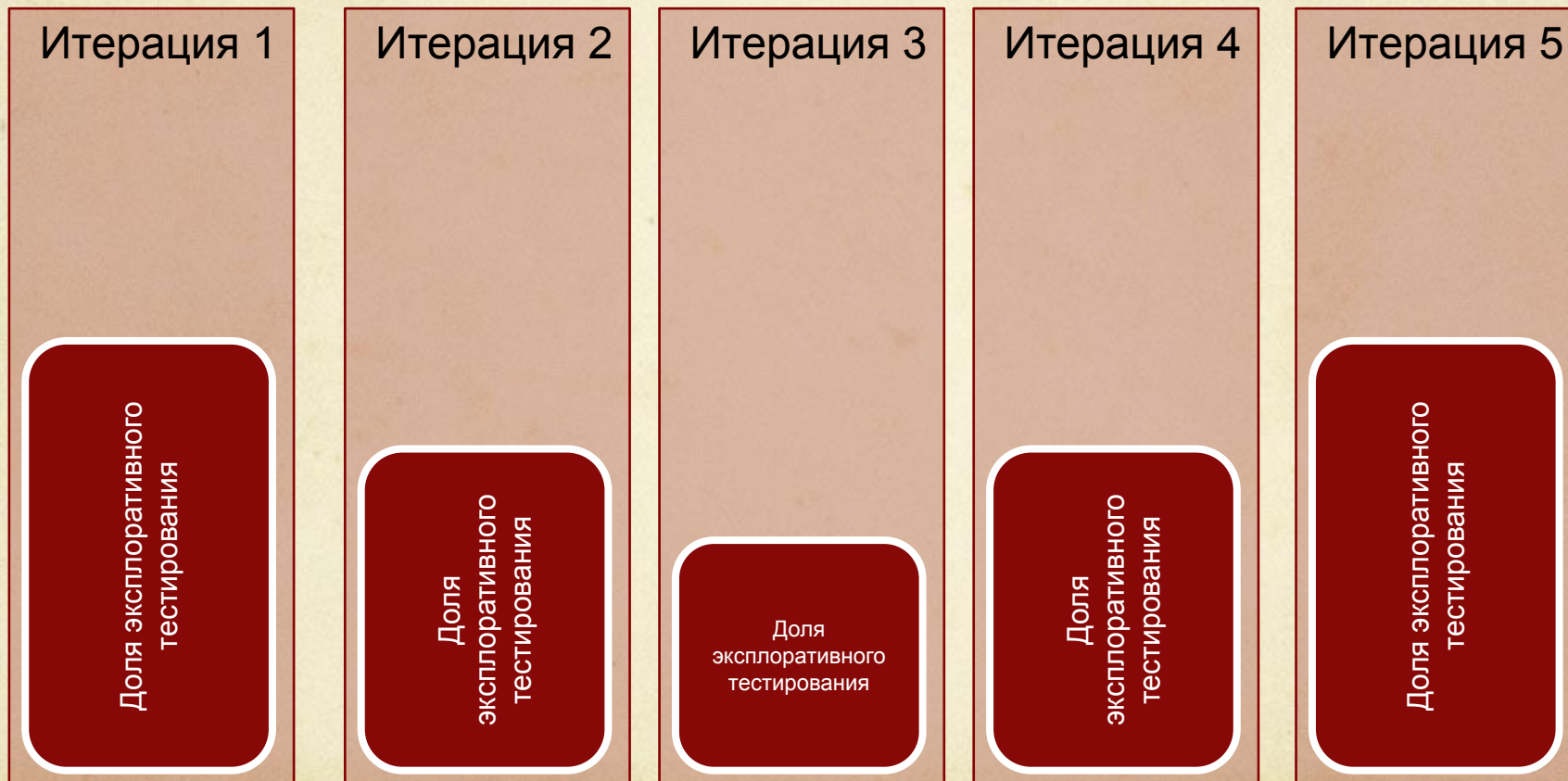
Тест-дизайн при итеративности



Доля эксплоративного тестирования в рамках итерации



Тенденция изменения доли эксплоративного тестирования



Общее число тестов



Изменение тестовых наборов при итеративности: **регресс**

Итерация 1

Набор 1

Набор 2

Набор 3

Итерация 2 (добавлен новый функционал)

Набор 1

Набор 2

Набор 3

Набор 4

Изменение тестовых наборов при итеративности: верификация билда (VVT)

Итерация 1

Тест 1

Тест 2

Тест 3

Итерация 2 (изменение компонента)

~~Тест 1~~

Тест 2

Тест 3

Тест 4

Итерация 2 (добавление компонента)

Тест 2

Тест 3

Тест 4

Тест 5

Автоматизация при итерационной разработке

Рекомендовано к автоматизации вне зависимости от итеративности:

- Верификация билда
- Регрессионное тестирование
- Нагрузочное тестирование
- Особое внимание на повторяющиеся операции

Особенности итеративности:

- Еще больше внимания регрессионному тестированию
- Прицел на операции будущих итераций

Поддержание эффективности тестового покрытия

Критерий эффективности:

Отношение количества найденных багов к
количеству пройденных тестов

Причины потери эффективности:

- Стабилизация продукта
- Эффект «пестицида»

ПРОБЛЕМА:

Тест-кейсы теряют
эффективность

РЕШЕНИЕ 1:

Изменить долю
эксплоративного
тестирования

РЕШЕНИЕ 2:

Изменить тестовое
покрытие

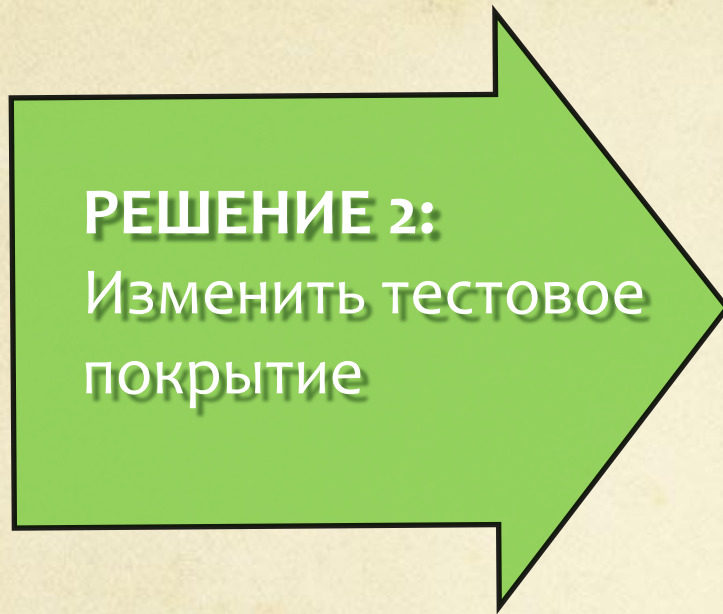
РЕШЕНИЕ 1:

Изменить долю
эксплоративного
тестирования

Снижение доли
использования
тест-кейсов в
пользу
тестирования
методом
свободного
поиска

Преимущества перехода к эксплоративному тестированию:

- не требует ресурсных затрат на изменение тестового покрытия
- иной путь выполнения того же самого теста
- выполнение редких и нетривиальных проверок
- «свежий взгляд» на тестовое покрытие



Преимущества изменения тестового покрытия - сохранение всех преимуществ тест-кейсов:

- Возможность использования начинающими сотрудниками
- Инструмент обучения и хранилище инструкций
- Облегчение заведения багов
- Наследование опыта в тест-кейсах

Повышение эффективности тестового покрытия

1. Аудит тестового покрытия
2. Покрытие кода
3. «Post mortem»:
 - Исследование багов
 - Анализ нахождения тестами багов: метрика количество багов в единицу времени при скриптовом / эксплоративном тестировании

Аудит тестового покрытия

Аудит проводится максимально возможным числом заинтересованных лиц.

Заинтересованные лица и чем они полезны:

Аналитик - определит полноту соответствия требованиям

Разработчик - подтвердит правильность используемых методов проверок проведения

Менеджер продукта - может помочь с определением приоритетов

Тест-менеджер - поможет определиться с полнотой/избыточностью тестового покрытия (возможно делегирует эту роль

Исследование покрытия кода

Покрытие кода (code coverage) - метрика, показывающая насколько исходный код программы был протестирован.

Основные способы измерения:

- *Покрытие операторов* — каждая ли строка исходного кода была выполнена и протестирована?
- *Покрытие условий* — каждая ли точка решения (вычисления истинно ли или ложно выражение) была выполнена и протестирована?
- *Покрытие путей* — все ли возможные пути через заданную часть кода были выполнены и протестированы?

Анализ результатов измерений позволяет детектировать как нехватку тестового покрытия, так и его излишки

Исследование багов

Выборка багов

Регламентирование работ

Исправление тестов

Необходимо ограничить круг багов:

- по принадлежности к части функционала (например, наиболее приоритетный)
- По критичности найденной баги
- использование флага «найдено по тест-кейсу» в системе баг-трекинга

Реализация наследования

Обеспечение переноса тест-кейсов
на новую версию продукта



Требования к ведению
тест-кейсов



Методика выявления
переносимых тестов

Требования к ведению тест-кейсов

Чем выше детализация тест-кейсов – тем больше вероятность того, что потребуется их редактирование в новой версии

Привязка к интерфейсу в тест-кейсах ведет к необходимости их редактирования

Можете ли вы себе позволить не детализировать тест-кейсы и не привязываться к интерфейсу?

Методика выявления переносимых тестов

Сравнивается функциональность версий продукта.

Вариант А: сопоставить чек-листы версий продукта

Вариант Б: сопоставить функциональные деревья версий продукта



ИТОГО

Итерационная разработка накладывает свой отпечаток на тестирование:

- Особый порядок использования видов тестирования
- Широкие возможности совершенствования тестового покрытия
- Дополнительное «ЗА» автоматизацию

Всем спасибо!

Просьба ко всем участникам оставить свои отзывы:

- Высказаться сейчас
- Прокомментировать в открытых источниках: на форуме, в блогах и т. д.

Узнать подробнее о всех превратностях тест-дизайна можно на тренинге «Тест-дизайн и все-все-все»

Расписание:

24-25 Сентября – Харьков

15-16 Октября – Нижний Новгород

29-30 Октября – Москва

18-19 Ноября - Самара

software-testing.ru