SOFTWARE TESTING - DOCUMENTATION

Testing documentation involves the documentation of artifacts that should be developed before or during the testing of Software.

Documentation for software testing helps in estimating the testing effort required, test coverage, requirement tracking/tracing, etc. This section describes some of the commonly used documented artifacts related to software testing such as:

- Test Plan
- Test Scenario
- Test Case
- Traceability Matrix

Test Plan

A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment in which testing will be performed, and the limitations of the testing and the schedule of testing activities. Typically the Quality Assurance Team Lead will be responsible for writing a Test Plan.

A test plan includes the following:

- Introduction to the Test Plan document
- Assumptions while testing the application
- List of test cases included in testing the application
- List of features to be tested
- What sort of approach to use while testing the software
- List of deliverables that need to be tested
- The resources allocated for testing the application
- Any risks involved during the testing process
- A schedule of tasks and milestones to be achieved

Test Scenario

It is a one line statement that notifies what area in the application will be tested. Test scenarios are used to ensure that all process flows are tested from end to end. A particular area of an application can have as little as one test scenario to a few hundred scenarios depending on the magnitude and complexity of the application.

The terms 'test scenario' and 'test cases' are used interchangeably, however a test scenario has several steps, whereas a test case has a single step. Viewed from this perspective, test scenarios are test cases, but they include several test cases and the sequence that they should be executed. Apart from this, each test is dependent on the output from the previous test.
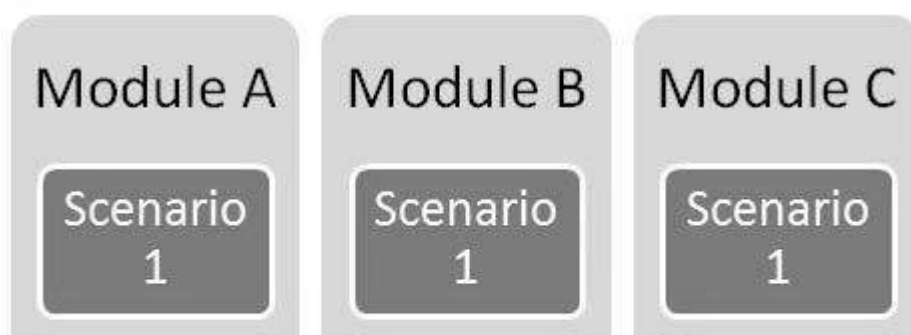
Test Case

Test cases involve a set of steps, conditions, and inputs that can be used while performing testing tasks. The main intent of this activity is to ensure whether a software passes or fails in terms of its functionality and other aspects. There are many types of test cases such as functional, negative, error, logical test cases, physical test cases, UI test cases, etc.

Furthermore, test cases are written to keep track of the testing coverage of a software. Generally, there are no formal templates that can be used during test case writing. However, the following components are always available and included in every test case:

- Test case ID

- Product module

- Product version

- Revision history

- Purpose

- Assumptions

- Pre-conditions

- Steps

- Expected outcome

- Actual outcome

- Post-conditions

Many test cases can be derived from a single test scenario. In addition, sometimes multiple test cases are written for a single software which are collectively known as test suites.

Traceability Matrix

Traceability Matrix $also known as Requirement Traceability Matrix - RTM$ is a table that is used to trace the requirements during the Software Development Life Cycle. It can be used for forward tracing $i.e. from Requirements to Design or Coding$ or backward $i.e. from Coding to Requirements$. There are many user-defined templates for RTM.

Each requirement in the RTM document is linked with its associated test case so that testing can be done as per the mentioned requirements. Furthermore, Bug ID is also included and linked with its associated requirements and test case. The main goals for this matrix are:

- Make sure the software is developed as per the mentioned requirements.

- Helps in finding the root cause of any bug.

- Helps in tracing the developed documents during different phases of SDLC.

Loading [MathJax]/jax/output/HTML-CSS/jax.js