

Тестирование ПО. Урок 4

Требования, артефакты тестирования

Требования

1

1

Требования

Требование (requirement) — описание того, какие функции и с соблюдением каких условий должно выполнять приложение в процессе решения полезной для пользователя задачи.

Бывают

- Продуктовые требования (PRD - Product Requirement Document)
- Системные требования (SRS - System Requirement Specification)
- Функциональные требования (FRS - Functional Requirement Specification)
- Технические требования (TR/TS - Technical Requirements/ Technical Specification)

Все они отличаются степенью детализации, то есть каждый следующий документ содержит всё больше деталей реализации.

Требования. Ожидание/Реальность

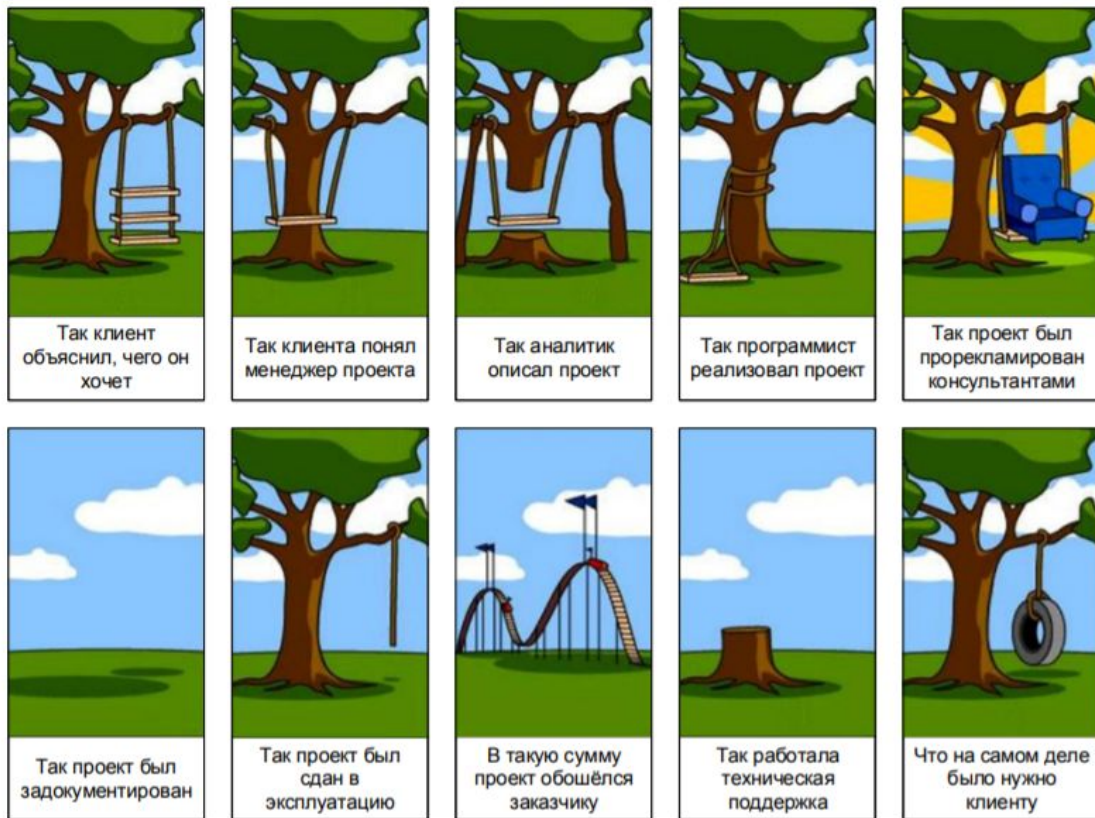


Рисунок 2.2.b — Типичный проект с плохими требованиями

Требования. Уровни

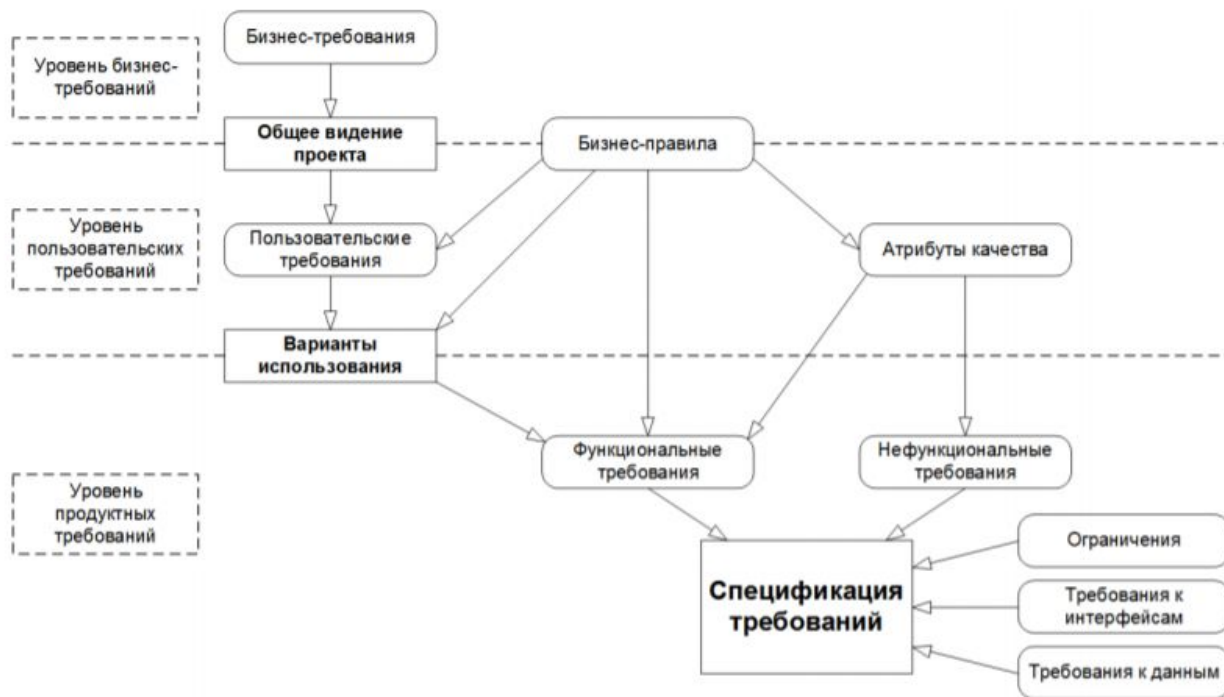


Рисунок 2.2.e — Уровни и типы требований

Требования. Свойства и качества



Рисунок 2.2.f — Свойства качественного требования

Артефакты тестирования

2

Артефакты тестирования

- **Тестовая стратегия, тест план** (Test Strategy, Test plan)
- **Тест кейсы** (Test Case)
- **Баг репорты** (Bug reports, defects, issues, etc.)
- **Тест репорт** (Test report)
- **Матрица соответствия требованиям** (Traceability matrix)
- **Чек-листы** (Checklist)
- **Автоматические тесты** (Automation test scenarios)

Тест план и тестовая стратегия

Тест план. Вопросы, на которые должен отвечать

Кто?

Что?

Когда?

Где?

Кем / Чем?

Как?

Тест план. Шаблон

1. Test subject

<should be information about what we are going to test in next format>

2. Feature to be tested: <feature name>

Requirements: <links to the requirements>

Jira tasks: <link to Jira epic>

3. Features Not to Be Tested:

<what and why we are not going to test>

4. Target release

<release when feature should be ready>

5. Resources

<list of QA team that is going to take part in testing>

6. Environments

<where we're going to test>

7. Test Approach

<how we are going to test feature. Which types/levels of testing. Are we going to automate, cover on unit test level, etc.>

8. Testing Tools

<which instruments will be used>

9. Risk Analysis & Dependency

<any risks, people, resource dependency that can hold on the project>

10. Exit criteria

<criteria of finishing testing >

Тест план и Тестовая стратегия

Тестовая стратегия (Test Strategy) – описывает общие для организации, независимые от проекта методы тестирования, объемы тестирования, подходы и тд.

Тест план (Test plan) – документ, описывающий объем (scope) тестирования и тестовых активностей (техники, правила, методы, уровни и тд.):

- Мастер тест план (master test plan) – описывает применение тестовой стратегии организации на конкретном проекте
- План тестирования уровня(или фазы) – описывает активности по тестированию определенного уровня или отдельной части программы

Тест кейс и тестовые системы

Тест кейс

Тестовый случай (Test Case) - это пошаговое описание необходимых действий для получения определенного результата

Характеристики хорошего теста

Хороший тест должен удовлетворять следующим критериям:

- Существует обоснованная вероятность выявления тестом ошибки
- Набор тестов не должен быть избыточным
- Тест должен быть наилучшим в своей категории
- Он не должен быть слишком простым или слишком сложным

Тест кейс. Test Rail

C1 Login to the system with correct name and password



Suit 1

Type	Priority	Estimate	References
Other	High	None	None
Automation Type			
None			

Preconditions

The user must be registered in the system

Steps

Step	Expected Result
1 Go to someurl.com	Page is opened. Login form appeared
2 Enter valid, already registered user name and correct password and press 'Login'	User successfully logged in Profile page is appeared

Тестовая система (Test management tool)

Система управления тестами (test management system) - это система, которая позволяет создавать, хранить, структурировать тест кейсы. В ней есть возможность управлять прогонами - выполнением тест кейсов, строить отчеты по результатам прогонов.

Основные функции:

- Тестовый редактор и просмотрщик (создание, удаление, перемещение, редактирование тест кейсов)
- Тестовые запуски (создание запуска, редактирование, удаление)
- Управление отчетами (создание отчетов, возможность обмена)

Тестовая система (Test management tool)

- **Test Link**
(<http://testlink.org>)
- **Test Rail**
(<https://brunoyamlearn.testrail.io/index.php?/cases/view/1>)
- **Hiptest**
- **Allure TestOps**
(<https://gameta.io/>)

Баг репорт и
баг трекинг системы



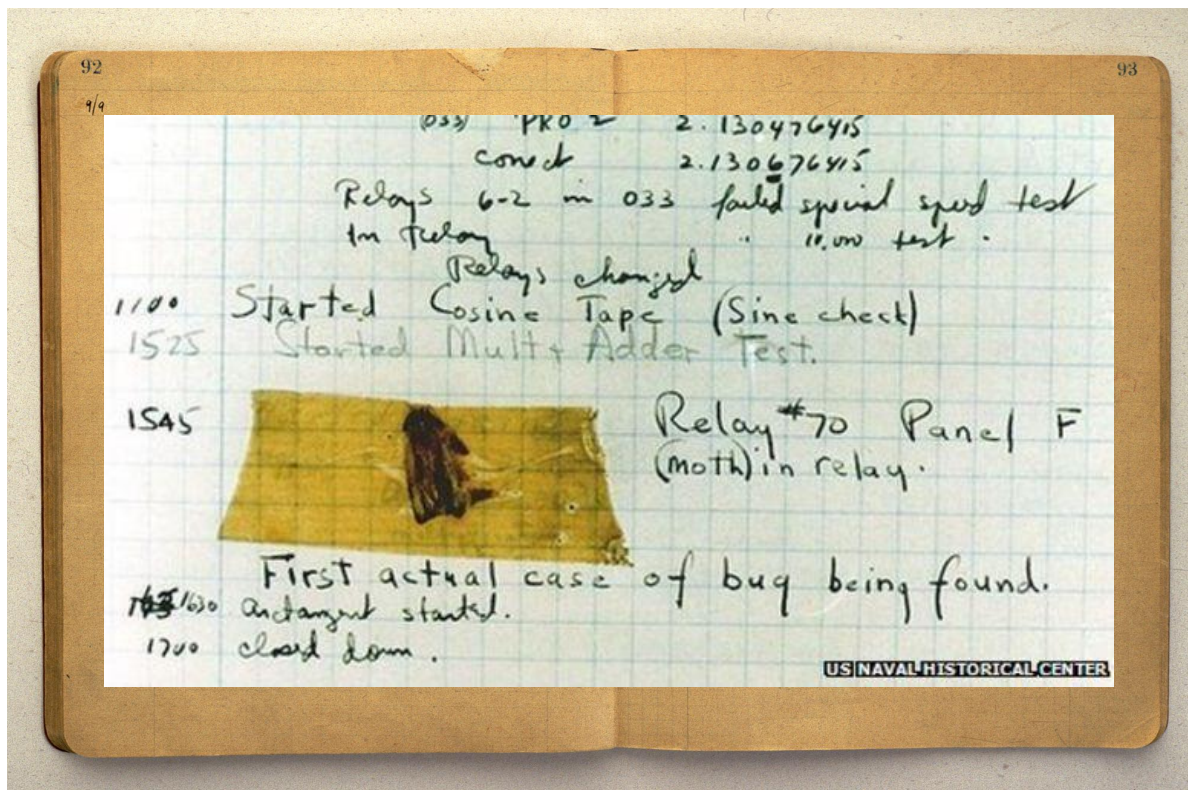
Что такое bug?

Баг (bug, defect) – это ошибка в коде или в работе программы.

Bug report – это отчёт об ошибке, в котором описано некорректное, несоответствующее требованиям поведение программного обеспечения



Первый баг репорт (1947)

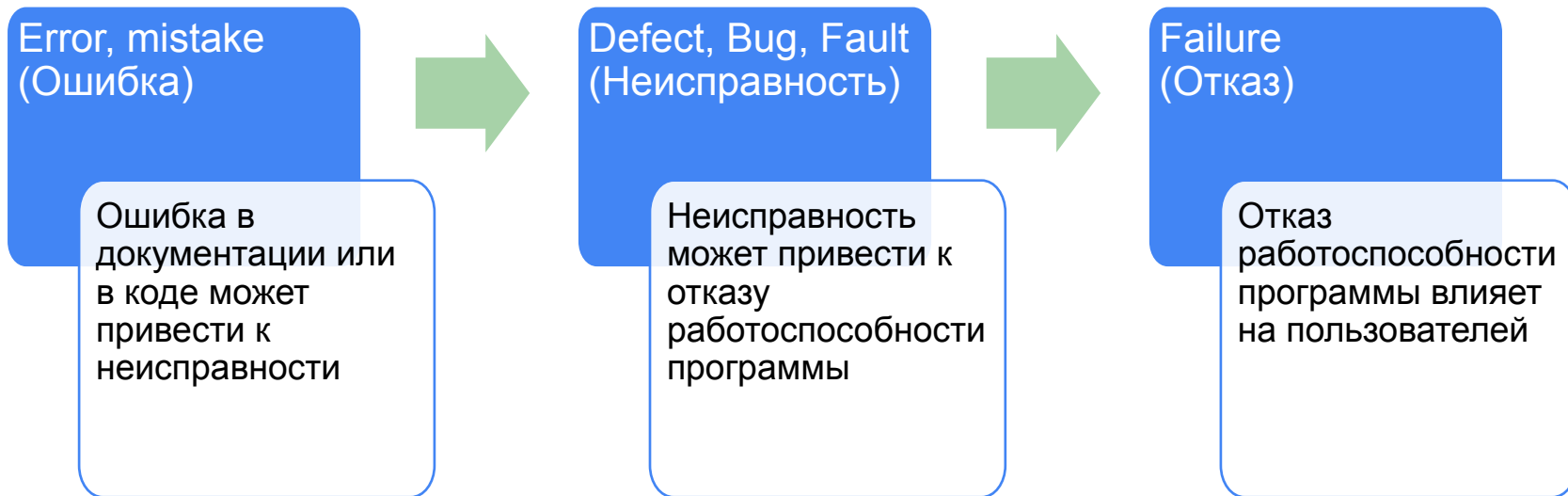


Структура баг репорт

Баг репорт должен содержать:

- Заголовок (title)
- Версию
- Приоритет/Важность (Priority/Severity)
- Короткое описание проблемы (Problem)
- Предусловия (Preconditions)
- Шаги воспроизведения (Steps to reproduce)
- Ожидаемый результат (Expected results)
- Фактический результат (Actual results)
- Техническую информацию (Technical info)
- Логи (Technical logs)
- Ссылка на тест кейс (Test case link)
- И любую другую информацию, которая поможет баг починить, воспроизвести и в будущем проверить

Bug, Defect, Failure, Mistake, Fault, Error



Priority/Severity

Серьезность (Severity) - это атрибут, характеризующий влияние дефекта на работоспособность приложения

Приоритет (Priority) - это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Можно сказать, что это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект

Определите Severity/Priority

- На главной веб странице сделана опечатка
- Приложение падает, если пользователь введет отрицательное число в поле возраст
- Приложение падает, когда пользователь вводит данные в поле, а потом их редактирует
- В логах приложения пишется сообщение с кодом ошибка, в то время когда в реальности ошибки нет(неправильный лог левел)
- Приложение перестает писать логи
- В интернет магазине нельзя добавить в корзину определенную группу товаров
- В интернет магазине невозможно произвести оплату

Советы на будущее

Советы

- Оптимизируйте тест дизайн (сокращение за счёт техник тест дизайна, за счёт области использования программы)
- Изучайте систему, которую тестируете (её область применения, аналоги, особенности)
- Изучайте технологии (протоколы, операционные системы, системы развертывания и тд.)
- Советуйтесь с разработчиками (спрашивайте реализацию, слабые места, показывайте свои тесты)
- Используйте автоматизацию в ручном тестировании (скрипты для заполнения, подготовки данных и др)
- Автоматизируйте важное
- Не гоняйте все авто(ручные) тесты. Чтобы этого достичь необходимо:
 - Приоритизировать тесты, делить их на области, функции
 - Выносить one-time тесты отдельно от всего

Вопросы?

Ресурсы

1. ISTQB Foundations,
<https://www.istqb.org/downloads/send/51-ctfl2018/208-ctfl-2018-syllabus.html>
2. IEEE 610.12-1990,
<http://www.mit.jyu.fi/ope/kurssit/TIES462/Materiaalit..>
3. QA fundamental terminology,
<http://softwaretestingfundamentals.com/>
4. Святослав Куликов, Тестирование ПО. Базовый курс,
https://svyatoslav.biz/software_testing_book_download/
5. Вики словарь тестировщика
<http://wiki.software-testing.ru>