

Development - QA Test Case Template

01/02/2024 6:48 pm EST

We must move towards having documented, extract step test cases in a consistent format across the company. This is so that new QA hires can quickly start working AND so that we can clearly communicate with the rest of the company what we tested and how we tested it.

Overview

- Aim for 100% test coverage for any given area or functionality.
 - This does not mean 100% test coverage will be achieved this means that we want to approach any area from every angle we can think of to get as much coverage in an area as possible.
- Goal is to easily hand off tests to anyone to be able to test and check if the functionality is correct.
- Test Cases will follow a general outline.
 - Allows for consistency.
 - Can be easily referenced.
 - Can be easily followed from start to finish.
- Test Cases Use own test data.
 - Is the starting point for test execution.
 - Leaves system in a known, good condition/state.

Quality Risks

- There will be scenarios and data combinations that will not be covered within every test case and multiple test cases may need to be created to cover a single user story or functionality. These are not exhaustive tests but are the building blocks to create an exhaustive repository.

What is a Test Case?

- A test case is a document which has a set of conditions or actions that are performed on the software application to verify the expected functionality of the feature.
 - There are several platforms that can be utilized to document test cases. Ex. DevOps, TestRail, Excel.
 - Uses a Requirements Traceability Matrix (RTM) to track the requirements and to check the current requirements are met.
 - Multi layered spreadsheet that links Test Cases to either Tickets or Test Plans.
 - Allows to easily follow and understand what Test Cases pass or not.
 - Endorses critical thinking by creating test cases that cover obscure situations or functionality that is not often used.
 - Start to finish walk through guide that can be handed off to any member of the team.

Test Case Format

- A test case must include:
 - Test Case ID
 - A test case should have a name or title that is self-explanatory.
 - Test Case Description
 - The description should tell the tester what they are going to test in brief.
 - Pre-Conditions
 - Any assumptions that apply to the test and any preconditions that must be met prior to the test being executed should be listed here.
 - Test Case Steps
 - The test steps should include the necessary data and information on how to execute the test. The steps should be clear and brief, without leaving out essential facts.
 - All Test Cases should start at the same starting point.
 - Test Data
 - It is important to select a data set that gives sufficient coverage. Select a data set that specifies not only the positive scenarios but negative ones as well.
 - If any data needs to be setup, the time to do it is while writing the Test Case.
 - Expected Results
 - The expected results tell the tester what they should experience following the test steps.
 - Attach screenshots if needed. (A great screenshot app that allows for boxes and arrows in windows is LightShot)
 - Actual Results
 - The tester can specify how the application behaved while test cases were being executed.
 - Pass/Fail Status
 - Easily Visible Pass or Fail Status
 - The tester can easily mark the Test Case as a Pass or Fail to quickly move through testing several Test Cases.
 - Test Environment
 - It is important to document about the environment the product was tested on. This would include but not be limited to Windows version, installed software, changes to system components or system software.
 - Source Work Item link
 - Easily see what work items the Test Case is linked to.
 - Comments
 - Any other useful information such as screenshots that tester wants to specify can be included here.

- Along with these parameters, testers can include additional parameters like test case priority, type of test case, bug id, etc.

Test Case Design Techniques and Things to Consider

- Stand in the Customer's Shoes
 - Any tester must be able to relate to customers, anticipate their needs and create test cases accordingly.
 - The test cases should be able to test features in line with how customers use them.
 - Keep the customers' requirements up front when writing good test cases.
- Test Case User Personas
 - May not be required for every feature but is a great way to understand user requirements.
 - A fictional profile of an end-user of a certain website or application.
 - For example. Alarm Operator, Manager, and Finances & Recording employees could all have different levels of access and can use the software vastly different from one another.
 - Focuses Test Cases on examining the user interface so that it is easily navigable and visually appealing.
 - Test Cases should be built in alignment with a persona.
 - Example: System Admin testing vs Operator testing
- Aim for 100% Test Coverage
 - The whole point of writing good test cases is providing test coverage as widely as possible.
 - Each test case must aim to cover as many features, user scenarios and as much of the workflow as possible.
 - Keep good flow from start to finish.
 - Create another Test Case if need be to cover other scenarios.
 - Plan test cases to cover each component, feature, and function outlined.
 - Think about what it takes to test each area.
 - Is there any setup required?
 - What data will be used?
 - What persona are you creating this test as?
 - Can another Test Case be created to have more test coverage?
- Test Case Management Tooling
 - Necessary for managing a stable release cycle.
 - Requires maintenance.
 - Should be done at the beginning of every testing cycle.
 - Organize existing Test Cases.
 - Update any outdated Test Cases.
 - Offers transparency for testing progress.
 - Offers metrics for Pass/Fail averages, number of Tests, etc..

- Dependent Test Cases
 - It is best to avoid this as far as possible since bug replication or even review of test cases tend to take more time and effort.
 - Test Cases can reference one another but all steps should be complete.
- Use Automation in Your Favor
 - Mark Test Cases that deal with repetitive actions for automation
 - When creating the Test Case keep in mind that the test case can be used to quickly create automation tests
 - Make sure all set up is notated.
 - Any specific datasets used.
- Test Scenarios Identification and Good Practice
 - Ease the testing and evaluation of a complex system.
 - Enumerate possible users, their actions, and objectives.
 - Evaluate users with hacker's mindset and list possible scenarios of system abuse.
 - List the system events and how does the system handle such requests.
 - List benefits and create end-to-end tasks to check them.
 - Read about similar systems and their behavior.
 - Know what else is available to your clients
 - Studying complaints about competitor's products and their predecessor.
 - Could me in conjunction with market research
- Ineffective Test Cases
 - Covers small part of a functionality.
 - Each test case should be crafted to test as much of the workflow as possible and doing so while operating across technical boundaries of the software being tested.
 - Simply repeating the requirements.
 - This narrows the scope and does not find "Corner-Cases".
 - Include Test Cases for "Corner-Cases"
 - Challenging but detects flaws that are most likely to disrupt the use experience.
 - Not Cataloged Systematically
 - Test cases should be filed in a manner that enables a test management system to access the right one at any time.
 - Leads to functionality being missed.
 - Organization and spending the time on maintaining organization creates effective repositories that a tester can quickly go from one test case to the next.