Test Case development

What is Test case?

A test case is a written document that was created for a particular test scenario to check compliance with a certain requirement. It contains a set of test data, preconditions, expected results, and postconditions.

Test Case acts as the starting point for the test execution, and after applying a set of input values, the application has a definitive outcome and leaves the system at some end point or also known as execution postcondition.

Typical Test Case Parameters:

- Test Case ID
- Test Scenario
- Test Case Description
- Test Steps
- Prerequisite
- Test Data
- Expected Result
- Test Parameters
- Actual Result
- Environment Information

Example:

Let us say that we need to check an input field that can accept maximum of 10 characters.

While developing the test cases for the above scenario, the test cases are documented the following way. In the below example, the first case is a pass scenario while the second case is a FAIL.

Scenario	Test Step	Expected Result	Actual Outcome
Verify that the input field that can accept maximum of 10 characters	Login to application and key in 10 characters	Application should be able to accept all 10 characters.	Application accepts all 10 characters.
Verify that the input field that can accept maximum of 11 characters	Login to application and key in 11 characters	Application should NOT accept all 11 characters.	Application accepts all 10 characters.

If the expected result doesn't match with the actual result, then we log a defect. The defect goes through the defect life cycle and the testers address the same after fix.

What is a Test Scenario?

A Test Scenario is defined as any functionality that can be tested. It is also called Test Condition or Test Possibility.

Scenario Testing

Scenario Testing in software testing is a method in which actual scenarios are used for testing the software application instead of test cases. The purpose of scenario testing is to test end to end scenarios for a specific complex problem of the software. Scenarios help in an easier way to test and evaluate end to end complicated problems.

How to Write Test Scenarios

- Step 1: Read the Requirement Documents like BRS, SRS, FRS, of the System Under Test (SUT). You could also refer uses cases, books, manuals, etc. of the application to be tested.

- Step 2: For each requirement, figure out possible users actions and objectives. Determine the technical aspects of the requirement. Ascertain possible scenarios of system abuse and evaluate users with hacker's mindset.

- Step 3: After reading the Requirements Document and doing your due Analysis, list out different test scenarios that verify each feature of the software.

- Step 4: Once you have listed all possible Test Scenarios, a Traceability Matrix is created to verify that each & every requirement has a corresponding Test Scenario.

- Step 5: The scenarios created are reviewed by your supervisor. Later, they are also reviewed by other Stakeholders in the project.

Requirements Traceability Matrix

To build the highest quality product, you need to make sure requirements are met, tests are run, and issues are resolved. And, optimally, you want to tie all of these items together so you can trace test results to requirements, requirements to issues, and so on. This requirements traceability is particularly important if you're in a highly regulated industry and need to prove compliance.This is met bycreating a requirements traceability matrix.

How to create a Requirement Traceability Matrix

The first step to build a requirements traceability matrix is to create the template, or shell, of your matrix. This is where you'll determine what you want to trace and why, and collect the necessary documents.

1)Define Your Goal

Your first step when creating a traceability matrix —whether you're using Excel or a dedicated requirements management tool —is to define your goal.

2)Define and Gather Your Artifacts

Next, you'll need to define which artifacts should be included, based on your goal.

At its most basic, a traceability matrix should include:

- Requirements

- Tests
- Test results
- Issues

Once you've defined your artifacts, you'll need to gather them. You'll also need to track down your test cases and results. If testing is in progress or completed, you'll need to find test statuses.If any tests have failed, you'll also need to gather any issues that may have been detected.

3)Create Your Requirements Traceability Matrix Template

Once you've defined and gathered your documents, you're ready to make your template.

You'll need to add a column for each of your artifacts. For a basic traceability matrix, your columns will be:

- Column 1: Requirements
- Column 2: Tests
- Column 3: Test Results
- Column 4: Issues

Test Bed

The test execution environment configured for testing. Test bed consists of specific hardware, software, Operating system, network configuration, the product under test, other system software and application software.

Test Bed configuration

It is the combination of hardware and software environment on which the tests will be executed. It includes hardware configuration, operating system settings, software configuration, test terminals and other support to perform the test.

Example

```
Web Server - IIS/Apache
Database - MS SQL
OS - Windows/ Linux
Browser - IE/FireFox
Java version : version 6
```