



Почему мы тестируем именно так?

Тестирование как управление рисками продукта

Григорий Сенин, Люксофт

No risk -- no test

“Good enough”:

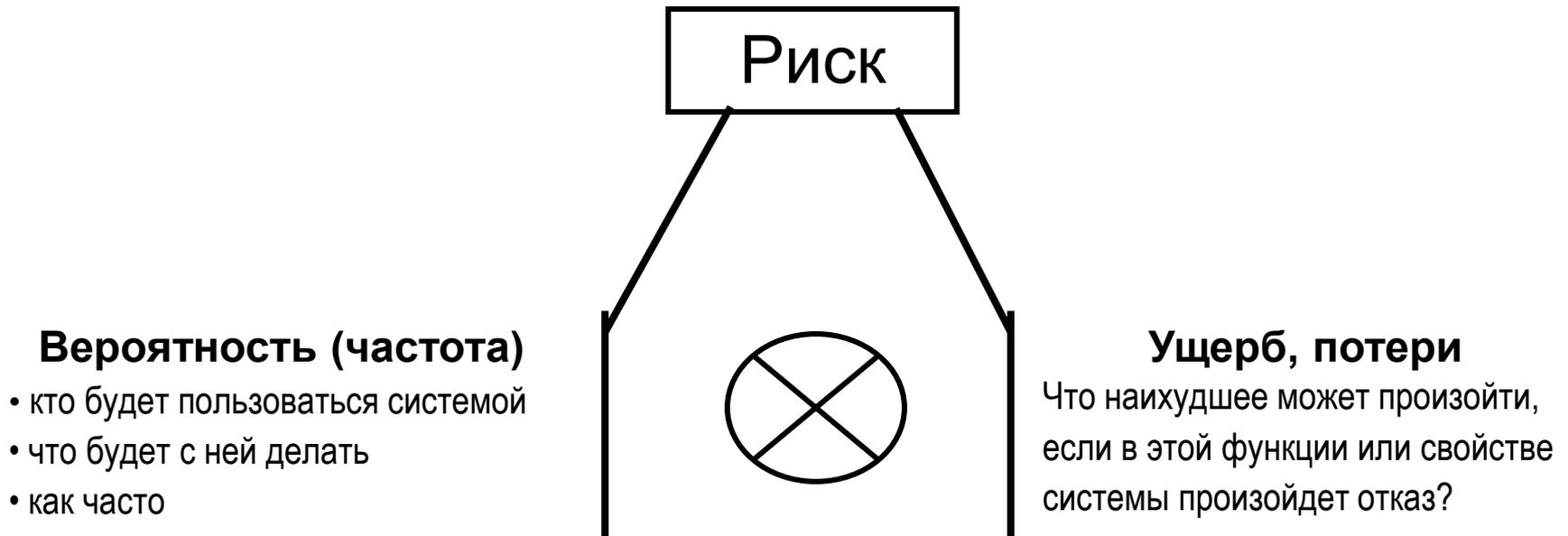
- Если тестировать в области риска становится дороже, чем нести потери от его осуществления, то тестирование нецелесообразно

Язык рисков

*Чем мы рискуем, если **сейчас** прекращаем тестирование и выпускаем продукт?*

- Чтобы ответить на этот вопрос, необходимо представлять себе риски продукта
- Области, таящие риск и НЕ охваченные тестами, трактуются как угроза продукту

Риски продукта



Анализ рисков устанавливает приоритеты
Больше риска – больше тестирования

Дефект – источник риска

- Проявление дефекта
 - неблагоприятное событие, которое может с некоторой вероятностью случиться при работе с продуктом
 - ⇒ с каждым дефектом, остающимся в продукте, связан риск
- Остающийся в продукте риск мог бы быть вычислен как суммарный риск от всех неисправленных, а также ненайденных дефектов

Примеры рисков, связанных с дефектами

- *«слишком долгий отклик системы»*
– потеря времени поль-ля
- *«неудобный интерфейс»*
– то же плюс невыполнение полезной функции
- *«отсутствие нужной функции»*
– негативное воздействие на бизнес-процесс предприятия пользователя
- *«крах системы»*
– потеря данных и др. тяжёлые последствия вплоть до потери исполнителем репутации
- При заказной разработке исполнитель может подвергнуться *штрафным санкциям*

Тестирование на основе рисков качества

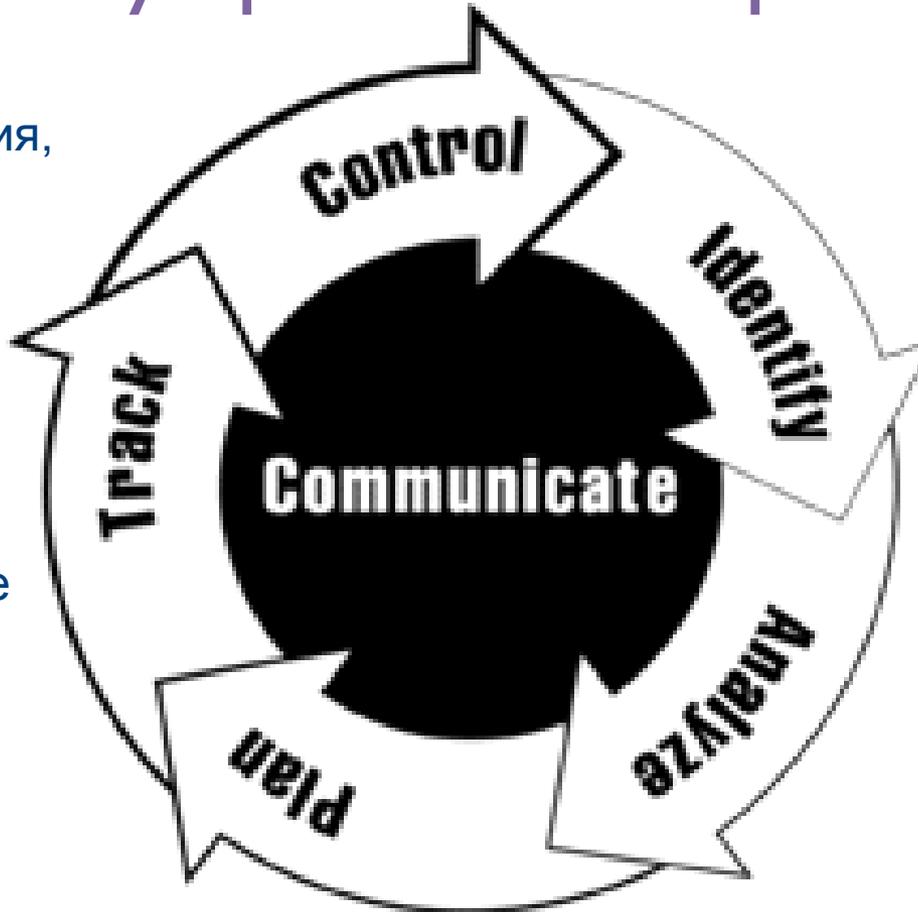
- **ЧТО** тестировать,
- что тестировать **раньше**,
- насколько **тщательно** тестировать,
- **когда завершать** тестирование.

Управление рисками =

определить, что для продукта существенно
при нехватке времени суметь пожертвовать несущественным

Тестирование с точки зрения управления рисками

- Измерения, отчёты



- Стратегия, подход

- Выполнение тестов

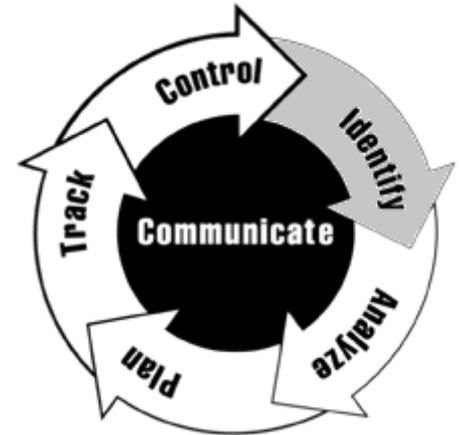
- Анализ и ранжирование требований

- Проектирование тестов

Тестирование с точки зрения управления рисками

Этап управления рисками	Деятельность группы тестирования	
Идентификация рисков	Стратегия тестирования	Получение перечня рисков
Анализ атрибутов рисков		Приоритеты тестирования; ранжирование требований
План сдерживания рисков		Тест-проектирование; планирование тест-циклов
Отслеживание рисков		Прогон тестов; регистрация дефектов,; верификация исправлений, отчёты
Контроль		Анализ результатов, метрики, коррекция приоритетов

Идентификация рисков



Три метода

1. от рисков к продукту: ведение общего перечня рисков
2. от продукта к рискам: анализ слабых мест продукта
3. от требований к рискам -- атрибуты качества

От рисков к продукту: перечни рисков

- пример

источник риска	код риска	категория риска	риск
Immature vendor capability	CUS-08	Technical Development Risk	Customer-supplied components have poor quality, resulting in extra testing, design, and integration work and in extra customer-relationship management.

Общие источники продуктовых рисков (Дж.Бах):

- «НОВОЕ»
- «СЛОЖНОЕ»
- «ВАЖНОЕ»
- «ИНТЕНСИВНОСТЬ»
- интерфейсы
- место поломки -- «где у них гнездо»
- ОПТИМИЗАЦИЯ
- ...

От продукта к рискам



Идентификация рисков по характеристикам качества

- **Functionality** – функциональные дефекты
- **Reliability** – низкая производительность
- **Usability** – неудобство пользовательского интерфейса
- **Efficiency** – неэффективная работа с вычислительными ресурсами
- **Maintainability** -- ...
- **Portability** -- ...
-- т.наз. **FRUEMP-методика** (по ISO 9126)

- **Уровень риска:** что случится, если требования данного типа не будут удовлетворены?
- Важность (приоритет, ранг) требования можно трактовать как размер потенциального ущерба

Анализ атрибутов рисков

Ущерб

- **Катастрофический:** крах системы, потеря данных
- **Серьёзный:** функция отсутствует
- **Средний:** функция отсутствует, но есть обходной путь
- **Незначительный:** неудобный интерфейс



Симптом нарушения требования

Критичность
требования

	функция недоступна	обходные пути, неочевидные действия	неудобство использования	косметические замечания, пожелания
высокая	фатальный	серьёзный	серьёзный	средний
средняя	серьёзный	серьёзный	средний	незначит
низкая	средний	средний	незначит	незначит

Анализ атрибутов рисков₂

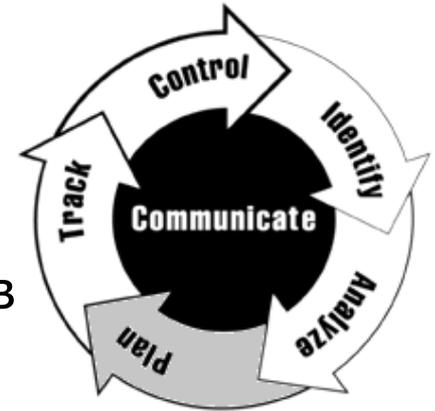
Вероятность – степень «видимости» дефекта

- **Всегда:** пользователь попадает сюда в каждой сессии
 - **Часто:** сюда «рано или поздно» попадает каждый пользователь, но не обязательно в каждой сессии
 - **Время от времени:** обычно здесь бывают только опытные пользователи
 - **Редко:** большинство здесь не бывает никогда
 - в эту область можно попасть только в результате весьма специфической последовательности шагов
- частоту функционального сценария
нужно учитывать при классификации дефектов

Критерии завершения тестирования

- минимизировать суммарный остающийся в продукте риск
 - Пример 1
 - Все критические тесты пройдены
 - Все критические или серьёзные дефекты закрыты
 - Пример 2
 - Все критические тесты пройдены
 - Все критические дефекты закрыты
 - Все незакрытые серьёзные дефекты проанализированы и величина стоящего за ними риска признана приемлемой

План сдерживания рисков продукта



- Сдерживание рисков \approx выявление дефектов
 - План тестирования – инструмент для этого
- Специфика продуктовых рисков:
 - Риски – рукотворные, дефекты вносятся в продукт!

**Риск можно «вызвать» в виде дефекта,
затем «разоблачить» и устранить**

Сдерживание рисков при тестировании

- Проектирование функциональных тестов = моделирование угроз продукту («провокация» рисков)
 - если функциональная область (или компонент) не покрыта тестами, это равносильно **принятию риска**
- Планирование тест-циклов
 - критичные компоненты/подсистемы тестируются раньше

На сдерживание рисков направлены также:

- Регрессионное тестирование (риск регрессии)
- Автоматизированное тестирование (успеть больше)

Приоритеты при тест-проектировании

Глубина проектирования функциональных тестов

вероятность/частота выполнения сценария

	высокая	средняя	низкая
высокая	глубокое		
средняя			
низкая			неглубокое

Отслеживание рисков

- Отслеживание рисков ведётся в форме отслеживания дефектов
 - Отчёты о тестировании, статистика
 - Суммарный риск продукта \Leftrightarrow метрики тестирования;
- Устранение рисков -- через обнаружение и исправление дефектов
 - более критичные сценарии прогоняются чаще



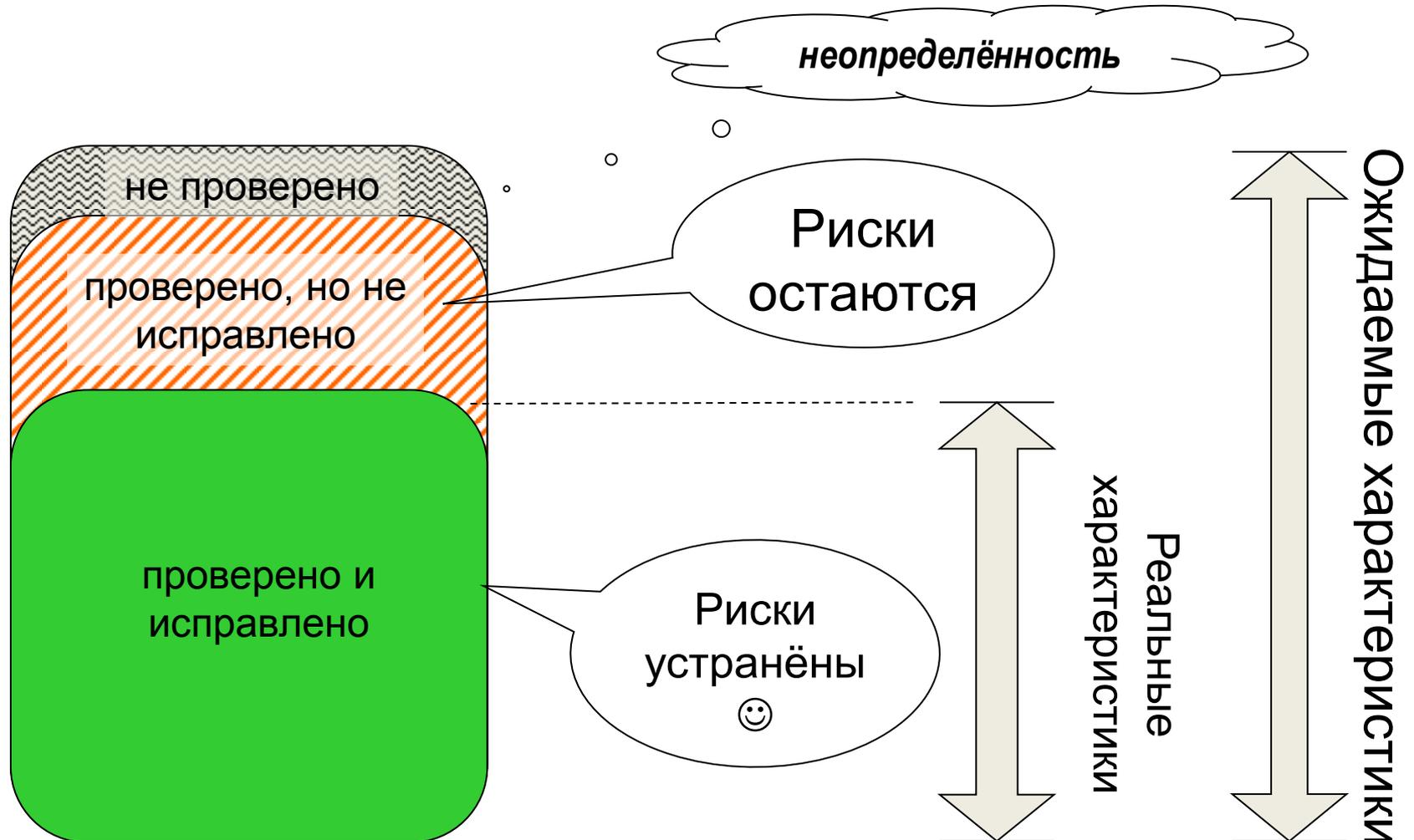
Управление риском в дефекте

- Идентификация риска – обнаружен дефект
- Анализ – дефект воспроизведён, изолирован, уточнён, обобщён, взвешен риск (серьёзность);
- Планирование – уточнены тест-спеки, контролируется статус дефекта (нужно закрыть)
- Отслеживание риска = судьба дефекта

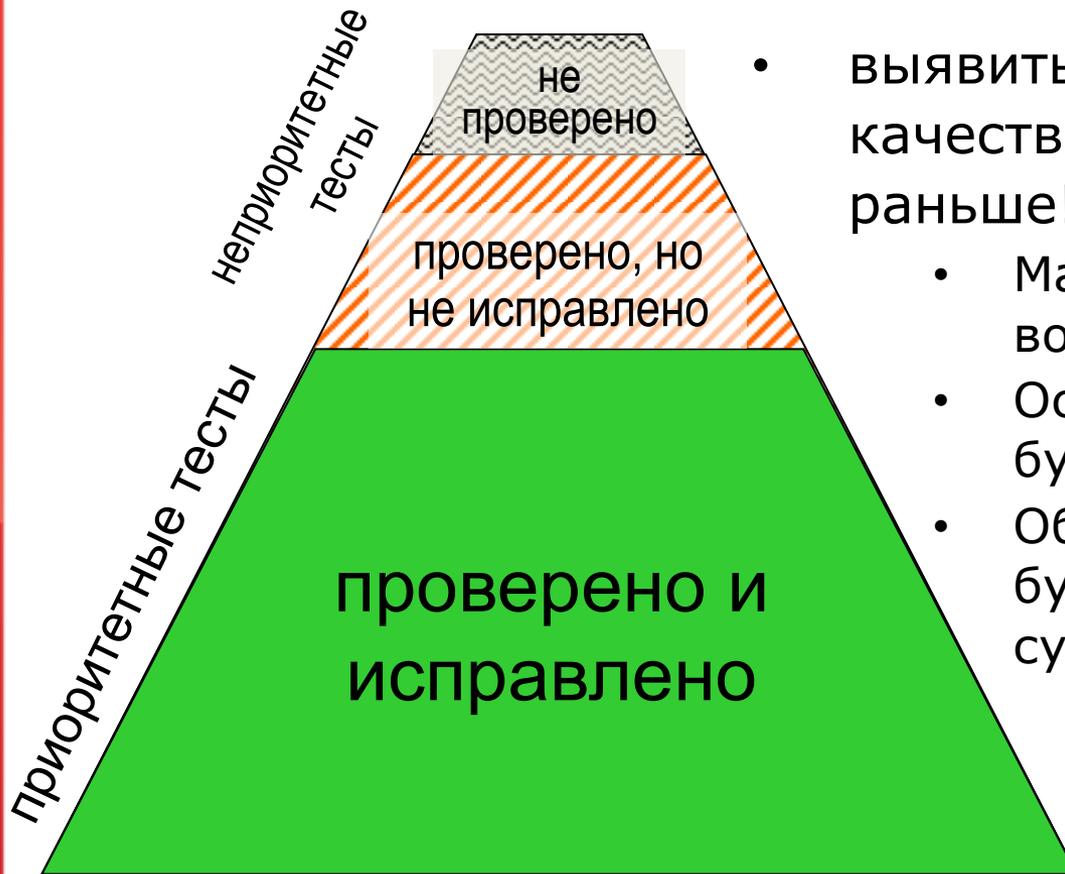
Отслеживание рисков – «судьба дефекта»

- Пока дефект не исправлен, риск в продукте остаётся
- “Risk elimination”: риск устраняется в результате исправления и верификации дефекта
- “Risk acceptance”: риск принимается, когда обнаруженный дефект решено не исправлять
- “Risk sharing”: когда не исправляется серьёзный дефект, его обычно необходимо разделить:
 - с менеджером проекта (проектное совещание)
 - с заказчиком (в ходе сдачи-приёмки)
 - с пользователем (описание дефекта в сопроводительной документации ~ Known Bugs/Workarounds)
- “Risk avoidance”: иногда мы пробуем избежать риска
 - переписать фрагмент кода, где коренятся дефекты
 - урезать код (даже путём сокращения функционала)

Типичное состояние продукта в конце разработки



Приоритеты помогают уменьшить риски продукта



- выявить самые высокие риски качества в продукте как можно раньше!
 - Максимальные риски будут вовремя устранены
 - Остающиеся в продукте риски будут на приемлемом уровне
 - Область неопределённости не будет источником существенных рисков

Questions?

