

## Лучшие практики в написании тест-кейсов.

Под best practices подразумеваются правила, которые помогают создавать простые, понятные и полезные тест-кейсы:

- Перед созданием нового тест-кейса убедитесь, что он не дублирует ни один из уже существующих в системе.
- Убедитесь, что тест-кейс покрывает 100% требований, которые вы должны проверить.
- Помните о теории методов тестирования, таких как анализ граничных значений, разделение эквивалентности, техника перехода состояния, угадывание ошибок.
- Помимо требований к системе, всегда помните о конечном пользователе, который будет взаимодействовать с системой.
- Не забудьте указать учетные данные, если они необходимы для выполнения теста.
- Позаботьтесь о тестировщиках, которые будут работать с этим тест-кейсом в будущем. В частности, убедитесь, что все ссылки верны и кликабельны. Пожалуйста, не используйте ссылки на продакшен.
- Используйте повелительное наклонение, например: «*Перейдите на главную страницу*», «*Введите данные*», «*Кликните*» и т. д. Такая формулировка упрощает понимание этапов тестирования и ускоряет их выполнение.

Чтобы проверить, насколько хорош ваш тест-кейс, покажите его человеку, который ничего не знает о проекте, над которым вы работаете. Вопросы, которые вы услышите, помогут определить слабые места вашего тест-кейса. Внесите изменения, иначе в будущем для поддержки тест-кейсов потребуется гораздо больше времени и усилий.

Хорошей практикой разработки тестовой документации будет предоставление всем участникам команды тестирования шаблона документа. При подготовке шаблона помните о следующем:

- зафиксируйте назначение документа (можно указать название, краткое описание, целевую аудиторию и т.д.);
- зафиксируйте все необходимые обозначения в легенде документа (например, статусы выполнения теста, названия модулей и т.д.);
- зафиксируйте ссылки на спецификацию функциональности, для которой разрабатывается тестовый документ;
- определите обязательные и необязательные атрибуты;
- определите причины заполнения необязательных атрибутов;
- определите список статусов прохождения тестов;
- при необходимости определите список приоритетов, а также список возможных значений других атрибутов.

Шаблон должен разрабатываться с учетом специфики каждого конкретного проекта, а также может быть расширен, сокращен или модифицирован для определенной функциональности, для тестирования которой, лучше использовать другие атрибуты.

Когда команда тестирования приступает к разработке тестовой документации следует обращать внимание на следующие аспекты:

- грамотность формулировок: названия элементов пользовательского интерфейса, действий системы и пользователя должны соответствовать названиям в спецификации; необходимо следить за краткостью, однозначностью, точностью формулировок;
- декомпозиция тестов: тесты должны быть структурированными и последовательными;
- простота восприятия и возможность быстро найти нужный тест: следует использовать инструменты для простой визуализации, разбиение на компоненты и действия, четкую структуру, фильтрацию, поиск, группировки, форматирование и т.д.;
- подготовка тестовых данных;
- избегание копипаста: путем осмысления и переработки информации из спецификации избегаем копирования требований в тестовые документы, а копирования одинаковых тестов, используемых в разных модулях системы, избегаем путем вынесения повторяющихся тестов в отдельные группы;
- соблюдение структуры документа: все необходимые атрибуты должны присутствовать и быть заполнены;
- поддержание тестовой документации в актуальном состоянии.

Чтобы команда тестирования следовала лучшим практикам, она как минимум должна о них знать. Можно составить документ, описывающий требования к тестовой документации, разместить его в общем доступе и по приходу новичка в команду, знакомить его с общепринятыми в компании практиками. Так появится возможность создать единообразный стиль ведения тестовой документации и сделать команду взаимозаменяемой.

## Требования к оформлению тестов.

### Формирование тест-кейсов

Обычно при написании тест-кейсов тестировщики пользуются таблицами Excel. Также можно использовать инструменты управления тестированием, такие как TestRail.

Общие требования к тест-кейсам:

1. язык описания тест-кейсов должен быть понятен широкому кругу пользователей, а не узкой группе лиц;
2. тест-кейс должен быть максимально независим от других тест-кейсов и не ссылаться на другие тест-кейсы (лучшая практика, когда зависимостей нет вообще);
3. тест-кейсы группируются в функциональные блоки по их назначению;
4. в тест-кейсах проверяющих работу функционала скриншотов быть не должно, иначе вы будете посвящать сотни часов на изменение всех скриншотов в тысячах тест-кейсах при изменении интерфейса тестируемой программы. Скриншоты могут быть добавлены только в тест-кейсы проверяющие отображение страниц и форм.

Если придерживаться данных правил, то тест-кейсы будут легко поддерживаемыми, легко читаемыми, не будут вызывать отторжения и могут быть использованы всеми участниками команды в процессе разработки программного обеспечения.

### Правила написания тест-кейсов:

1. Заголовок:

- должен быть чётким, кратким, понятным и однозначно характеризующим суть тест-кейса;
- не может содержать выполняемые шаги и ожидаемый результат.

## 2. Предусловие:

- может содержать полную информацию о состоянии системы или объекта, необходимом для начала выполнения шагов тест-кейса;
- может содержать ссылки на информационные источники, которые необходимо изучить перед прохождением тест-кейса (инструкции, описание систем...);
- не может содержать ссылки на тестируемый ресурс, если у информационной системы более одной среды (прод, тест, препрод...), данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
- не может содержать данные для авторизации, данная информация должна быть вынесена в инструкцию, и ссылка приложена в предусловии;
- не может содержать выполняемые шаги и ожидаемый результат, если нам нужно, чтобы до выполнения шагов проверки у нас была открыта главная страница, то мы в предусловии указываем «открыта главная страница сайта»;
- не может содержать ожидаемый результат.

## 3. Шаги проверки:

- должны быть чёткими, понятными и последовательными;
- следует избегать излишней детализации шагов. Правильно: «ввести в поле число 12».  
Неправильно: «нажать на клавиатуре на цифру '1', следующим шагом нажать на клавиатуре на цифру '2'»;
- должны использоваться безличные глаголы.  
Правильно: нажать, ввести, перейти.  
Неправильно: нажмите, введите, идите;
- не должно быть комментариев и пояснений, если есть необходимость привести мини-инструкцию, то оформляем инструкции в базе-знаний и в предусловии ссылаемся на неё;
- не должно быть жёстко прописанных статических данных (логины, пароли, имена файлов) и примеров, для исключения эффекта пестицида.

## 4. Ожидаемый результат:

- должен быть у каждого шага проверки;
- должно быть кратко и понятно описано состояние системы или объекта, наступающее после выполнения соответствующего шага;
- не должно быть избыточного описания.

## 5. Общие требования к тест-кейсам:

- язык описания тест-кейсов должен быть понятен широкому кругу пользователей, а не узкой группе лиц;
- тест-кейс должен быть максимально независим от других тест-кейсов и не ссылаться на другие тест-кейсы (лучшая практика, когда зависимостей нет вообще);
- тест-кейсы группируются в функциональные блоки по их назначению;
- в тест-кейсах проверяющих работу функционала скриншотов быть не должно, иначе вы будете посвящать сотни часов на изменение всех скриншотов в тысячах тест-кейсах при изменении интерфейса тестируемой программы. Скриншоты могут быть добавлены только в тест-кейсы проверяющие отображение страниц и форм.

## Основные атрибуты тест-кейсов.

Как и все тестировочные документы, тест-кейс имеет определенный формат. Он содержит следующие атрибуты:

- **ID** — уникальное сочетание букв и цифр. Номер тест-кейса — уникальный идентификатор тест-кейса (такие системы как TestRail, TestLink и подобные)

автоматически присваивают тест-кейсам уникальные номера). Если у вас тысячи тест-кейсов, то при общении с коллегами, вам будет удобнее сообщить номер тест-кейса ссылаясь на него, а не пытаться словами рассказать, где и как найти определённый тест-кейс.

1. **Заголовок** — основная идея тест-кейса, краткое и ёмкое описание его сути. Например, заголовок тест-кейса для ручного тестирования страницы входа может выглядеть следующим образом: «*Проверить вход пользователя с корректными данными*».
2. **Предусловия** — список действий, которые необходимо выполнить или учесть перед выполнением тест-кейса. При необходимости здесь могут указываться учетные данные.
3. **Шаги** — описание последовательности действий, необходимых для проверки.
4. **Постусловия** — список действий, возвращающих систему в исходное состояние (указывается при необходимости).
5. **Ожидаемый результат** — то, что мы ожидаем получить после выполнения определённых действий в соответствующем шаге.
6. **Фактический результат** — то, что мы получаем после выполнения тест-кейса (указывается при необходимости).
7. **Статус** — Success (успех), Failed (провал), Blocked (блокировка) (указывается при необходимости).

Кроме того, для некоторых тест-кейсов могут потребоваться дополнительные атрибуты:

- **Требования к среде** — специальное оборудование, программное обеспечение и т. п. вещи, необходимые для выполнения тест-кейса и не перечисленные в соответствующей спецификации проекта тестирования.
- **Специальные процедурные требования** — особые процедуры настройки, выполнения или очистки, уникальные для этого тест-кейса.
- **Межкейсовые зависимости** — тест-кейсы, которые нужно выполнить перед этим тест-кейсом.

### Характеристики хорошего тест-кейса

Прежде всего, тест-кейс не должен быть зависимым или связанным с другими тест-кейсами. Он должен быть полным и самодостаточным. Следует избегать расплывчатых описаний шагов или ожидаемых результатов. Любые ограничения, отсутствие необходимой информации или чрезмерное количество деталей делают тест-кейсы менее эффективными.

Короче говоря, хороший тест-кейс:

- понятен любому члену команды;
- аккуратно и точно написан;
- соответствует требованиям;
- воспроизводим;
- пригоден для многократного использования.

### Чего не должно быть в тест-кейсе

1. Зависимостей от других тест-кейсов;
2. Нечеткой формулировки шагов или ожидаемого результата;
3. Отсутствия необходимой для прохождения тест-кейса информации;
4. Излишней детализации.

Обязательные атрибуты, требования тест-кейса:

- **ID (Идентификатор тест-кейса)** – уникальный номер документа, присваивается зачастую автоматически.
- **IDEA(Идея)** – Какую именно идею мы хотим проверить. Зачастую информация из поля "Идея" переносится в Title.
- **Title(Наименование тест-кейса)** – краткое описание проверки.
- **Steps (Шаги выполнения тест-кейса)** – сценарий(шаги) для выполнения тестового сценария.
- **Expected result (Ожидаемый результат)** – результат, который должен получиться по итогу выполнения сценария. Описывается для каждого шага в кейсе.
- **Actual result (Фактический результат)** – добавляется напротив ожидаемого результата каждого шага. В случае успешного прохождения шага тест-кейса, ставится отметка «Passed». Некоторые проекты игнорируют данный атрибут, добавляя фактический результат только в случае проваленного теста.
- **Comment (Комментарий)** – дополнительная информация тестировщика после выполнения тест-кейса.
- **Status (Статус)** – результат (Статус) прохождения тест-кейса. Добавляется тестировщиком после прохождения тест-кейса.

#### Необязательные атрибуты, требования тест-кейса:

- **DATA (Тестовые данные)** – значения, данные, табличные данные, необходимые для проверки. Могут быть в виде таблицы, либо вынесены в отдельный файл.
- **Priority (Приоритет тест-кейса)** – приоритет выполнения, очередность.
- **Enviroment (Окружение)** - тест-кейс является специфичным для проверяемого окружения.
- **Type (Тип тест-кейса)** – ставится отметка «Позитивный» или «Негативный» сценарий. Помогает распределить, отсортировать и понять тестовые сценарии, написанные согласно требованиям.
- **Pre-condition (Предусловия)** – необходимые действия, которые надо выполнить перед выполнением тест-кейса. Например, регистрация, авторизация, скачивание файла и так далее.
- **Post-condition(Постусловия)** – действия, которые необходимо сделать по окончании выполнения тест-кейса. Например, очистить данные, удалить записи, файлы и так далее.