

Practice 5

checklist, test case. Test
design techniques

Agenda

- Test design techniques
- Основные тестовые артефакты: тест кейс, чек-лист

STLC. Software testing life cycle

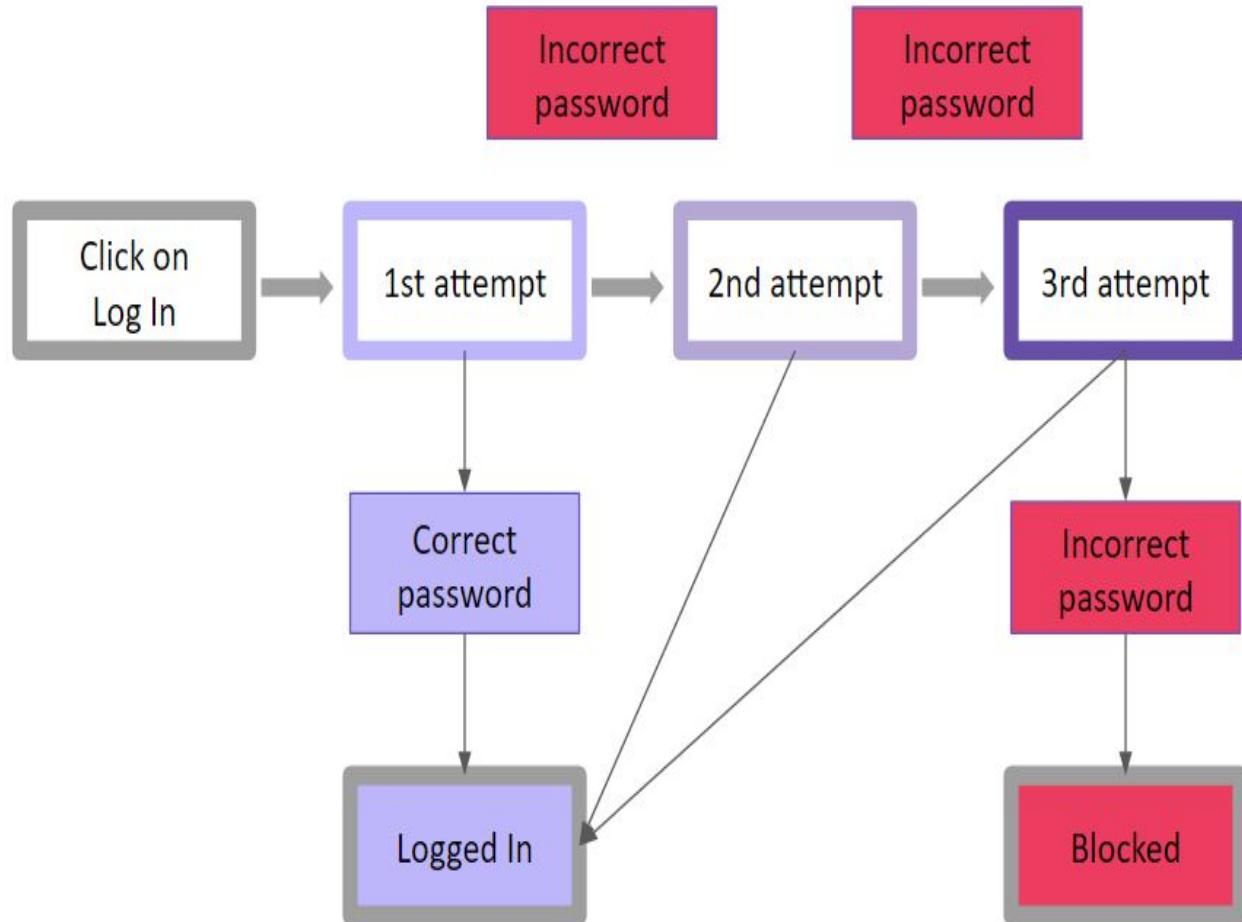
STLC (Software testing life cycle) means a software testing process divided into specific stages that are performed sequentially in order to improve the quality of the system/product under test.

Main phases:

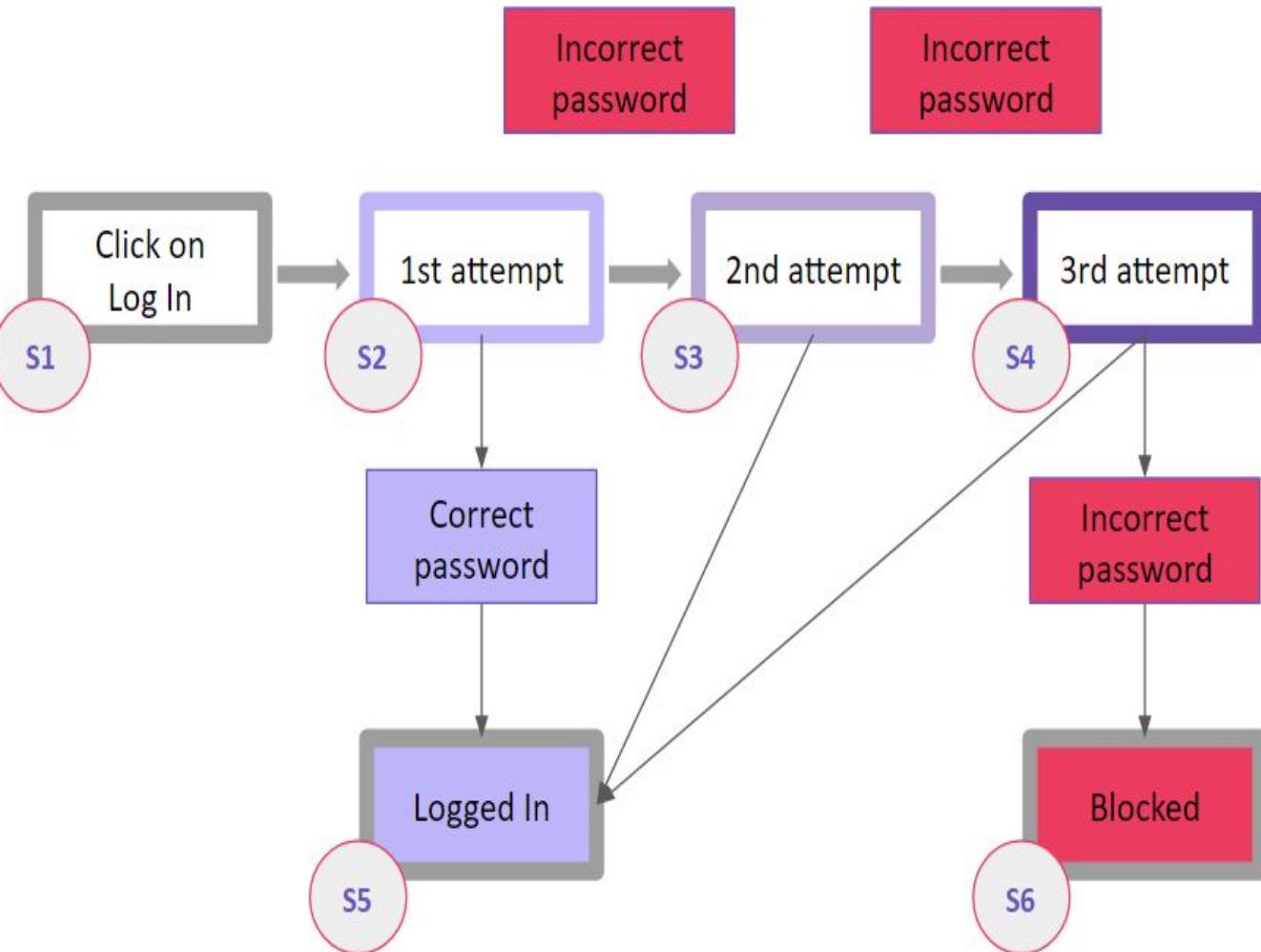
- Requirements analysis
- Planning
- Development of test cases
- Deploying a test environment
- Testing
- End of life cycle testing (analysis and reporting)

Test design techniques

- Equivalent Class Partitioning
- Boundary Value Analysis
- State Transition
- Pairwise Testing
- Error Guessing (e.g., majority of testing challenges with input fields)
- Exploratory testing



Самый простой пример state transition — визуализация входа в учетную запись во время тестирования веб-сайта или мобильного приложения. Допустим, мы тестируем систему, которая предлагает ограниченное количество попыток ввода правильного пароля. Если пользователь не вводит правильный пароль, система блокирует доступ (временно или навсегда, сейчас не имеет значения).



Блоки разного цвета обозначают конкретные состояния системы. Добавим метки, обозначающие состояния, и получим следующее (см картинку)

		Correct password	Incorrect password
State 1	Click on Log In	State 5	State 2
State 2	1st attempt	State 5	State 3
State 3	2nd attempt	State 5	State 4
State 4	3rd attempt	State 5	State 6
State 5	Logged In		
State 6	Blocked		

Подобная диаграмма упрощает сопоставление возможных входных данных с ожидаемыми результатами. Наличие визуализации прямо перед глазами помогает сохранять ясную голову и правильно связывать состояния. Позже вы сможете лаконично и удобно систематизировать данные — например, в таблице, по которой можно будет смотреть во время тестирования.

Можно использовать сразу несколько техник, например в дополнении к state transition.

разберем состояние, когда пользователь вводит корректный пароль.

Представим, что требования к паролю следующие:

пароль: min 5 chars max 25 chars, Alphanumeric, English alphabet

Какие здесь техники мы можем использовать?

Equivalent Class Partitioning: range до 5, 5-25, больше 25 chars

valid: 10 chars

invalid: 3, 28 chars

Boundary Value Analysis

valid: 5,6 chars 24,25 chars

invalid: 4, 26 chars

Alphanumeric:

valid: abcd1234yt

invalid: ad12, 12345

English alphabet:

invalid: check Cyrilic with valid length

Let's find some improvements for a test case (manual testing), fix mistakes

TestCaselId	Description/Test Summary	Preconditions	Test Steps	Expected Result
TC1	User logs in the application with the correct password	1.Prepare testing data for correct password: Alphanumeric, English,	1. Open the app via link.... 2. Go to log in page 3. Enter admin login 4. Enter the correct password with valid length 5.Click on Submit button	1.The app is opened 2. Log in page is opened 3.Admin login is entered 4.Password is entered 5.Log in page is closed. 6.User is transferred to admin panel.

TestCaselid	Description/Test Summary	Preconditions	Test Steps	Expected Result
TC1	<p>Admin User logs in the application with the valid (correct) password</p>	<p>1.Prepare testing data for correct password: Alphanumeric, English,</p> <p>Valid length 5-25</p> <p>2. The app is launched</p> <p>3.(optional)Test env -description (browser, version)</p>	<p>1. Open the app via link....</p> <p>2. Go to log in page</p> <p>3. Enter admin login</p> <p>4. Enter the (correct) password (with valid length)</p> <p>5.Click on Submit button</p>	<p>1.The app is opened</p> <p>2. Log in page is opened</p> <p>3.Admin login is entered</p> <p>4.Password is entered</p> <p>5.Log in page is closed. User is transferred to admin panel.</p>

Разработка тест кейсов



Действия на этапе разработки тест кейсов:

- Создание тест кейсов, чек-листов
- Создание тестовых сценариев при необходимости
- Проверка тест-кейсов и скриптов автоматизации (если на проекте есть автоматизация)
- Создание тестовых данных в тестовой среде

Тест кейс — это набор действий, выполняемых для проверки конкретной функциональности программного приложения. Тест кейс содержит шаги тестирования, тестовые данные, предварительные условия, разработанные для конкретного тестового сценария для проверки любого требования. Тест кейс включает в себя определенные переменные или условия, с помощью которых инженер по тестированию может сравнить ожидаемые и фактические результаты, чтобы определить, работает ли программный продукт в соответствии с требованиями заказчика.

Основные атрибуты тест кейса:

1. Тест кейс Id — уникальный идентификатор тест кейса. Это обязательное поле, которое однозначно идентифицирует тест кейс, например. TC_01.
2. Описание тест кейса (заголовок) - Это поле определяет цель тест кейса.
3. Предварительные условия— набор предварительных условий, которые необходимо выполнить перед выполнением шагов теста.

Например, при тестировании функциональности приложения после входа в систему у нас может быть обязательное поле «Пользователь должен быть зарегистрирован в приложении».

4. Шаги теста — подробные шаги для выполнения тест кейса. Это самое важное поле тестового кейса. Тестирующий должен стремиться иметь четкие и недвусмысленные шаги в поле шагов теста, чтобы любой другой специалист мог повторить шаги, не испытывая затруднений либо недопониманий.

5. Ожидаемый результат — ожидаемый результат для прохождения теста. На основе выполненных шагов тестирования и использованных тестовых данных мы приходим к ожидаемому результату, например. пользователь должен успешно войти в систему и перейти на домашнюю страницу.

6. Результат теста — состояние выполнения теста «пройдено/не пройдено» (обязательное поле)

Могут присутствовать такие поля как: модуль (какая функция тестируется), приоритетность тест кейса

Свойства качественных тест кейсов

- Простой и конкретный (любой из команды тестировщиков должен быть в состоянии выполнить тест-кейсы без помощи автора);
- Не содержит ненужных шагов либо предварительных условий;
- Предварительные условия описаны понятно;
- Определен набор тестовых данных (для автоматизации либо исходя из требований проекта);
- Ясный, краткий и полный (без предположений в описании тест кейса, шагах, ожидаемом результате и т. д.);
- Написан без грамматических ошибок;
- Написан без двусмысленных слов;
- Всегда перечисляются ожидаемые результаты для каждого шага;
- Не дублирует другие тест кейсы и не ссылается на них (ег, ожидаемый результат как в тест кейсе ТС_23);
- Можно использовать много раз для тестирования (если не было изменений в требованиях);
- ВАЖНО! тесты кейсы для негативных проверок никогда не объединяем

TestCaseId	Component (optional)	Priority (optional)	Description/Test Summary	Preconditions	Test Steps	Expected Result	Status (optional)
GoogleSearch_1	Search_Bar_Module	P0	<p>1) Verify search results are displayed after entering search term and pressing enter</p> <p>2) Search results are displayed after entering search terms and pressing enter</p>	<p>1.Chrome browser is launched</p> <p>2.Chrome version is</p> <p>3.</p>	<p>1. Open in the browser's URL bar</p> <p>http://google.com and press enter.</p> <p>2. Input the search term - "Apple" in the google search bar.</p> <p>2. Input any search term in the google search bar.</p> <p>3. Press enter button</p>	<p>1.The start page is opened</p> <p>2. Input term "Apple" appears</p> <p>2.Entered term appears in google search bar</p> <p>2. Search results related to 'apple' should be displayed</p> <p>3.Search results related to entered term is displayed</p>	Pass

Test suite (набор тестов)

Test suite — это набор или контейнер множества тест кейсов, связанных друг с другом. Группировка тестовых наборов в комплекты с соответствующими именами упрощает управление ими. Можно группировать тесты в наборы по нескольким критериям:

- по типу теста (например, интеграционные тесты или модульные тесты)
- по времени выполнения (например, медленные, средние и быстрые тесты)
- наконец, по модулям или областям в приложении

То, как набор тестов будет выглядеть на практике, будет зависеть от типа тестирования. Если мы говорим о ручном тестировании, набор тестов может быть таким же простым, как папка, содержащая текстовые документы, или это может быть более сложный подход (с использованием программных средств).

Checklist (чек лист)

В тестировании чек-лист — это список проверок для тестирования продукта. Чек-листы очень просты. Любой из них содержит список блоков, разделов, страниц и других элементов, функций, которые необходимо протестировать.

Для создания рабочего чек-листа тестировщику следует обратить внимание на следующие рекомендации:

- Одна единица = одна проверка. Минимальная полная операция, выполняемая тестировщиком при проверке, составляет один пункт чек-листа.
- Тестировщик должен опираться на требования при составлении чек-листа, чтобы не тестировать то, что не является существенным.
- Все члены команды должны использовать названия пунктов чек-листа в единой форме, чтобы работа с чек-листом не вызывала неоднозначных толкований. Например, инфинитивные глаголы или существительные во всех абзацах: «проверить»/«добавить»/«отправить»..
- Детализировать чек-лист в зависимости от задачи (добавить ожидаемые результаты, тестовые данные).
- Удобно объединять чек-листы в матрицы, где тестировщики могут отражать не только сами проверки, но и условия проверки (платформа, версия продукта, сотрудник и т.д.) и статус проверки. Матрицы — это компромисс между чек-листами и тест-кейсами. Их легче поддерживать, чем тест-кейсы, так как в такой таблице нет шагов. У них одна строка = одна проверка

Site "checklist.example"	Tester 1	Tester 2	Tester 3
	Firefox Browser 70.0.1	Google Chrome 78.0.3904.108	Opera 65.0.3467.42
Registration			
E-mail	Passed	Passed	No run
Social Media	Failed	Passed	Failed
Field validation	Passed	Passed	Passed
Authentication			
User is activated	Passed	Passed	Failed
User is not activated	Passed	Skipped	Passed
Password reset	Passed	No run	Passed
Field validation	Passed	Failed	Skipped
Profile			
Login reset	Passed	Passed	Failed
E-mail reset	Failed	Skipped	Passed
E-mail reset	Failed	Passed	Skipped
Field validation	Passed	Skipped	

Check	Expected result
Verify that the "Hire now" button changes color while you hover over it	<p>1. The "Hire now" button should become black.</p> <p>2. Text should become red.</p>
Verify that the "Technologies" link is underlined by default.	Underlining should be indicated with a red line.

Преимущества:

- чек-лист легко читается;
- быстро протестировать чек-лист: в тест-кейсе нужно отметить статус каждого шага, а в чек-листе достаточно одной строчки;
- чек-лист является источником результатов для отчета: можно быстро подсчитать, сколько проверок было выполнено и в каком они состоянии, узнать количество открытых отчетов;
- в любой момент можно узнать статус — всегда есть что проверить в первую очередь, можно заказать позиции из чек-листа или изменить порядок, когда это необходимо.

Недостатки:

- неопределенность набора тестов: каждый тестировщик выполняет пункт чек-листа по-своему;
- недостаточная детализация;
- новичков сложнее обучать: пункты чек-листа чаще абстрагируются от конкретных элементов интерфейса и описывают, что нужно сделать;
- чек-лист менее эффективен для начинающих тестировщиков, лучше использовать тест-кейсы.



THE END