Chapter 5

QA Test Plan and Test Cases

Contents

What is a Test Plan Document?

- a detailed document that describes the test strategy, objectives, schedule, estimation and deliverables and resources required for testing.
- helps us determine the effort needed to validate the quality of the application under test.
- serves as a blueprint to conduct software testing activities as a defined process which is minutely monitored and controlled by the test manager.

Importance of Test Plan

- Help define project requirements, identify weaknesses and inconsistencies,
- Provide data about the level of quality and test a product during all stages of software dev. lifecycle.
- Only code-based tests aren’t enough, QA provides a creative human element, enhancing the success of coded tests.

How to write a Test Plan

1. Analyze the product (Architechture, users, purpose, workflow, SR/HR requirements)
2. Design the Test Strategy (scope, types, issues, Logistics)
3. Define the Test Objectives
4. Define Test Criteria (80%-100% of test cases coverage)
5. Resource Planning (HR, SR)
6. Plan Test Environment (Test team-Dev Team)
7. Schedule & Estimation (Include test-time In timeline)
8. Determine Test Deliverables (documents, tools, components)

Important Link: **Detailed briefing for each of the points above**

TEST PLAN SAMPLE

Test Deliverable Components

1. Test deliverables are provided before testing phase.

- Test plans document.
- Test cases documents

2. Test deliverables are provided during the testing

- Test Scripts
- Test Data
- Error logs and execution logs.

3. **Test deliverables are provided after the testing cycles is over.**

- Test Results/reports
- Defect Report
- Installation/ Test procedures guidelines
- Release notes

What is a Scenario

- An action the user may undertake with a website or app
- A situation the user may find themselves in while using that software.
- Created to ensure that every single functionality offered by a website or app is working as expected.

Why create a Test Scenario

- Creating Test Scenarios ensures complete Test Coverage
- Customers to ensure the Application Under Test is thoroughly tested. It ensures that the software is working for the most common use cases.
- They serve as a quick tool to determine the testing work effort and accordingly create a proposal for the client or organize the workforce.
- They help determine the most important end-to-end transactions or the real use of the software applications.
- For studying the end-to-end functioning of the program, Test Scenario is critical.

When Not to Create Test Scenarios

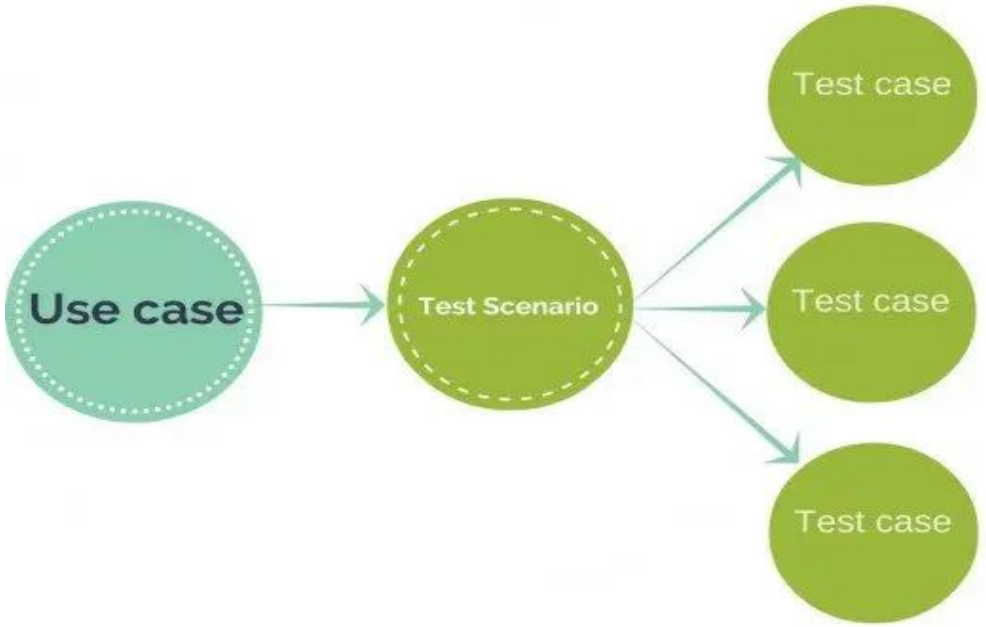
- The Application Under Test is complicated, unstable and there is a time crunch in the project.
- Projects that follow Agile Methodology like Scrum, Kanban may not create Test Scenarios.
- Test Scenario may not be created for a new bug fix or Regression Testing. In such cases, Test Scenarios must be already heavily documented in the previous test cycles. This is especially true for Maintenance projects.

How to create a Test Scenario

- **Step 1.** Carefully study the Requirement Document – Business Requirement Specification (BRS), Software Requirement Specification (SRS), Functional Requirement Specification (FRS)
- **Step 2.** identify what possible user actions need to be tested for it. Figure out the technical issues associated with the requirement. Also, remember to analyze and frame possible positive and negative scenarios.
- **Step 3.** Enumerate test scenarios that cover every possible feature of the software. Ensure that these scenarios cover every user flow and business flow involved in the operation of the website or app.
- **Step4.** After listing the test scenarios, create a Traceability Matrix to ensure that every requirement is mapped to a test scenario.
- **Step 5.** Get the scenarios reviewed by a supervisor, and then push them to be reviewed by other stakeholders involved in the project.

What is Test Case?

- A Test Case is a set of actions executed to verify a particular feature or functionality of your software application.
- Compares Expected and Actual results



A typical Test Case contains:

- TC_Name
  Format:TC_Module_SubmoduleName_Functionality
- Test Scenario
- Type
- Steps
- Description
- Expected result

Execute

- Actual Result
- Status
- (Pass/ Fail)

Example 1: Test Scenario for eCommerce Application

1. **Test Scenario 1:** Check the Login Functionality
   Specific Test Cases for this Test Scenario would be
   a. Check system behavior when valid email id and password is entered.
   b. Check system behavior when *invalid* email id and *valid* password is entered.
   c. Check system behavior when *valid* email id and *invalid* password is entered.
   d. Check system behavior when *invalid* email id and *invalid* password is entered.
   e. Check system behavior when email id and password are left blank and Sign in entered.
   f. Check Forgot your password is working as expected
   g. Check system behavior when valid/invalid phone number and password is entered.
   h. Check system behavior when “Keep me signed” is checked
   Note: As evident, Test Cases are more specific.

Test Scenario & Test Cases

2. **Test Scenario 2:**Check the Search Functionality
3. **Test Scenario 3:** Check the Product Description Page
4. **Test Scenario 4:** Check the Payments Functionality
5. **Test Scenario 5:** Check the Order History
6. **Test Scenario 6** :Check Home Page behavior for returning customers
7. **Test Scenario 7:** Check Category/Product Pages
8. **Test Scenario 8:** Check Customer Service/Contact Pages
9. **Test Scenario 9:** Check Daily Deals pages

Test Scenario Template

Example of Test case

- Test Case Example 1
- Test Case Example 2

Test case covering login page

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results
TU01	Check Customer Login with valid Data	1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = pass99	User should Login into an application	As Expected
TU02	Check Customer Login with invalid Data	1. Go to site http://demo.guru99.com 2. Enter UserId 3. Enter Password 4. Click Submit	Userid = guru99 Password = glass99	User should not Login into an application	As Expected

What is Requirements Traceability Matrix (RTM)?

- A Traceability Matrix is a document that co-relates any two-baseline documents that require a many-to-many relationship to check the completeness of the relationship.
- It is used to track the requirements and to check the current project requirements are met.

Requirement Traceability Matrix?

- **RTM** is a document that maps and traces user requirement with test cases.
- It captures all requirements proposed by the client and requirement traceability in a single document, delivered at the conclusion of the Software development life cycle.
- The main purpose of RTM is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

What is Requirements Traceability Matrix (RTM)?

Importance:

- A simple way is to trace the requirement with its corresponding test scenarios and test cases.

Types and How to Create RTM

Assignment

1. Write scenarios and test case of the test site(atleast 10 scenarios) (https://www.demoblaze.com/index.html)

Link for Practice and Learning

- https://www.demoblaze.com/index.html
- https://www.applause.com/blog/functional-testing-types-examples
- https://www.parasoft.com/blog/how-to-write-test-cases-for-software-examples-tutorial/
- https://bambooagile.eu/insights/what-is-stlc/
- https://medium.com/@concisesoftware/everything-you-should-know-about-qa-in-software-development-the-beginners-guide-3e7afacf607c
- https://www.guru99.com/what-everybody-ought-to-know-about-test-planing.html
- https://www.guru99.com/test-scenario.html
- https://www.guru99.com/traceability-matrix.html

THANK YOU!!

ANY QUERIES?