Лекция 10 (ч. 1)
Тест-сеты, тест-сьюты и тест-кейсы. Правила их составления.

Практическое занятие 6
Создание тест-сета. Написание первого тест-кейса.

Содержание занятия:

- Что такое тест-сет, тест-сьют и тест-кейс.
- Правила оформления тест-кейсов.
- Практическое занятие 6: Создание тест-сета. Написание первого тест-кейса.

Тест-сет (Test Set, TS) – набор тестовых сценариев полученных на основании внутренней структуры компонента или спецификации, предназначенный для убеждения в 100% достижении заданных критериев покрытия (Smoke, Regression Testing).

Тест-сьют (Test Suite, TS) – комплект тестовых наборов для исследуемого компонента или системы, в котором обычно постусловие одного теста используется в качестве предусловия для последующего (Feature Testing).

Тест-кейс (Test Case, TC) – набор входных значений, предусловий выполнения, ожидаемых результатов и постусловий выполнения, разработанный для определенной цели или тестового условия, таких как выполнения определенного пути программы или же для проверки соответствия определенному требованию. [IEEE 610]

Но, проще говоря, **тест-кейс (ТК)** – это подробное описание проверки. Такое, которое можно будет дать человеку с улицы и он все поймет. В тест-кейсе есть название, предварительные шаги, шаги и результат. И куча других примочек, которые будут зависеть от стандартов оформления на вашей работе.

По сути, **Тест-сет** и **Тест-сьют (ТС) –** это практически одно и то же, поэтому мы будем использовать формулировку сьютов.

Грубо говоря, тест-кейсы это как файлики для нашей программы (всякие так экзешники, файлы справки, библиотеки и т. п.), т. е., всё, что относится к программе. А тест-сет/сьют – как папка для этих файликов, чтобы они не перемешались с файлами других программ.

Правила оформления тест-кейсов

Прежде, чем писать ТК, мы должны создать тест-сет/сьют (ТС) и заполнить необходимую информацию:

- название тест-сета/сьюта;
- приоритет ТС среди других подобных в этом спринте;
- на кого назначен ТС (как правило, на того же тестировщика, который писал ТК);
- название функционала, на который пишутся эти ТК.

Важно помнить, что ТС мы создаём для группы ТК, которые будут проверять определённый функционал (фичу) в данном спринте.

ТК имеет стандартные атрибуты:

ID — уникальный идентификатор. Если проставляется руками, то желательно делать его осмысленным. Часто генерируется автоматически.

Priority (приоритет тест кейса) – атрибут, который указывает приоритетность ТК, и принимает значение, согласно принятой в компании классификации (A-B-C-D-E, High-Medium-Low и тд.)

Req. ID – атрибут, указывающий на связанный с тестом пункт требования

Module – здесь указывается связанный с тест кейсом модуль.

Sub-Module – аналогично предыдущему пункту, только вписывается более мелкая структурная единица.

Test description (описание тест кейсов) – важный атрибут, в котором указывается заголовок (**SUMMARY**, суть теста), условия для его выполнения, шаги и действия после прохождения теста. Test description, соответственно, может и должен содержать:

- Pre-Conditions
- Steps to Reproduce (STR)
- Post-Conditions

Expected result (ожидаемый результат тест кейса) – данный атрибут отражает ожидаемый результат по каждому шагу.

Result – сюда заносятся данные о результате пройденного теста (passed/failed и др.)

Comment – сюда можно вносить свои примечания к тесту (вопросы заказчику, какие-либо наблюдения или детали).

Правила описания ТК:

1. SUMMARY:

- Пишем в стиле «**Что** (проверяем)-**Где** (проверяем) **Когда**/При каких условиях/Как именно (проверяем)»

Напр.: *Проверить, что пользователь авторизуется в систему **(ЧТО)** | с помощью правильно заполненной формы логина **(ГДЕ)** | и при нажатии кнопки «Вход» на ней **(КОГДА).***

-Используем «**Verify that** / **Check that** / **Проверить, что**» в начале Summary, чтобы указать на сам факт проверки.

Напр.: *Verify that user can be authorized in the system using properly filled login form and clicked on the “Enter” button on it.*

ВАЖНО: выбрав для себя вариант **Verify that** или **Check that,** придерживаться его во всех остальных ТК навсегда.

Можно использовать формулировку **«Verification of … / Проверка …»**, но это более чек-листовый формат, поэтому лучше всё-таки использовать рекомендации выше.

Также, лучше избегать словосочетания **«Make sure that …»**, т. к. чисто семантически это не означает проверку в значении **«протестировать»**, а значит **«убедиться, обратить внимание»** (но не вдаваясь в подробности). Обычно эту формулировку мы будем использовать в шагах нашего теста, чтобы заострить на чём-либо внимание тестировщика.

- Быть не длинным и не коротким.

Бывают случаи, когда мы не можем вместить всю информацию в **Summary**, тогда мы можем немного обобщить.

Напр.: у нас проверка отображения плитки товара на странице с товарами (проверка того, что плитка имеет изображение товара, артикул, название товара, цену, кнопку **«Купить»**).

Плохой пример: Verify that item tile has *item image, item article number, item name, price and **«BUY»** button* on the page with products.

Хороший пример: Verify that item tile has *specified attributes* on the page with products.

Дополнительно в этом поле может быть указан тип ТК (QA или DEV), принадлежность к User Story (US)/Use Case (UC) с её номером и сквозной нумерацией.

Напр.:

QA US4.1 Verify that system displays an additional occupancies section if service chosen by user has one or more occupancies

Этот ТК для исполнения тестировщиком, написан для US4, порядковый номер 1 (первый).

Аналогичный для разработчика будет такой:

DEV US4 Verify that system displays an additional occupancies section if service chosen by user has one or more occupancies

Часто DEV ТК порядковых номеров не имеют.

NOTE: если в системе есть триггер для пометки QA/DEV ТК, то смысла в **Summary** указывать эту инфу нет.

ИТОГО: Summary нашего ТК должен рассказывать о чём этот тест и как его можно пройти, т. е., по этому полю должно быть на 90% ясно что проверяется и как проверять, не открывая шагов, но при этом не быть слишком длинным, но и не слишком коротким (кроме тех случаев, если иное невозможно).

2. DESCRIPTION (как правило, отдельное поле сразу за **Summary**):
Краткое описание о чём тест. Как правило, в стиле ***Verification of login ability***

3. Pre-Conditions:
Это всё то, что поможет нам пройти тест-кейс, но прямого отношения к текущему тесту не имеет, т. е., подготовит почву для нашего непосредственного теста.
Напр.: регистрация нового пользователя в системе для проверки возможности логина.
- имеют собственную нумерацию.
- пишется в стиле «что-то должно быть сделано» (перед началом самого теста. ***New user should be registered in the system.***

- пишется обезличено: избегаем **«я/ты/мы/вы/I/you/we»**, вместо этого заменяем на **«user/client/customer»**.

User should see opened pop-up.

- писать нужно в едином стиле: все пункты предварительных шагов должны быть по одному шаблону («что-то должно быть сделано»)
- можно ссылаться на инструкции, документацию или другие ТК (но очень нежелательно, т. к. со временем файлы, на которые ссылаются, могут изменить своё расположение или вовсе быть удалены). В данном случае рекомендую создать свой файл-инструкцию или копию и приложить к ТК, сославшись в тексте шага, где он нужен, на него.

New user should be created (please refer to «User creation.doc») – for attached file OR ***(please refer to «User creation» TC)*** – for another Test Case

ВАЖНО: не увлекаемся слишком ссылками на другие ТК (а вообще стараемся избегать, помним, да?).

-предварительных шагов может и не быть вовсе – это нормально, ведь не всегда нам есть что сделать до начала теста (но, как минимум, зайти на тестовый энвайронмент мы всегда можем и описать это здесь).

4. STR:

Здесь описываем непосредственно шаги нашего теста.

- писать нужно в едином стиле: использовать нейтральные глаголы: ***click on / press / нажать, do / make / сделать, navigate / пойти, open / открыть.***
- можно прикреплять мокапы и ссылаться на них (зачастую для UI ТК) в шаге, где они уместны: ***«… (please refer to “Main Logo.jpg”)***.

- можно описывать тестовые данные для ввода в шагах. **Напр.:** у нас проверка возможности создания 2 пользователей с заполнением полей **«Имя», «Фамилия» и «Адрес».** Тогда шаг у нас будет таким:

N. Create 2 users with the next attributes:

USER 1: Ivan Petrov, Gogolevskaia St., 21, Kyiv;

USER 2: Ihor Ivanov, Lesi Ukrainki St., 124, Lviv.

А можно (и нужно) создать отдельный ***.xls** файл с вводными данными, прикрепить и сослаться на него, как с мокапами.

N. Create 2 users with defined attributes (please refer to «Users' data.xls»)

- каждый основной шаг должен иметь ожидаемый результат.

Step	Expected Result
N. Create 2 users with defined attributes (please refer to «Users' data.xls»)	2 users should be created successfully with defined attributes

- 1 шаг = 1 действие (как правило).
- иметь минимум шагов для максимального описания (тривиальные шаги можно объединять: *«Залогиниться в систему и открыть страницу X»).*
- последний шаг ТК – проверочный в стиле ***«Проверить, что <что-то работает как и ожидается>»***. Это обязательный шаг, т. к. в ТК может быть много шагов и к концу шагов тестировщик забудет на что ему нужно будет обратить внимание и может упустить цель теста. Тогда придётся проходить заново.

5. Post-Conditions:

Это те действия, которые приводят нашу систему в изначальное состояние. **Напр.:** если мы создавали пользователей в системе для теста, то по его окончании мы должны их удалить, чтобы следующий ТК проходить в идеальных условиях.

На практике так делают редко, ведь часто каждый предыдущий ТК – предусловие для следующего (при грамотном планировании ТК и описании проверок в хронологическом порядке). Поэтому, если Pre-Conditions может не быть, то Post-Conditions будет всегда (как минимум, закрыть браузер или вкладку, разлогиниться и т. п.).

- имеют собственную нумерацию.
- пишется в стиле «что-то должно быть сделано» (после окончания самого теста.

1. New user should be deleted from the system.

2. Browser should be closed.

Другие атрибуты ТК:

Assignee – ответственный за этот тест-кейс (правки, исполнение), QA или DEV (зависит от типа ТК - для тестировщика или разработчика)

Reviewer – ответственный за проверку соответствия ТК реквайрментам после написания

Product Owner – ответственный за фичу (разработчик реквайрментов)

Test Case Priority – приоритет для исполнения ТК

Automation Status – manual / automated

Time to Write – затраченное время на написание (конкретно этого)

Time to Run – оценочное время на прохождение ТК (конкретно этого)

Complexity – сложность, насколько сложные шаги (как много нужно сделать предподготовок, предусловий и т. п.)