Manual QA course

Lecture 4. Основные понятия в тестировании. Тестовые артефакты. Часть 2

Дорофеев Максим

hillel
компьютерная школа

Test Case

Артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для **проверки реализации** тестируемой функции или её части.

Идеальный тестовый случай(Test Case)

- Уникальный идентификатор тест – кейса;
- Название;
- Окружение (Опционально);
- Предусловия (Опционально);
- Шаги;
- Ожидаемый результат;
- История редактирования (Опционально).

Чего не должно быть в Test Case

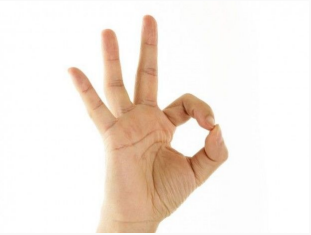
- Зависимостей от других Test Case;
- Нечеткой формулировки шагов или ожидаемого результата;
- Отсутствия необходимой для прохождения Test Case информации;
- Излишней детализации.

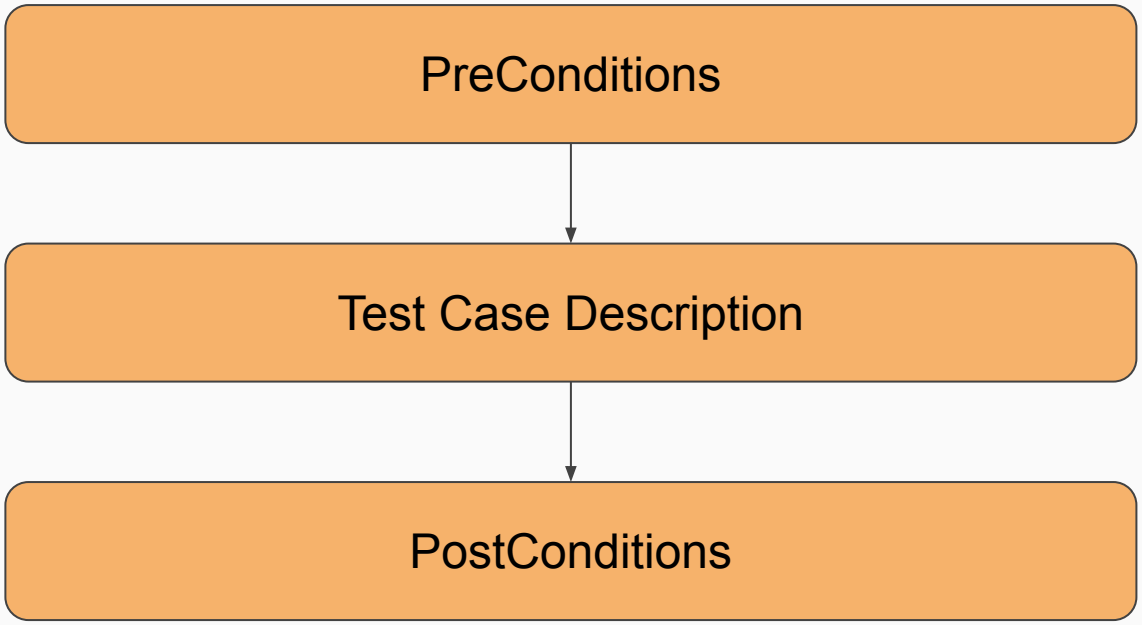
Test Case. Виды тест кейсов.

Позитивный тест кейс использует только корректные данные и проверяет, что приложение правильно выполнило вызываемую функцию.

Негативный тест кейс оперирует как корректными так и некорректными данными (минимум 1 некорректный параметр) и ставит целью проверку исключительных ситуаций (срабатывание валидаторов), а также проверяет, что вызываемая приложением функция не выполняется при срабатывании валидатора.

Test Case. Структура





Test Case. Пример

Тест-кейс № 1. Создание жильца без ФИО.

Шаги

1. Зайти на сайт www.dev_test.com (логин - test, пароль - test).

2. Войти под учетной записью администратора (логин - admin, пароль - 1)

3. Перейти на вкладку "Жильцы".

4. Нажать на кнопку "Создать карточку жильца".

5. Нажать на кнопку "Сохранить", не заполняя никакие данные.

Ожидаемый результат

Появляется сообщение об ошибке "Заполните обязательные поля, отмеченные *", карточка не сохраняется.

Test Case. Пример

Проверка отображения страницы		
Действие	Ожидаемый результат	Результат теста
Открыть страницу Логин	- Окно Логин открыто - Название окна - Логин - Логотип компании отображается в правом верхнем углу - На форме 2 поля - Имя и Пароль - Кнопка Логин доступна - Линк забыл пароль - доступен	...

Test Case. Пример

Steps to reproduce:
1. Open main page.
2. Click link Sign In.
3. Fill Email.
4. Fill Password.
5. Press button Sign In.

Expected results:
User should be logged to site.

Test Case. Пример с предусловием

Pre conditions (Optional):
User should be on forgot password page.

Steps to reproduce:
1. Fill Email.
2. Click button Reset Password.

Expected result:
User should get email with reset password instructions.

Test Case. Плохой пример

Тест-кейс № 01. Создание жильца.
Шаги:
1. Зайди на сайт www.production.com.

2. Нажми на кнопку "Войти" в правом верхнем углу экрана.

3. Авторизуйся с правами администратора.

4. Перейди на вкладку "Жильцы".

5. Нажми на кнопку "Создать карточку жильца".

6. Введи корректные ФИО, например, "Иванов Иван Иванович" и сохрани карточку.

Ожидаемый результат — карточка создана.

Test Case. Зачем?

«Планирование, и только потом – выполнение!»

Тест-кейсы дают нам структурированный системный подход, что снижает вероятность пропуска ошибки.

Test Case. Зачем?

Тест-кейсы – хороший способ хранения части проектной информации.

Test Case. Зачем?

Написание тест-кейсов – один из способов протестировать проектную документацию ещё до выхода первого билда.

Test Case. Зачем?

Наличие тест-кейсов значительно ускоряет регрессионное тестирование

Test Case. Зачем?

Test Case можно доверить выполнять новичку или призванному на помощь коллеге из другого отдела, который ничегошеньки о проекте не знает. Дополнительных вопросов с его стороны будет по минимуму — все и так (должно быть) понятно!

Test Case. Зачем?

Имея тест-кейсы, мы можем в любой момент «вспомнить», что мы делали месяц, полгода, год назад.

Test Case. Зачем?

Тест-кейсы позволяют легко отслеживать прогресс:

- X% тестов выполнено;

- Y% тестов прошло/завалилось;

- Z% требований покрыто тестами.

Test Case. Зачем?

- «Планирование, и только потом – выполнение!» Тест-кейсы дают нам структурированный системный подход, что снижает вероятность пропуска ошибки;
- Тест-кейсы – хороший способ хранения части проектной информации;
- Написание тест-кейсов – один из способов протестировать проектную документацию ещё до выхода первого билда;
- Наличие тест-кейсов значительно ускоряет регрессионное тестирование;
- Тест-кейсы – прекрасный способ быстро ввести в курс дела новичка или сотрудника, только что подключившегося к проекту;
- Имея тест-кейсы, мы можем в любой момент «вспомнить», что мы делали месяц, полгода, год назад;
- Тест-кейсы позволяют легко отслеживать прогресс (X% тестов выполнено, Y% тестов прошло (завалилось), Z% требований покрыто тестами).

Test Case. Целесообразны

- **Жизненно важные системы**, ошибка в которых может привести к гибели (самолетостроение, медицина, ПО для атомных станций);

- При тестировании **сложных систем** или **сложных частей** системы, чтобы не запутаться в чек-листе.

Test Case. Нецелесообразны

- Простые системы (веб-сайты, мобильные приложения и т. п.);

- Ситуации, когда в команде всего один или два тестировщика, знающие свой продукт. Время, потраченное на создание и поддержку тест-кейсов никогда не окупится.

Test Case. Рекомендации

- Начинайте с коротких тест-кейсов;

- Тест-кейс это не набор обязательных шагов;

- Перед написанием детализированных тест-кейсов запишите все, что можно протестировать в вашем приложении в вольной форме.

Test Case. Детализация Test Case

Это уровень детализации описания тестовых шагов и требуемого результата, при котором обеспечивается разумное соотношение времени прохождения к тестовому покрытию.

Test Case. Test Case Pass Time

Это время от начала прохождения шагов тест кейса до получения результата теста.

Test Case. Достоинства

- Время (приоритизация проверок);

- Более быстрое введение в проект новых людей или подключение коллег из других проектов для проведения сессии тестирования;

- Напоминание о конфигурировании и настройке системы;

- Незаменимы при работе над на «тяжелых» проектах;

Test Case. Достоинства

- Понимание информации одинаково всеми участницами процесса;

- Напоминание о старой функциональности, которую все еще нужно тестировать;

- Лучше, чем ничего)

Test Case. Недостатки

- Разные Test Case, для одного функционала очень похожи;
- Сложность поддержки;
- Неактуальное состояние;
- Следуя сценарию, можно упустить важные проблемы;
- Валидация неболшого кусочка функциональности;

Test Case. Недостатки

- Тестировщик проверяет продукт, а не тестирует его;

- Тестировщики выключают мозг, проходя Test Case;

- Любой может выполнять их, они не заменяют опытных тестировщиков, которые могут **тестировать**;

Check - list

Это список, содержащий ряд необходимых проверок для какой-либо работы. Отмечая пункты списка, вы, команда или специалист, можете узнать о состоянии или корректности выполнения этой

Check – list. Правила оформления

- Один пункт – одна операция;

- Пункты написаны в утвердительной форме;

- Оптимальное количество пунктов – до 20;

Check – list. Внедрение

- Тестирование
- Оформление;
- Удобный доступ.

Check - list. Пример

Проверка	Результат		
	Win XP	XP SP4	Win Vista
Операции с файлами	ok	ok	ok
Создание файла	ok	ok	ok
Открытие файла	ok	ok	ok
Сохранение документа	ok	ok	ok
Печать	ok	ok	ok
Редактирование файлов	bugs	bugs	bugs
Отмена	ok	ok	ok
Копирование	ok	ok	ok
Вырезание	bug #146	bug #146	bug #146
Вставка	ok	ok	ok
Удаление	ok	ok	ok
Поиск	bug #123	bug #133	ok
Поиск с заменой	bug #126	ok	ok

Check - list. Плохой пример

Проверка поля “Имя”		
Свое имя	Ольга	Регистрация прошла успешно, на почту отправлено письмо-приветствие.
Короткое имя	Ян	Регистрация прошла успешно, на почту отправлено письмо-приветствие.
Длинное имя (составное)	Розалинд Аруша Аркадина Алталун Флоренс Луна	Регистрация прошла успешно, на почту отправлено письмо-приветствие.

Check - list. Пример

Описание	Пример	Результат
Корректный email (популярный домен)	olga@mail.ru	Регистрация прошла успешно, на почту отправлено письмо-приветствие
Корректный email (корпоративная сеть)	olga@company.ru	
Точка внутри email	ok.molechka@gmail.com	
Кириллический email	олечка@муси ки.рф	На кириллический email ушло письмо
Пустая почта		Ошибка “Введите email”
Одно слово вместо домена	olgak@fdgfdg	Ошибка “Введите адрес в формате mail@site.com”

Чек лист. Зачем?

- Не забыть требуемые тесты.
- Для деления задач по уровню квалификации.
- Для сохранения отчётности и результатов тестирования.

Check – list. Преимущества

- Структурирование информации;

- Повышение скорости обучения новых сотрудников;

Bug Report

Документ, описывающий ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и ожидаемого результата.

Bug Report. Структура

- Короткое описание (Summary);
- Проект (Project);
- Компонент приложения (Component);
- Номер версии (Version);
- Серьезность (Severity);
- Приоритет (Priority);
- Статус (Status);
- Автор (Author);
- Назначен на (Assigned To);

Bug Report .Структура. Окружение

Информация об окружении, на котором был найден баг: операционная система, сервис пак, для WEB тестирования - браузер и его версия и прочее.

Bug Report .Структура. Описание

- Шаги воспроизведения (Steps to Reproduce)
Шаги, по которым можно легко воспроизвести ситуацию, приведшую к ошибке;

- Фактический Результат (Result)
Результат, полученный после прохождения шагов к воспроизведению;

- Ожидаемый результат (Expected Result)
Ожидаемый правильный результат.

Bug Report .Структура. Дополнение

Прикрепленный файл (Attachment)
Файл с логами, скриншот или любой другой документ, который может помочь прояснить причину ошибки или указать на способ решения проблемы.

Bug Report. Severity vs Priority

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.

Bug Report. Severity vs Priority

Priority - Показывает степень важности выполнения задач для **БИЗНЕСА**.

Severity - Показывает технологическую степень влияния **БАГА** на **ВСЮ СИСТЕМУ**.

Bug Report .Severity

S1 Блокирующая (Blocker)
Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)
Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)
Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 Незначительная (Minor)
Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)
Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

Bug Report. Priority

P1 Высокий (High)
Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

P2 Средний (Medium)
Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

P3 Низкий (Low)
Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

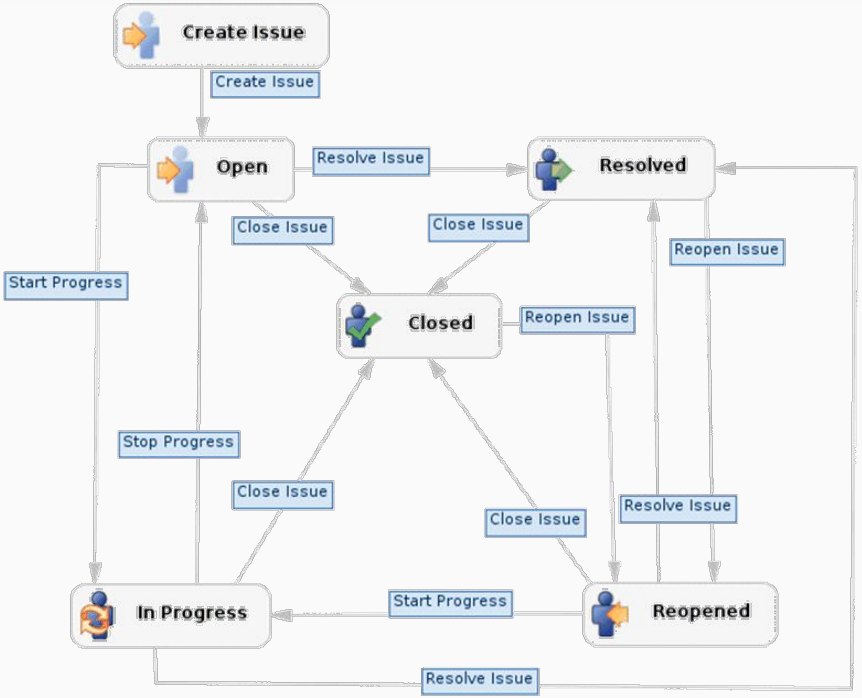
High -> Medium -> Low

Bug tracking system

Система отслеживания ошибок - прикладное средство учета информации, созданное для:

- Учета и контроля над ошибками и неполадками, найденными в программе
- Учета пожеланий пользователей
- Слежения за процессом устранения этих ошибок и выполнения или невыполнения пожеланий

Bug life cycle JIRA



Вопросы и ответы

Ссылки

http://www.protesting.ru/testing/
http://wiki.software-testing.ru/%D0%A7%D0%B5%D0%BA-%D0%BB%D0%B8%D1%81%D1%82
http://www.quizful.net/interview/qa/software-bug
http://okiseleva.blogspot.ru/2015/03/blog-post_33.html
http://www.protesting.ru/testing/bugreport.html
http://okiseleva.blogspot.ru/2013/08/blog-post.html
Lee Copeland. A Practitioner's Guide to Software Test Design
Severity and Priority difference
Severity and Priority
Severity vs Priority на пальцах

Домашнее задание

1. Составить **тест кейсы** (1 позитивный и 1 негативный) на регистрации нового пользователя в любой социальной сети;

2. Составить **баг репорт** (пополнение мобильного телефона через терминал банка. После введения тестового телефона 0770978675 и внесения денег в терминал,при нажатии кнопки «Пополнить», ничего не происходит, через 30 секунд терминал возвращается на главную страницу, деньги не возвращаются).
мой email dorofeev.maxim90@gmail.com
Письма с темой, фамилией и именем пожалуйста отправляйте на почту.