

Что такое тест-кейс и как его писать

Тест-кейс — это проверка. "Выполни тест-кейс по вводу отрицательных значений" = проведи проверку такую-то и проверь, что результат будет такой-то.

Устоявшегося русско-язычного определения нет, помните об этом. Главное — понимать суть.

Тест-кейс — это такое описание проверки работы системы, которое может выполнить **любой** человек из команды, будь то тестировщик, разработчик, аналитик или даже бизнес-заказчик.

Набор тест-кейсов называется тестовым набором (test suite).

Иногда этот набор некорректно называют тест-планом. Тест-план — это именно план: когда, что, зачем, какими ресурсами.

Стандартные атрибуты тест-кейса

1. **Номер** — уникальный идентификатор тест-кейса. Его удобно использовать для одинакового понимания, о какой проверке идет речь (например, дать ссылку в баге).
2. **Название** — краткое описание **сути** проверки. Должно помещаться в твиттер и быть понятным! Кратко, но емко.
3. **Предварительные шаги** — описание действий, которые необходимо выполнить, но прямого отношения к проверке они не имеют (например, зарегистрироваться в системе для проверки создания элемента). Если предварительных шагов нет, то секция не заполняется.
4. **Шаги** — описание действий, необходимых для проверки (например, создание элемента).
5. **Ожидаемый результат (ОР)** — сама проверка: что мы ожидаем получить после выполнения шагов ("Элемент создан").

Пример оформления (один ожидаемый результат)

Есть внутренний сайт компании, которая проводит интернет "Самый_лучший_в_своем_роде" — www.test.ru. Тестовый стенд, на котором проверяются доработки перед выкладкой в PROD (он же *production*, окружение для пользователей) находится по другому адресу — www.dev_test.ru.

Примечание: www.test.ru — абстрактное обозначение некоего сайта, не надо туда заходить и искать эту систему. Приводится здесь, чтобы показать ошибки в написании тест-кейсов.

На сайте можно заводить карточки обслуживаемых зданий и карточки их жильцов. Карточки создает администратор, на тестовой машине всегда есть пользователь с правами админа, логин / пароль — admin / 1. При входе на тестовый сервер есть дополнительная авторизация, чтобы туда не могли попасть люди "извне", с логином и паролем test / test.

Тест-кейс № 1. Создание жилья без ФИО.

Шаги

1. Зайти на сайт www.dev_test.ru (логин - test, пароль - test).
2. Войти под учеткой администратора (логин - admin, пароль - 1)
3. Перейти на вкладку "Жильцы".
4. Нажать на кнопку "Создать карточку жилья".
5. Нажать на кнопку "Сохранить", не заполняя никакие данные.

Ожидаемый результат

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

Преимущества и недостатки тест-кейсов

Преимущества: тест-кейсы 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!

Недостатки (вытекают один из другого):

1. *Очень много копипасты много копипасты копипасты.* В примере выше заполняется поле "ФИО". Тест-кейсы "ввести в поле только символы, только числа, строку нулевой длины и т. д." будут очень похожи друг на друга, первые шаги одинаковые и, положив руку на сердце, будут копипаститься. Попробуйте написать хотя бы три тест-кейса на один функционал и сразу увидите эту проблему.
2. *Сложно поддерживать.* Представьте, что вкладку "Жильцы" переименовали в "Заказчики". Чтобы актуализировать тест-кейсы, надо внести изменения в сотни сценариев, что утомительно даже в режиме "*Ctrl + C, Ctrl + V*".
3. *Неактуальное состояние.* Тест-кейсы копируются друг от друга, и часто в них остаются неактуальные части из исходного кейса, которые забыли изменить.

Последний недостаток перечеркивает достоинства. Тестировщик, который уже год как работает на проекте, поймет и неактуальный кейс, тем более если выполняет их подряд, начиная с первого. А тестировщик, который ничего о проекте не знает и получил пару кейсов из середины тестового набора, не сможет понять, о чем в них идет речь.

Чтобы тест-кейсы честно выполняли свою роль, их надо поддерживать, периодически проверять на правильность и дорабатывать... Это отнимает очень много времени и сил.

Чтобы упростить этот процесс, могут быть использованы тест-кейсы с одним сценарием выполнения, но несколькими входными параметрами и разными ожидаемыми результатами. Фактически мы получаем мини чек-листы с предварительными шагами.

Примеры оформления (несколько ожидаемых результатов)

Рассматриваем все тот же абстрактный сайт www.test.ru. Допустим, что поле "ФИО" по ТЗ решили ограничить 40 символами (*тут будет ссылка почему так не надо делать*).

Когда говорят о нескольких ожидаемых результатах, это может означать:

- Даны несколько вариантов вводимых данных и для каждого прописан свой ожидаемый результат.
- Несколько шагов, а не только последний, содержат ожидаемый результат.
- После одного сценария выполняются сразу несколько проверок.

Несколько вариантов вводимых данных

Тест-кейс № 2. Создание жилья, проверка поля "ФИО".

Шаги:

1. Зайти на сайт www.dev_test.ru (логин - test, пароль - test).
2. Войти под учеткой администратора (логин - admin, , пароль - 1)
3. Перейти на вкладку "Жильцы"
4. Нажать на кнопку "Создать карточку жильца".
5. Заполнить поле ФИО (см "Ожидаемый результат")
6. Нажать на кнопку "Сохранить".

Ожидаемый результат

Вводимое значение	Ожидаемый результат
Киселева Ольга Евгеньевна	Ок, карточка сохраняется
<Оставить поле пустым>	Ошибка – «Заполните обязательные поля, отмеченные *», карточка не сохраняется
2*4*6*8*11*14*17*20*23*26*29*32*35*38*41*	Ошибка – «Максимальная длина поля – 40 символов, введено - 41», карточка не сохраняется. (Такую строчку легко сформировать с помощью инструмента perlclip)
&*%#(^\$@*&	Ошибка – «Поле ФИО может содержать только буквы русского алфавита», карточка не сохраняется
Kiseleva Olga Evgenievna	Ошибка – «Поле ФИО может содержать только буквы русского алфавита» (см. <i>статью про идиотов и ограничения</i>), карточка не сохраняется
...	...

Для этого варианта тест-кейса запись в виде таблички: данные – результат — наше всё!

Результаты для нескольких шагов из кейса

Другой вариант записи тест-кейса с несколькими ожидаемыми результатами — когда результаты пишутся на разные пункты шагов выполнения проверки, то есть на разные этапы сценария.

Тест-кейс № 3. Создание жильца с полным ФИО.

Шаги:

1. Зайти на сайт www.dev_test.ru (логин - test, пароль - test).
2. Войти под учеткой администратора (логин - admin, пароль - 1)
3. Перейти на вкладку "Жильцы"
4. Нажать на кнопку "Создать карточку жильца".
5. Ввести корректные ФИО, например, "Иванов Иван Иванович".
6. Нажать на кнопку "Сохранить".

Ожидаемый результат

1. Открывается окно ввода логина / пароля с соответствующими полями для ввода, кнопкой "Войти" и сообщением "Для входа в систему введите, пожалуйста, свои данные".
2. Вход в систему успешно осуществлен. В правом верхнем углу отображается надпись "Здравствуйте, admin". Открыта главная страница сайта.
4. Открылась страница "Создание нового жильца" с полями "Фамилия", "Имя" и "Отчество" и кнопкой "Сохранить".
6. Окно с информацией о жильце закрывается и отображается общий список, в котором присутствует новая карточка. Эту карточку можно открыть и на ней отображаются введенные данные, то есть в поле ФИО указано "Иванов Иван Иванович".

Несколько проверок после одного сценария

Тест-кейс № 4. Создание жильца с самым полным ФИО.

Шаги:

1. Зайти на сайт www.dev_test.ru (логин - test, пароль - test).
2. Войти под учеткой администратора (логин - admin, , пароль - 1)
3. Перейти на вкладку "Жильцы"
4. Нажать на кнопку "Создать карточку жильца".
5. Ввести корректные ФИО, например, "Иванов Иван Иванович".
6. Нажать на кнопку "Сохранить".

Ожидаемый результат

1. Окно с информацией о жилье закрывается и отображается общий список, в котором присутствует новая карточка.
2. Эту карточку можно открыть.
3. В открытой карточке отображаются введенные данные, то есть в поле ФИО указано "Иванов Иван Иванович".

Области применения

Так как тест-кейсы очень сложно поддерживать, то чаще используют чек-листы (*тут будет ссылка на статью по чек-листам*) или комбинацию "чек-листы & тест-кейсы".

В последнем случае большинство проверок пишут в виде чек-листов, а особо сложные (пойди туда, не знаю куда, принеси то, не знаю что, кувырknись три раза и громко крикни "ДЕДЛАЙН!", только тогда формочка и откроется) уже в виде тест-кейсов, чтобы каждый раз не вспоминать, как этот хитрый сценарий работает.

Тест-кейсы нужны:

1. Жизненно важные системы, ошибка в которых может привести к гибели (самолетостроение, медицина, ПО для атомных станций). Здесь надо тестировать очень аккуратно и тщательно.
2. При тестировании сложных систем или сложных частей системы, чтобы не запутаться в чек-листе.

Тест-кейсы не нужны:

1. Простые системы (веб-сайты, мобильные приложения и т. п.).
 2. Ситуации, когда в команде всего один или два тестировщика, знающие свой продукт. Время, потраченное на создание и поддержку тест-кейсов никогда не окупится.
- Познакомьтесь со своей системой и потом уже решайте, что подходит именно для нее — творческие чек-листы, формальные тест-кейсы или микс из этих подходов.

Стандартные ошибки при оформлении тест-кейсов

Читать теорию - одно, делать на практике - другое. Обычно в теории все понятно, а на практике получаем примерно такой кейс (все совпадения случайны, тест-кейс написан как агрегация различных ошибок):

Тест-кейс № 01. Создание жилья.

Шаги:

1. Зайди на сайт www.test.ru.
2. Нажми на кнопку "Войти" в правом верхнем углу экрана.
3. Залогинься с правами администратора.
4. Перейди на вкладку "Жильцы".
5. Нажми на кнопку "Создать карточку жилья".
6. Введи корректные ФИО, например, "Иванов Иван Иванович" и сохрани карточку.

Ожидаемый результат — карточка создана.

Попробуйте, не смотря ответ, сами найти проблемные зоны в этом тест-кейсе. А потом проверите себя 😊

Разберем ошибки кейса 01.

1. Абстрактное название

На первый взгляд название хорошее, короткое и понятное — мы ведь правда создаем жилья. Но! Если мы теперь создадим еще пяток тест-кейсов на ввод некорректных ФИО, то у них будет точно такое же название.

В итоге новый тестировщик, получив задание проверить кейс «Создание жилья», обнаружит в системе два десятка проверок с таким названием и впадет в ступор, какой выбирать?

Всегда помните про "кратко, но **емко**". По названию тест-кейса тестировщик, знающий проект, должен понять, что надо делать, не заглядывая в шаги. Так что дополняем название — *Создание жилья без отчества, Создание жилья, цифры в поле "Имя"* и т.д...

2. Повелительное наклонение

Чтобы коллегам было приятнее работать с тест-кейсами, лучше делать их описание обезличенным — "Выполнить, загрузить"...

3. PROD

В данном примере идет ссылка на PROD.

Никогда нельзя проводить тестирование на PROD-е! Исключение составляет дымовой тест, проводящийся после обновления PROD-системы. Тестовый набор для этого создается отдельно и тщательно выверяется.

ВСЕ остальное тестирование проводится **ТОЛЬКО** на тестовом стенде. В описании тест-кейсов и багов должны быть ссылки только на тестовый сервер. Иначе попросим коллегу с другого проекта помочь нам с тестированием, а он пойдет на PROD и ... или сломает что-то, или испортит реальные данные.

4. Нет ссылки на сайт

Написан URL, но не кликабельный. Нужно выделить, скопировать, открыть новую страницу, вставить... Гораздо лучше было бы просто нажать на него!

5. Слишком детализировано

Пункт "*Нажми на кнопку "Войти" в правом верхнем углу экрана*" содержит много подробностей про пользовательский интерфейс. Если кнопка в новой версии программы переедет в другое место, то придется вносить исправление и в тест-кейс. Чем меньше в документации зависимость от UI (*user interface*, пользовательский интерфейс), тем лучше.

Перепишем данный шаг: *Войти под учетной записью администратора (admin/1)*

Описание шага не стало менее понятным, и мы избавились от привязки к интерфейсу. Если вместо кнопки сделают ссылку или человек просто *Enter* нажмет, то суть шага не изменится: мы же в данном кейсе не логин проверяем, а создание жилья.

6. Нет нужной информации - непонятно, как авторизоваться

Есть пункт "Залогинься с правами администратора" — отлично, но как это сделать? Увидев этот пункт, я пойду искать кого-нибудь, кто в курсе, есть ли тестовый пользователь с такими правами и какие у него логин и пароль.

Если такой пользователь присутствует всегда (или создается каждый раз для тестирования), то есть его логин и пароль "статические" (не меняются), то их всегда надо прописывать, чтобы не возникало дополнительных вопросов.

Тест-кейсы составляются тогда, когда нужно, чтобы любой человек со стороны, не знающий проекта, мог присоединиться и помочь, выполнить тест-кейсы. Не задавая коллегам при этом дополнительные вопросы.

Тест-кейс я должна уметь выполнять, впервые увидев проект, там должна быть **ВСЯ** необходимая мне информация.

7. Нет описания проверки

"Карточка создана" — кратко, но не емко. Не имея знаний о проекте, тестировщик может только предполагать, что включает в себя этот пункт.

Достаточно ли того, что карточка закрылась без ошибок? Или она должна теперь отображаться в списке карточек? А сколько в системе таких списков? Должна ли система отображать введенные данные, если открыть карточку на просмотр? Что конкретно нужно проверять?

Поправим тест-кейс по всем замечаниям. Вот что получилось:

Тест-кейс № 02. Создание жилья с корректными ФИО.

Шаги:

1. Зайти на сайт www.dev_test.ru.
2. Войти под учетной записью администратора (логин - admin, пароль - 1)
3. Перейти на вкладку "Жильцы"
4. Нажать на кнопку "Создать карточку жилья".
5. Ввести корректные ФИО, например, "Иванов Иван Иванович".
6. Нажать на кнопку "Сохранить".

Ожидаемый результат

1. Окно с информацией о жилье закрывается и отображается общий список, в котором присутствует новая карточка.
2. Эту карточку можно открыть.
3. В открытой карточке отображаются введенные данные, то есть в поле ФИО указано "Иванов Иван Иванович".

Уже хорошо, но можно ли еще улучшить этот тест-кейс?

Сейчас снова попробуйте, не смотря ответ, найти проблемные зоны в этом тест-кейсе. А потом проверите себя 😊

Итак, ошибки кейса 02:

1. Абстрактное название.

Слова "корректный", "правильный" ит.д. в названии тест-кейса такой же маркер, как "ошибка" в названии бага. Таких слов надо избегать.

Позитивных проверок можно придумать хоть сто. Но чем-то они будут различаться. «Создание жилья, у которого нет отчества», — это тоже кейс с корректным ФИО. Только из такого названия сразу ясно, про что кейс.

Поэтому забудьте про слова "корректный", "некорректный" и т.п., пытайтесь писать понятнее. И всегда помните принцип "кратко, но **емко**". А разделение кейсов на смысловые группы (негативные тесты, позитивные тесты, тесты на особые случаи) сделайте в системе управления тест-кейсами через флаги или отдельные наборы тестов.

2. Нет нужной информации

Зайти на сайт www.dev_test.ru

Ок, я открываю этот сайт, а там авторизация. Как мне туда попасть?

Никак! Идти и узнавать логин/пароль. А зачем, если это легко было исправить указанием логина/пароля в скобках или ссылкой на страницу со всеми логинами и паролями (они

все же могут меняться и лучше менять в одном месте)?

Исправленная версия тест-кейса:

Тест-кейс № 03. Создание жилья с полным ФИО.

Шаги:

1. Зайти на сайт www.dev_test.ru (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 жилья".
5. Ввести корректные ФИО, например, "Иванов Иван Иванович".
6. Нажать на кнопку "Сохранить".

Ожидаемый результат

1. Окно с информацией о жилье закрывается и отображается общий список, в котором присутствует новая карточка.
2. Эту карточку можно открыть.
3. В открытой карточке отображаются введенные данные, то есть в поле ФИО указано "Иванов Иван Иванович".

Определения из книг по тестированию

Ron Patton. [Software Testing](#).

Test cases list the specific items that will be tested and describe the detailed steps that will be followed to verify the software.

Тест-кейсы перечисляют конкретные вещи, которые будут протестированы, и описывают детальные шаги, которые необходимо выполнить для проверки программного обеспечения.

Lee Copeland. [A Practitioner's Guide to Software Test Design](#).

The purpose of the test case specification is to specify in detail each test case listed in the test design specification. The test case specification is composed of the following sections:

- Test case specification identifier - A unique identifier so that this document can be distinguished from all other documents.
- Test items - Identifies the items and features to be tested by this test case.
- Input specifications - Specifies each input required by this test case.
- Output specifications - Specifies each output expected after executing this test case.
- Environmental needs - Any special hardware, software, facilities, etc. required for the execution of this test case that were not listed in its associated test design specification.

- Special procedural requirements - Defines any special setup, execution, or cleanup procedures unique to this test case.
- Intercase dependencies - Lists any test cases that must be executed prior to this test case.

Цель спецификации тест-кейсов — описать в деталях каждый тест-кейс. Она состоит из следующих секций:

- Идентификатор тест-кейса — уникальный идентификатор, благодаря которому данный документ может быть отличен от остальных.
- Элементы теста — идентифицирует элементы и фичи, которые необходимо протестировать по этому кейсу.
- Входные данные — описывает каждое входное значение, необходимое для выполнения тест-кейса.
- Выходные данные — описывает каждый результат, ожидаемый после выполнения тест-кейса.
- Окружение — любое специальное аппаратное, программное обеспечение, аппаратура и т.д., необходимое для выполнения тест-кейса и не перечисленное в документации по тест-дизайну (верхнеуровневая документация).
- Особые процедурные требования — описывает любые специальные действия по подготовке к тесту, его выполнению или очистке системы после выполнения кейса.
- Зависимости — список любых тест-кейсов, которые должны быть выполнены перед данным кейсом.