

Основы тест-аналитики

Позитивное и негативное тестирование



На этом уроке

1. Создадим тест-кейсы на основе декомпозиции
2. Объединим тест-кейсы в тестовые наборы
3. Научимся отличать позитивные тесты от негативных

Оглавление

[Атрибуты тест-кейса](#)

[Правила написания тест-кейсов](#)

[Заголовок](#)

[Предусловие](#)

[Шаги](#)

[Ожидаемый результат](#)

[Позитивные и негативные тесты](#)

[Создание тест-кейсов для Test IT](#)

[Тест № 1](#)

[Тест № 2](#)

[Тестовые наборы](#)

[Домашнее задание](#)

[Используемые источники](#)

[Глоссарий](#)

[Дополнительные материалы](#)

Атрибуты тест-кейса

Тест-кейс — это профессиональная документация тестировщика, последовательность действий направленная на проверку какого-либо функционала, описывающая как придти к фактическому результату.

Тест-кейс обладает рядом атрибутов.

Номер — уникальный идентификатор тест-кейса.

Название — краткое описание проверки по принципу “Что? Где? Когда?”

Предусловие — описание действий, которые необходимо выполнить перед выполнением теста.

Например, если для теста нужен пользователь в базе данных, то в предусловии описывается процесс создания пользователя. Если предусловия нет, то секция не заполняется.

Шаги — описание действий, необходимых для проверки.

Ожидаемый результат — что должно произойти после выполнения каждого шага.

Фактический результат - то, что произошло по факту после выполнения теста.

Каждый тест-кейс дает один из трех результатов:

- **Положительный результат**, если фактический результат соответствует ожидаемому результату,
- **Отрицательный результат**, если фактический результат не соответствует ожидаемому результату. В этом случае найдена ошибка.
- **Выполнение теста блокировано**, если после одного из шагов продолжение теста невозможно.

Правила написания тест-кейсов

Заголовок

Должен быть кратким, понятным и однозначно характеризующим суть тест-кейса. Не может содержать выполняемые шаги и ожидаемый результат. Название - это всегда основная идея тест-кейса. Из него должно быть понятно: что проверяет данный тест или какую ошибку он должен обнаружить.

“Проверка регистрации” - неудачное название, т.к. непонятно, что именно проверяется.

“Проверка регистрации” - пользователь зарегистрирован успешно” - тоже неудачное, т.к. содержит ожидаемый результат.

“Проверка регистрации нового пользователя по телефону” - удачное название.

Предусловие

Содержит полную информацию о состоянии системы или объекта, необходимом для начала выполнения шагов тест-кейса и ссылки на информационные источники, которые необходимо изучить перед прохождением тест-кейса (инструкции, описание систем...). Сюда выносятся информация, необходимая для теста, но не связанная напрямую с его идеей.

Предусловия может и не быть — это нормально. Не надо высасывать их из пальца там, где они не нужны. Предусловие отвечает на вопрос: что нужно сделать перед выполнением теста, чтобы при прохождении шагов не нужно было отвлекаться на дополнительные действия.

Шаги

Писать обезличено: “перейти”, “нажать”, а не “переходим”, “нажми”

Писать в едином стиле. Стиль написания тестов обсуждается участниками команды и фиксируется в Confluence или других системах.

Описание шагов должно быть максимально лаконичным. Сравним:

1. Авторизоваться в системе с правами администратора.
2. Авторизоваться в системе путем ввода логина и пароля, которые соответствуют учетной записи администратора.

И так понятно, что для авторизации нужно ввести логин и пароль, так что это уточнение в шаге лишнее.

Шаги не дают комментариев и пояснений. Инструкции оформляются в базе знаний, а в тестах указываются ссылки на нее.

Ожидаемый результат

Должен быть у каждого шага тест-кейса и однозначно описывать корректное поведение системы.

“Главная страница сайта отображается корректно” - это неправильное описание ожидаемого результата, т.к. непонятно, что значит “корректно”.

“Главная страница сайта отображается в соответствии с макетом” - это правильное описание ожидаемого результата, т.к. сравнение с макетом позволит обнаружить баги.

Постусловие

Содержит информацию том, как привести систему или объект в исходное состояние после завершения тестирования. Примеры постусловия:

- ВЫЙТИ ИЗ СИСТЕМЫ

- удалить созданные тестовые данные
- вернуть в первоначальное состояние измененные объекты

Позитивные и негативные тесты

Позитивное тестирование – это тестирование с применением сценариев, которые соответствуют нормальному (штатному, ожидаемому) поведению системы. С его помощью мы можем определить, что система делает то, для чего и была создана.

В рамках негативного тестирования применяются сценарии, которые соответствуют внештатному поведению тестируемой системы. Это могут быть исключительные ситуации или неверные данные.

Негативное тестирование направлено на проверку устойчивости системы к различным воздействиям, валидации неверных данных, обработки исключительных ситуаций. Сценарии позитивного тестирования направлены на проверку работы системы с теми типами данных, для которых она разрабатывалась.

Создание позитивных тест-кейсов предшествует созданию негативных тест-кейсов. Сначала мы проверяем работу системы, когда пользователь работает с системой “правильно”, а потом приступаем к проверке отклика системы на пользователя, который допускает различные ошибки (ввод неверных данных, например). Система должна быть готова ответить на неверный запрос. Это и есть цель негативного тестирования.

Наиболее распространенные негативные проверки:

- Обязательные поля для ввода данных.
 - Отмечены символом *.
 - Система сообщает об ошибке, если оставить поля пустыми или заполнить пробелами.
- Типы данных в полях.
 - Лимит на ввод символов.
 - Ввод недопустимых символов: буквы в номере телефона, цифры в имени и т.д.
- Числовые граничные значения.
 - Ввод значения, которые меньше минимально допустимых или превышают максимально допустимые
 - положить в корзину -5 штук товара
 - положить в корзину 10 штук товара, если в наличии есть 8 штук
- Валидность даты.
 - Ввод даты рождения в будущем.
 - Ввод даты рождения, которая противоречит бизнес-логике (возраст меньше 18 лет при оформлении кредита)
 - Несуществующая дата (32.13.2000)

Количество и тип негативных проверок зависит от того, в какой стадии находится проект. Чем более ранняя стадия разработки, тем больше времени требует негативное тестирование. Если продукт (или его модуль) долго работает без добавления новых функций, то негативное тестирование не будет эффективным.

Создание тест-кейсов для Test IT

Рассмотрим создание позитивного и негативного тестов для проверки системы Test IT. Вспомним чек-лист, который создавали в уроке 5. В нем мы обозначили 2 проверки поля “Название”. Расширим до тест-кейсов.

В чек-листе мы проверяли поведение отдельных полей: названия, описания и т.д. Для тест-кейса это слишком мелкая проверка, поэтому объединим их и напишем тест на пользовательский сценарий “Создание проекта”.

Тест № 1

Название: Успешное создание нового проекта

Предусловие: есть учетная запись пользователя с ролью Администратора.

Шаги

Шаг	Тестовые данные	Ожидаемый результат
1. Перейти в раздел “Проекты”	Роль пользователя - Администратор	Открыт раздел “Проекты”. Сверху есть кнопка “Новый проект”
2. Нажать “Новый проект”		Открылось окно создания нового проекта на вкладке “Общие настройки”.
3. Заполнить обязательные поля валидными значениями	см. чеклист	Поля заполнены
4. Нажать кнопку “Сохранить”		Проект открывается для работы с ним. Заполненные поля соответствуют указанным на предыдущих шагах

Тест № 2

Название: Негативные проверки создания нового проекта

Предусловие: есть учетная запись пользователя с ролью Администратора.

Шаги

Шаг	Тестовые данные	Ожидаемый результат
1. Перейти в раздел "Проекты"	Роль пользователя - Администратор	Открыт раздел "Проекты". Сверху есть кнопка "Новый проект"
2. Нажать "Новый проект"		Открылось окно создания нового проекта на вкладке
3. Заполнить название и описание проекта	<ul style="list-style-type: none">• название невалидное, описание валидное• название валидное, описание невалидное• название и описание невалидные Подробнее см. чеклист	Поля заполнены
4. Нажать кнопку "Сохранить"		Проект не сохраняется. Невалидные поля подсвечиваются красным. Под полем появляется сообщение об ошибке (см. документацию)

Тестовые наборы

Тест-кейсы объединяются в тестовые наборы по разным признакам.

1. По модулям: для каждого раздела программы - свой набор тестов.
2. По уровню тестирования
 - a. интеграционные тесты
 - b. системные (end-to-end) тесты
3. По приоритету
 - a. смоук-тесты
 - b. регрессионные тесты
 - c. тесты для приемки
4. По типам тестирования
 - a. функциональные тесты
 - b. нефункциональные тесты (UI, кросс-браузерность и т.д.)

Для определения теста в набор используются возможности TMS или метки.

Домашнее задание

Для страницы "Регистрация на портале чудес" написать тест-кейсы по требованиям:

https://docs.google.com/document/d/1fkCnsw9tDXd21uVDuT9biUkVZ1FlrhEj8s_vkoh2ls/edit?usp=sharing

Страница: <https://limelab.tech/test>

Тест-кейсы должны удовлетворять требованиям:

1. Составлены по шаблону с соблюдением правил составления тестов
2. Обязательно заполнены метки, в т.ч. "Позитивный / Негативный"
3. В наборе обязательно присутствуют и позитивные, и негативные тесты.

Тест-кейсы добавить в гугл-документ, который использовали для предыдущих дз на вкладку "Позитивные и негативные тесты"

Используемые источники

1. <https://habr.com/ru/post/246463/>
2. <http://okiseleva.blogspot.com/2014/08/blog-post.html> - Что такое тест-кейс и как его писать
3. <https://software-testing.ru/library/5-testing/66-top-13>
4. <https://habr.com/ru/post/481628/>

Глоссарий

Заголовок - обязательный атрибут тест-кейса, который однозначно характеризует его суть. Из него должно быть понятно: что проверяет данный тест или какую ошибку он должен обнаружить.

Предусловие - необязательный атрибут тест-кейса. Содержит полную информацию о состоянии системы или объекта, необходимом для начала выполнения шагов тест-кейса и ссылки на информационные источники, которые необходимо изучить перед прохождением тест-кейса (инструкции, описание систем...).

Шаги - обязательный атрибут тест-кейса. Последовательные действия, которые выполняются для проверки. У каждого шага должен быть свой ожидаемый результат

Постусловие - необязательный атрибут тест-кейса. Содержит информацию том, как привести систему или объект в исходное состояние после завершения тестирования.