**Test Case**

**Test case** – документ, который содержит в себе набор шагов для проверки одного ожидаемого результата**. В тест кейсе нету фактического результата**. Test case можно создавать в разных текстовых редакторах (word, excel, даже блокнот). Test case имеет два состояния:

1. **Состояние дизайна** test case **–** когда мы создаем инструкцию, для проверки одного ожидаемого результата.
2. **Состояние** **test execution –** когда мы уже выполняем тесты

Любой test case начинается с определения того, что мы будем тестировать и как правило это идея, которую предложил заказчик.

Тест-кейсы пишутся на основании тестовых условий

**Обязательные атрибуты тест кейса:**

1. **IDEA:** сначала, мы описываем, что в принципе мы собираемся сделать (напр. Проверить что мы можем пересохранить документ в HTML формате без искажения и после открыть его в браузере)

2. **Title:** Короткий заголовок/имя документа, который говорит, что мы сделали (напр. проверка сохранения документа в HTML формате и открытие этого документа в браузере)

3. **Environment**(опционально): пишется, если test case предназначен для конкретного окружения (напр. Test case создан для проверки в браузере Chrome).

4. **Pre-condition**(опционально): это какие-то действия, которые мы должны выполнить до исполнения test-case, чтобы мы могли пройти test case целиком. (напр. Чтобы мы могли проверить функцию аутентификации, в базе данных должен быть этот user. Чтобы мы могли проверить функцию logout, user должен быть залогинен).

5. **Steps** (Шаги): на этом этапе составления test case, мы должны описать какие действия мы должны выполнить. Напротив шагов должен быть и ожидаемый результат этого шага.

Шаги должны быть понятными. Любой участник команды должен понять смысл шага (Шаг должен быть однозначный). В шагах не рекомендуется писать тестовые данные.

**Пример:**

1. Launch text editor app (Открыть текстовый редактор).
2. Open any html file in Google Chrome (Открыть html файл в хроме, если нам не важен браузер, то мы просто пишем Open any html file in browser).
3. Make any change in content HTML document (Сделай любые изменения в контенте HTML документа).
4. Save change document (Сохрани изменения в документе)
5. Reload page browser (Обнови страницу браузера)
6. Observe the result on screen in your browser (Наблюдайте за результатом на экране вашего браузера)

После этих шагов у нас появляется Ожидаемый результат, из-за ожидаемого результата мы и делали все шаги.

6**.Expected result:** The document is displayed without any corruptions and shows changes made during steps (Документ отображается без каких-либо повреждений и показывает изменения, сделанные на шагах)

**Необязательные атрибуты:**

**Status:**

Когда мы будем выполнять test case (Состояние test execution), у него появится еще поле Status: где мы выбираем либо тест кейс был успешным, либо завалился, либо мы ставим блокировку Пример:

**Passed –** Статус ставится тогда, когда ожидаемый результат совпадаем с фактическим

**Failed** – Статус ставится тогда, когда ожидаемый результат отличается от фактического и мы создаем баг-репорт в баг трекинговой системе

**Blocked –** Статус ставится тогда, когда мы не можем дойти до последнего Steps, мы не знаем, является ли наш тест Passed/Failed.

**Приоритет:**

Высокий приоритет, означает, что этот шаг мы должны выполнить в первую очередь

**Связанный требования:**

Тест кейсы составляются на основании требований, которые у нас есть, а значит мы можем привязать текст кейс к конкретному требования.

**Преимущества Test case**

1. Все расписано детально, есть четкое последовательность шагов.
2. Мы знаем на каком steps мы фейлимся.
3. На основании test case, мы можем быстро создать баг-репорт
4. Мы можем узнать соотношение Passed/Failed тестов.

**Недостатки Test Case**

1. Test case требует довольно много времени на их написания