

УДК 81'25

Трудности и пути решения проблем при написании тестовых сценариев

Е.В. Алференко✉, Л.В. Дудина

Воронежский институт высоких технологий, Воронеж, Россия

В данной статье описываются основные ошибки, с которыми может столкнуться тестировщик при составлении тестовых сценариев. К рассмотрению представлены варианты корректного написания тест-кейсов, а также даны рекомендации по предотвращению ошибок при создании тестов. Статья будет полезна тем, кто не только начинает свой путь в ИТ, но и тем, кто заинтересован в качественной разработке и поставке программного обеспечения.

Ключевые слова: программное обеспечение, тестирование, тестовый сценарий, тест-кейс, пользователь, безопасность.

Difficulties and ways to solve problems by writing test cases

E.V. Alferenko✉, L.V. Dudina

Voronezh Institute of High Technologies, Voronezh, Russia

This article describes the main errors that a tester may encounter when compiling test scenarios. Options for the correct writing of test cases are presented for consideration, as well as tips for preventing errors when creating tests. The article will be useful for those who are not only starting their way into IT, but also for those who are interested in high-quality software development and delivery.

Keywords: software, testing, test scenario, test case, user, software security.

Системы с программным обеспечением являются неотъемлемой частью нашей жизни, от бизнес-приложений (таких как банковское программное обеспечение) до потребительских товаров (таких как автомобили). Многие люди имели опыт использования программного обеспечения, которое не работало так, как ожидалось. Программное обеспечение (ПО), которое не работает корректно, может привести ко многим проблемам, включая потерю денег, времени или деловой репутации, и стать причиной травмы или смерти. Тестирование программного обеспечения – это способ оценить качество программного обеспечения и снизить риск его отказа.

Тщательное тестирование компонентов и систем, а также связанной с ними документации может помочь снизить риск сбоев и отказов во время работы ПО. Когда дефекты обнаружены и исправлены, это способствует качеству компонентов или систем.

Тестирование – это не только процесс прогона тестов. Важным аспектом в тестировании является проектирование тестов. Во время проектирования тестов тестовые условия воплощаются в высокоуровневые тестовые сценарии, наборы высокоуровневых тестовых сценариев и другое тестовое обеспечение. Так, анализ тестирования отвечает на вопрос «что тестировать?», а проектирование тестов отвечает на вопрос «как тестировать?».

Предметом исследования послужили трудности и пути решения проблем при написании тестовых примеров при тестировании программного обеспечения.

В своей работе тестировщики (или инженеры по обеспечению качества) полагаются на инструкции, чтобы установить, что должно быть проверено в соответствии с критериями приемки. Одна из этих инструкций – это тестовый сценарий, и инженеры по тестированию пишут его сами [1].

Не существует уникальной подборки тестовых сценариев. Тем не менее, существует простой в использовании набор методов и решений, которые при правильном применении дают хорошие результаты.

Следует разграничить понятия «тестовый сценарий» и «тест-кейс».

В сценарии тестирования подробно описывается последовательность действий, которые тестировщик выполняет для проверки целостности программного обеспечения. Она состоит из нескольких пунктов, в которых описывается, что именно делать с разработанной программой и каким должен быть результат.

Следуя тестовому сценарию, специалист последовательно выполняет пункты и смотрит, соответствует ли фактический результат ожидаемому. Если результат не может быть достигнут в тот или иной момент, о нем составляется отчет.

Обычно тестовый сценарий описывает проверку определенной функции: регистрация пользователя, заказ продукта, загрузка фотографии. Можно проверить работу нескольких функций в связке. Это позволяет проконтролировать функциональность и выявлять ошибки, которые мешают работе приложения в целом.

В тесте есть аналогичный инструмент – тест-кейс. Тест-кейс – это один из результатов тестирования программного обеспечения. Следуя инструкциям по тестированию, инженер по обеспечению качества может сравнить фактические и ожидаемые результаты тестирования, чтобы увидеть, работает ли программное обеспечение в соответствии с первоначальным замыслом. Отличие от тестового сценария состоит в том, что тест-кейс описывает не функцию, а отдельно детали, которые участвуют в этой функции.

Соотношение исследуемых понятий наглядно представлено на рисунке.

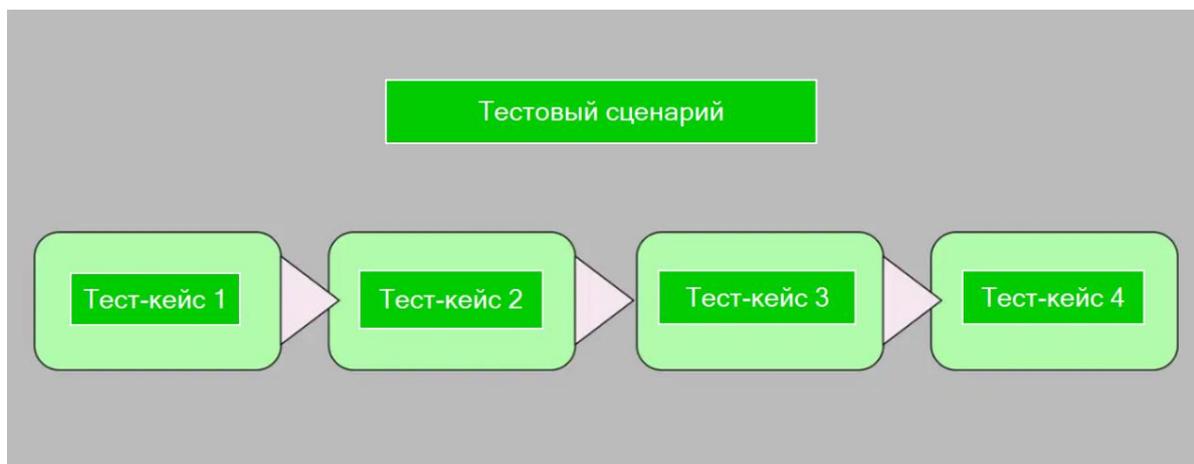


Рисунок. Схема соотношения тестовых сценариев и тест-кейсов

Изучение того, как писать тестовые примеры, требует большого терпения к деталям и хорошего понимания тестируемого приложения [2]. В качестве золотого правила при написании тестовых примеров необходимо придерживаться следующей структуры:

- идентификатор тестового примера;
- описание;

- предварительное условие;
- степень важности;
- тестовая среда или тестовое окружение (dev стенд, test стенд, стенд заказчика);
- тестовые шаги;
- тестовые данные;
- ожидаемые и фактические результаты.

Основная цель каждого тестового примера – предоставить ясное и краткое пошаговое руководство, чтобы получить четкое представление о том, что пошло не так.

Шаги, которые можно выполнить, чтобы предоставить хорошие образцы ручных тестовых примеров для страницы входа, представлены в таблице.

Таблица

Примеры тестовых шагов для написания тест-кейсов

	ШАГИ	ДЕТАЛЬНАЯ ИНФОРМАЦИЯ	ПРИМЕР
1	Идентификатор	Убедитесь, что вы присвоили каждому тестовому примеру уникальный идентификатор	TC_UI_1
2	Описание тестового примера	Правильно выбирайте тестовые примеры из тестовых сценариев	Тестовый сценарий: подтвердите вход в Gmail Тест-кейс: введите действительное имя пользователя и пароль
3	Предусловие	Условия, которые должны быть выполнены до запуска тестового примера. Упомянуть только в том случае, если есть предварительные условия	У пользователя есть действующая учетная запись Gmail
4	Степень важности	Степень влияния, которое дефект (или ошибка) может оказать на работу программного обеспечения. Уровни: Тривиальный, Незначительный, Существенный, Критический, Блокирующий	Критический
5	Тестовое окружение	Настройка программного и аппаратного обеспечения для выполнения тестовых примеров	Phone 8 Plus + iOS 14.4
6	Тестовые шаги	Опишите свой путь к ожидаемому результату	1. Введите правильное имя пользователя. 2. Введите правильный пароль для конкретного пользователя. 3. Нажмите на кнопку Войти.
7	Тестовые данные	Идентификация данных для тестирования может занять много времени и иногда может потребовать повторного создания тестовых данных	Имя пользователя: <u>test_user@qawerk.com</u> Пароль: test

Продолжение таблицы

	ШАГИ	ДЕТАЛЬНАЯ ИНФОРМАЦИЯ	ПРИМЕР
8	Ожидаемый результат	Результат, который ожидают тестировщики при запуске тестового примера.	Успешный вход в систему.
9	Фактический результат	Результат, полученный тестировщиками при запуске тестового примера	Появилось сообщение об ошибке. Не удалось войти в систему.
10	Статус	Указывается при сравнении ожидаемых и фактических результатов теста (пройдено /не пройдено)	Результат: не пройдено

Наиболее распространенные ошибки, которые допускают тестировщики при создании тестовых примеров, сводятся к трем основным типам [3].

1. Максимальное упрощение тестовых примеров.

Иногда тестировщик настолько сильно увлекается сокращением излагаемой информации, что тесты не могут быть выполнены в соответствии с каким-либо алгоритмом. Вот наглядный пример такой работы.

Тестовые шаги:

1. Выберите пользователя в таблице, чтобы отобразить контекстное меню со следующими параметрами: редактировать, создать, удалить и копировать;

2. Необходимо проверить, все ли доступные опции работают правильно.

Довести это до автоматизма будет крайне сложно. Во-первых, когда тестировщик нажимает кнопку «Удалить», все контекстное меню закрывается. Затем его необходимо снова открыть. Подобные сценарии нигде не описаны.

Во-вторых, пользователь будет удален. Для тестирования других функций требуется учетная запись другого пользователя. И опять же, подобные вещи нигде не указаны в спецификации.

2. Переадресация на другой контент за один шаг.

Например, на шаге 10 в принципе не рекомендуется создавать следующий тестовый шаг, такой как «Повторить шаги 8 и 9 для пользователя группы Admin».

Либо эта проверка вообще не должна существовать, либо ее необходимо выполнить отдельным шагом. Такие рекомендации могут значительно увеличить время тестирования.

3. Ветвление в тестовом примере.

Например:

1. Откройте страницу пользователя и проверьте, отображается ли таблица с выбранными столбцами «Адрес электронной почты», «Имя» и «Тип»;

2. Если пользователей нет, определите, что таблица существует.

Такие действия нельзя выполнять в одном кейсе. Тест должен содержать только один сценарий: либо проверку отображения таблицы, либо наличие указанных столбцов в таблице, либо наличие пользователей.

Основные ошибки при создании тестовых сценариев выглядят следующим образом:

- сценарий слишком длинный;
- неполное, неправильное или непоследовательное описание условий тестирования или подготовки тестовой среды;
- пропущены «очевидные» шаги;

- использование устаревшей информации о тестируемой системе;
- не очевидно, кто должен выполнить действие: пользователь или система;
- не ясно, каков ожидаемый результат, в случае успешного прохождения теста.

Для избежания ошибок при написании тестовых примеров предлагаются следующие рекомендации:

1. Всегда нужно записывать тестовые сценарии и тест-кейсы в системе для документирования тестов (например, TestRail);

2. Тестовые примеры должны быть подготовлены в очень подробной форме;

3. Необходимо избегать повторных тестов и тестовых шагов;

4. Необходимо описать все возможные данные для проверки;

5. Тестовые примеры должны быть распределены таким образом, чтобы автоматизация начиналась с наиболее важных проверок;

6. Нет необходимости объединять много тестов (особенно сложных) в одном тесте. Лучше написать три тест-кейса с меньшим количеством шагов, нежели один тест-кейс с множеством шагов.

Написание тестовых примеров требует практики, знания тестируемого программного обеспечения и знакомства с передовыми методами создания тестовых примеров. Желательно сначала изучить примеры проверенных тестовых компаний и использовать шаблоны, которые они предоставляют, такие как шаблоны тестовых сценариев, а также шаблоны для конкретных методов, такие как шаблоны модульных тестов.

Хорошо структурированные и профессионально написанные тестовые примеры могут сделать процесс тестирования более плавным и сэкономить время и усилия всех заинтересованных сторон. Чтобы написать хорошие тестовые сценарии, нужно следовать проверенным рекомендациям:

Каким должен быть тест-кейс:

- понятным для всех: шаги в тесте легко воспроизвести и повторить;
- можно использовать много раз в тестировании, легко адаптировать под новые изменения;

– тест-кейсы с большим количеством шагов рекомендуется разбивать на несколько тест-кейсов;

– обеспечение 100% тестового покрытия.

Каким не должен быть тест-кейс:

- иметь расплывчатое описание;
- не должно быть описано более одной тестовой цели;
- нельзя игнорировать при написании тестов оценки экспертов и бизнес-аналитиков;

– нельзя предполагать ожидаемые результаты или придумывать их, они должны быть четко прописаны [4].

Следует ориентироваться на **примеры конкретных требований и тест-кейсы для тестирования:**

Описание проблемы:

В документах на английском языке неправильно написано слово «months». Слово «month» написано по-немецки «Monaten».

Предложение с ошибкой находится на третьей странице в документе. Документы можно найти в трех разделах: поминутная оплата, паушальная оплата и фиксированная цена.

Вот текст с ошибкой:

The total of all charges in this Proposal, subject to changes in the service during the term of the Agreement, for the contractual term of Monaten for.

Требования:

Ошибка исправлена во всех трех разделах в документах на английском языке (поминутная оплата, паушальная оплата и фиксированная цена).

Тест-кейс1:

Описание:

Убедиться, что в предложении (на английском языке) вместо «Monaten» указано «months» в разделе «Модель с поминутной оплатой».

Предварительное условие:

Модель с поминутной оплатой уже существует в системе. Пользователь успешно прошел авторизацию.

Шаги:

- нажать кнопку «Просмотреть документы»;
- нажать «Документы (EN)»;
- скачать файл документа на английском языке.

Ожидаемый результат:

В файле экспорта «Документ (EN)» слово «month» написано правильно.

Фактический результат:

В файле экспорта «Документ (EN)» слово «month» написано правильно.

Проведенное исследование позволяет сделать вывод о том, что тестирование играет важную роль в разработке программного обеспечения и является неотъемлемой частью жизненного цикла его разработки, так как:

- повышает надежность, качество и производительность программного обеспечения;
- помогает разработчику проверить, правильно ли работает программное обеспечение, убедиться, что программное обеспечение выполняет то, для чего оно предназначено;
- помогает понять разницу между фактическим и ожидаемым результатом, что обеспечивает качество конечного продукта.

Благодаря тому, что тестирование ПО становится частью программирования, разработчики имеют возможность исправлять ошибки уже на начальной стадии разработки. Это позволяет сократить риск появления дефектов в готовом продукте. Если ошибки найдены на начальном уровне, разработчик может создать надежное программное обеспечение. Таким образом, чем раньше начинается процесс, тем раньше обнаруживаются ошибки и тем ниже стоимость их исправления.

Без надлежащей оценки производимое программное обеспечение может нанести существенный вред как пользователям, так и производителю. Потеря денег и доверия к компании – это лишь небольшая часть последствий. Хорошее и эффективное тестирование программ помогает повысить их безопасность. Следование инструкциям при написании тестовых сценариев, внимательность при тестировании ПО помогают предотвратить большую часть ошибок.

СПИСОК ИСТОЧНИКОВ

1. ISTQB Foundation Level Syllabus (Сертифицированный тестировщик Программа обучения Базового уровня) [Электронный ресурс] – URL: https://www.gasq.org/files/content/gasq/downloads/certification/ISTQB/Foundation%20Level/ISTQB_CTFL_Syllabus_2018-RU.pdf (дата обращения: 22.12.2023).
2. Блэк Рекс. Ключевые процессы тестирования. Планирование, подготовка, проведение, совершенствование / Рекс Блэк. – М.: Лори, 2006. – 576 с.
3. Распространенные ошибки начинающих тестировщиков [Электронный ресурс] – URL: https://ibs-training.ru/about/news/Rasprostranennye_oshibki_nachinayushchikh_testirovshchikov (дата обращения: 22.12.2023).
4. Тест-кейс и чек-лист в тестировании: ошибки, отличия | Test Planet [Электронный ресурс] – URL: <https://test-planet.by/news/tipichnye-oshibki-pri-sostavlenii-chek-listov-i-test-kejsov> (дата обращения: 22.12.2023).

ИНФОРМАЦИЯ ОБ АВТОРАХ

Алференко Елена Вячеславовна, кандидат филологических наук, доцент, Воронежский институт высоких технологий, Воронеж, Россия.

e-mail: Elena_alferenko@mail.ru

Дудина Лилия Владимировна, студент, Воронежский институт высоких технологий, Воронеж, Россия.

e-mail: dudina_lili@bk.ru