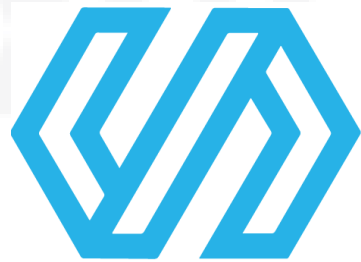


Test Cases and Test Suites



SKILLO

Agenda

1. Recap
2. What is a test case?
3. Why we need test cases?
4. Test case structure and attributes
5. Test cases execution
6. Test case types
7. Test cases maintainability

Agenda

- 8. Bad practices
- 9. Good practices
- 10. Test suites
- 11. Test suite attributes
- 12. Test suites by different criteria
- 13. Test case management
- 14. Exercises

Recap

1. What is VCS
2. Differences between CVCS and DVCS
3. GIT commands

What is a test case?

ISTQB definition: A set of input values, execution preconditions, expected results and execution postconditions, developed for a particular objective or test condition, such as to exercise a particular program path or to verify compliance with a specific requirement.

IEEE definition (IEEE 610): A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly.

What is a test case?

Test case definitions include the following (IEEE 610):

- pre-conditions
- set of input values
- set of expected results
- how to execute the test and check results
- expected post-conditions

Why we need test cases?

- Requirements traceability
- We know when testing is done
- Reusability
- We identify potential problems on design phase, before the actual implementation

Test case attributes

Test Case ID	<i>Unique id of a test case. For ex. - TC001</i>
Name	<i>Clear, short and descriptive. It should be possible to execute the test case only by name</i>
Priority	<i>How important is the test case. Usually priority is 1-4 (1 is the highest priority)</i>
Description	<i>More detailed explanation of the test case. It is not required if the name is enough and clear</i>
Prerequisites	<i>Precondition in the system/test environment</i>
Input Data	<i>User accounts, SQL query, some data etc.</i>
Date Created	<i>When the test case is created</i>
Author	<i>Who created the test case</i>
Test Steps	
Step	Expected Result
<i>Actions need to be executed in order to achieve the expected result</i>	<i>The result that needs to be verified once the test is executed. Often the expected result is a quote of a requirement.</i>

Test case example

Test Case ID	TC001
Name	Successful registration in abv.bg
Priority	1
Description	Fill all registration form fields and create new abv account
Prerequisites	Mobile number and username not taken and used in abv.bg before
Input Data	https://abv.bg
Date Created	18.05.2015
Author	Hyusein Tyurkmen
Test Steps	
Step	Expected Result
1. Open site url	User is redirected on the homepage. "Регистрирай се!" option is available
2. Click on "Регистрирай се!"	Registration form is opened
3. Fill correctly all form fields with the data	
4. Click "Създай АБВ профил" button	A message for succesfull registration is shown and the user is redirected to the abv inbox

Test case execution

- Test case execution is finished once we have compared the actual and expected result
- Possible statuses after execution
 - **PASS** – when Actual result is equal to Expected result
 - **FAIL** – when Actual result is not equal to the Expected result
 - **BLOCKED** – other bugs or environmental issues prevent us to execute the test case
 - **SKIPPED** – when the test case is not executed due to insufficient testing time

Test case execution

- Failed test cases need the following investigation and efforts

Main reasons of FAIL status :

- changes in the product not communicated with the QA
- wrongly defined expected results
- real bug in the software

Test case types

Categories by level of details criteria:

- **High Level**
 - No input data, steps and exact expected result are specified
 - Each QA can execute it differently and can find a new bugs
 - Easy to maintain
 - Does not guarantee the coverage of the functionalities
- **Low level**
 - Detailed steps and expected results with given input data
 - The execution is exactly the same every time
 - Hard to maintain
 - It is clear exactly how much functionality is covered

Test case types

Categories by Expected Result criteria:

- **Positive**
 - Verify the functions are working correctly (cover happy paths)
 - Cover main user actions
 - Ensure customers that the product is designed as expected
 - Should always be performed
- **Negative**
 - Verify system does not crash in case of invalid input data
 - Verify system in case of incorrect usage
 - Negative test cases should be written after the positive ones
 - Usually they are performed on mission critical systems

Test case types

Categories by area of testing criteria:

1. Functional test cases
2. Performance test cases
3. Security test cases
4. Integration test cases
5. Database test cases
6. Acceptance test cases
7. Usability test cases

Test cases maintainability

Maintainability — simplicity and ease of changing a test case to reflect the changes in the software

Good practices for easy maintainability

- Document with common steps linked to the test cases – great benefit for large test case suites and big projects. Usually stored in Knowledge base
- Test case steps - level of details
- Do not add obvious steps

Bad practices

1. Dependency between the test cases

- depends on the results of the execution of other test cases
- refer to the steps of other test cases

2. Poor description of the test case steps

- Not clear and concrete
- The test case can be executed only by its author

3. Poor description of the idea and/or Expected Results

- references to external document in the expected results or in the descriptions
- not precise and clear expected results

Good practices

1. Independency of the test cases
2. Clear and precise description of the test case steps
3. Clear idea and/or expected results
4. Write test cases with the idea that everybody should use and understand them
5. First cover positive scenarios
6. No unnecessary steps included
7. Linked to the requirements object of testing
8. The test cases should be short and simple sentences should be used
9. There should be some naming convention agreed

Test suites

Definition

A combination of test cases that check a concrete component of the product or concrete specification

Working with test suite

- Adding more test cases to the suite
- Adding more details to the test cases
- Delete, obsolete the test cases in the suite
- Improve maintainability of the test cases

Test suite attributes

Attributes:

- Author
- Specification ID
- PM
- Developer
- Priority
- Overview
- Global Setup
- Included test cases

Test suite example

Author	
Spec ID	
PM	
Developer	
Priority	1-4 (1 is the highest priority)
Overview	Short description explaining what the test suite includes
Global setup	Global prerequisites needed for the test cases included in the test suite

Test cases included in the suite

TCID	Name
TC001	Successful registration in abv.bg
TC002
TC008

Test suites by different criteria

Combining test cases based on different criteria:

- ☐ By priority
- ☐ By specification area object of testing
- ☐ By components of testing
- ☐ By types of testing (NFT, RT)

Test case management

TestLink

Zephyr For Jira

Exercises

1. Create positive and negative test cases for login in abv.bg
2. Review the Functional specification document available in Moodle and create test cases based on it.

Q & A

THANK YOU