

# Topic 7. Test Cases and Checklists.

## Содержание

1. Checklists.
2. Test Cases.

## Тестовые Артефакты

В соответствие с процессами или методологиями разработки ПО, во время проведения тестирования создается и используется определенное количество **тестовых артефактов** (документы, модели и т.д.).

Наиболее распространенными тестовыми артефактами являются:

**Набор тест кейсов и тестов (Test Case & Test suite)** - это последовательность действий, по которой можно проверить соответствует ли тестируемая функция установленным требованиям.

**Дефекты / Баг Репорты (Bug Reports / Defects)** - это документы, описывающие ситуацию или последовательность действий приведшую к некорректной работе объекта тестирования, с указанием причин и

**Чек-листы — один из фундаментальных инструментов тестирования.** Они позволяют не забывать о важных тестах, фиксировать результаты своей работы и отслеживать статистику о статусе программного продукта.

Иногда чек-листами называют подробные инструкции о тестируемом продукте, содержащие последовательность действий, множество деталей и т.д. Это не так! Главный принцип чек-листов заключается в том, что каждый тестировщик по-своему проходит их, расширяя тестовый набор своей экспертизой.

**Какие преимущества чек-листов по сравнению с тест-кейсами:**

- ✓ нивелирование эффекта пестицида в регрессионном тестировании
- ✓ расширение тестового покрытия за счёт отличий при прохождении
- ✓ сокращение затрат на содержание и поддержку тестов: не надо писать много буковок!
- ✓ отсутствие рутины, которую так не любят квалифицированные тестировщики
- ✓ возможность проходить и комбинировать тесты по-разному, в соответствии с предпочтениями тестировщика

При этом, **чек-листы сохраняют множество плюсов**, за которые так популярны детальные тест-кейсы:

- статистика: кто, когда, что проходил (с детализацией по сборке продукта и окружению, на котором проводилось тестирование)
- памятка, которая помогает не забыть важные тесты
- возможность оценить состояние продукта, его готовность к выпуску

Конечно, было бы нечестно рассказать про плюсы и умолчать **о минусах чек-листов**:

- начинающие тестировщики не всегда эффективно проводят тесты без достаточно подробной документации
- чек-листы невозможно использовать для обучения начинающих сотрудников, так как в них недостаточно подробной информации заказчику или руководству может быть недостаточно того уровня детализации, который

**Итого**, выбор очевиден: если у вас высокая текучка, низкоквалифицированные сотрудники или этого требует руководство, выбора нет, и придётся создавать и поддерживать подробные, детальные тест-кейсы. Но если в вашей команде квалифицированные сотрудники, то чек-листы значительно удобнее и помогут вам получить максимум пользы от тестовой документации, **не тратя время на бюрократию!**

**Тестовый случай (Test Case)** - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.

Под тест кейсом понимается структура вида:

**Action > Expected Result > Test Result**

**Виды Тестовых Случаев**

**Позитивные и негативные:**

**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).

Процесс придумывания и написания тест-кейса (**test case generation**).

Процесс проверки — исполнение тест-кейса (**test case execution**).

- **шаги** – последовательность действий;
- **ожидаемый результат** —это некий предмет, который мы должны найти;
- **фактический результат** —это то, что мы реально нашли.

**Каждый** тест-кейс, исполнение которого **завершено**, дает нам **одно из двух**:

1. Положительный исход (**PASS**), если ФР равен ОР, либо
2. Отрицательный исход (**FAIL**), если ФР не равен ОР: **баг!**



Иногда возникает ситуация, когда мы заблокированы (test case execution is blocked), так как не можем пройти ВСЕ шаги тест-кейса. В таком случае мы рапортуем баг и откладываем исполнение тест-кейса до устранения бага.

### **Плохой стиль:**

1. Зависимость тест-кейсов друг от друга.
2. Нечеткая формулировка шагов.
3. Нечеткая формулировка идеи и/или ожидаемого результата.

**Data-driven** ("управляемый данными"), т.е.

когда данные и инструкции по их применению не смешаны, а разделены и слинкованы.

**Maintainability** (поддерживаемость), т.е. насколько легко и просто можно изменить тест-кейс при изменениях в ПО

# Test case attributes

УНИКАЛЬНЫЙ ID (ID)

ПРИОРИТЕТ (**Test Case Priority**)

ИДЕЯ (**Idea**)

ПОДГОТОВИТЕЛЬНАЯ ЧАСТЬ (**Add Info**)

ИСТОРИЯ РЕДАКТИРОВАНИЯ (**Revision History**)

ШАГИ (**Steps**)

ОЖИДАЕМЫЙ РЕЗУЛЬТАТ (**Expected Result**)

ФАКТИЧЕСКИЙ РЕЗУЛЬТАТ (**Actual Result**)

ПРИЛОЖЕНИЯ (**Attachments**)

...Others

Перечисляющиеся в тест-кейсе шаги должны быть объективно четкими и ясными. Нужно помнить:

- что очевидно для вас сейчас, может стать совершенно непонятным через пару месяцев.

Так, сокращенные шаги с нерасшифрованными аббревиатурами и прочими веселыми прибабасами, понятными вам сейчас, могут впоследствии стать китайской грамотой для вас самих, так что проще будет написать тест-кейс заново, чем пробираться через дебри неосмотрительно сделанных описаний;

- тест-кейс, который не может быть исполнен никем, кроме его автора, должен быть публично сожжен, растерт в порошок и развеян по ветру.

Нужно помнить, что ожидаемый результат — это информация, на основании которой (вкуче с фактическим результатом) мы принимаем решение об исходе тест-кейса. Следовательно, точность и четкость в формулировке ожидаемого результата играют наиважнейшую роль.

#	Item	Status	Comment
<b>Section One</b>			
<tag>1.1	Do something	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>1.2	Do something else	<input type="checkbox"/> Pass <input checked="" type="checkbox"/> Fail	
<tag>1.3	Do the same again	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>1.4		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<b>Section Two</b>			
<tag>2.1		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>2.2		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>2.3		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>2.4		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<b>Section Three</b>			
<tag>3.1		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>3.2		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>3.3		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>3.4		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<b>Section Four</b>			
<tag>4.1		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>4.2		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>4.3		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>4.4		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<b>Section Five</b>			
<tag>5.1		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>5.2		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>5.3		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	
<tag>5.4		<input type="checkbox"/> Pass <input type="checkbox"/> Fail	

# Литература:

1. <http://www.protesting.ru/>
2. <http://ru.qahelp.net/>
3. <http://habrahabr.ru/>
4. The Scrum Master Training Manual, v. 1.2., By Nader K. Rad, Frank Turley, Copyright © 2013 Management Plaza.
5. «Тестирование Дот Ком или пособие по жесткому обращению с багами в интернет-стартапах» Р. Савин.