MANUAL TESTING CONCEPTS

What is Software Testing?

Software testing is the process of evaluating a system with the intend of finding bugs. It is performed to check if the system satisfies its specified requirements. Testing measures the overall quality of the system in terms of its correctness, completeness, usability, performance and other functional and non-functional attributes.

Why is Testing required?

Software Testing as a separate activity in SDLC, is required because-

- Testing provides an assurance to the stakeholders that product works as intended.
- Avoidable defects leaked to the end user/customer without proper testing adds bad reputation to the development company.
- Separate testing phase adds a confidence factor to the stakeholders regarding quality of the software developed.
- Defects detected in earlier phase of SDLC results into lesser cost and resource utilization for defect resolution.
- Saves development time by detecting issues in earlier phase of development.
- Testing team adds another dimension to the software development by providing a different view point to the product development process.

Who does Testing?

Software Testing is/can be done by all technical and non technical people associated with the software. Testing in its various phases is done by-

- **Developer** - Developer does the unit testing of the software and ensure that the individual methods work correctly
- **Tester** - Testers are the face of the software testing. A tester verifies the functionality, usability of the application as functional tester, a tester checks the performance of the application as a Performance tester, a tester automates the manual-functional test cases and creates test scripts as an automation tester

- **Test Managers/Lead/Architects** - Define the test strategy and test plan
- **End users** - A group of end users do the User Acceptance Testing (UAT) of the application to make sure the software can work in the real world

What is Software Quality?

- Software quality is the conformance of a software system to its requirements. In software perspective, quality is defined by two factors - Validation and Verification. Validation checks if the process used during software development is correct or not whereas in verification the product is evaluated to check if its meets the specifications.

Attributes of Software Quality

- **Correctness** - Correctness measures the software quality for the conformance of the software to its requirements
- **Reliability** - Checks if the software performs its functions without any failure within the expected conditions
- **Robustness** - Robustness is the ability of the software to not crash when provided with unexpected input
- **Usability** - Usability is the ease of operating the software
- **Completeness** - Completeness is the extent tp which the software system meets its specifications
- **Maintainable** - Maintainability is the measure of the amount of effort required for software maintenance after it has shipped to end user
- **Portability** - Ability of the software to be transformed from one platform or infrastructure to other
- **Efficiency** - Efficiency is the measure of resources required for the functioning of the software

What is Test Strategy?

- A Test Strategy is a high-level document describing the way testing will be carried out. In a test strategy document, we document the test objectives and set of guidelines for achieving those objectives. It is presented by the project manager to all the stakeholders in the testing process. It can have a scope of an entire organization or a particular project.

Test Strategy Approaches

There are different apporaches to test strategy are-

- **Analytical Approach** - Based on the risk analysis
- **Model-based Approach** - Based on the various statistical models
- **Consultative Approach** - Based on the consultation with technology or domain experts
- **Methodical Approach** - Based on the experience
- **Heuristic Approach** - Based on the exploratory techniques
- **Standard-compliant** Approach - Based on the industry standards and processes

Test Strategy Document Template

A test strategy document can contain the following fields-

- Test Strategy Id - An identifier of the test strategy document and its various versions.
- Introduction - A brief introduction to the purpose and scope of the document.
- Standards to use - The different standards or set of guidelines to be followed.
- Risks and Mitigations - The different risks associated with in testing and their mitigation strategies.
- Entry Critieria - The set of pre-requisite that must be performed before testing can start.
- Exit Critieria - The criteria defining when the testing can be stopped.
- Test design techniques - The test design techniques to be used like - equivalence partitioning, boundary value analysis etc.
- Test environment - The test environment specifications.
- Configuration management of testware - Specification of the right version of testware for testing.
- Test process improvement - The apporaches to use for improving the test process.
- Approvals - The persons approving the test strategy document.

What is a defect?

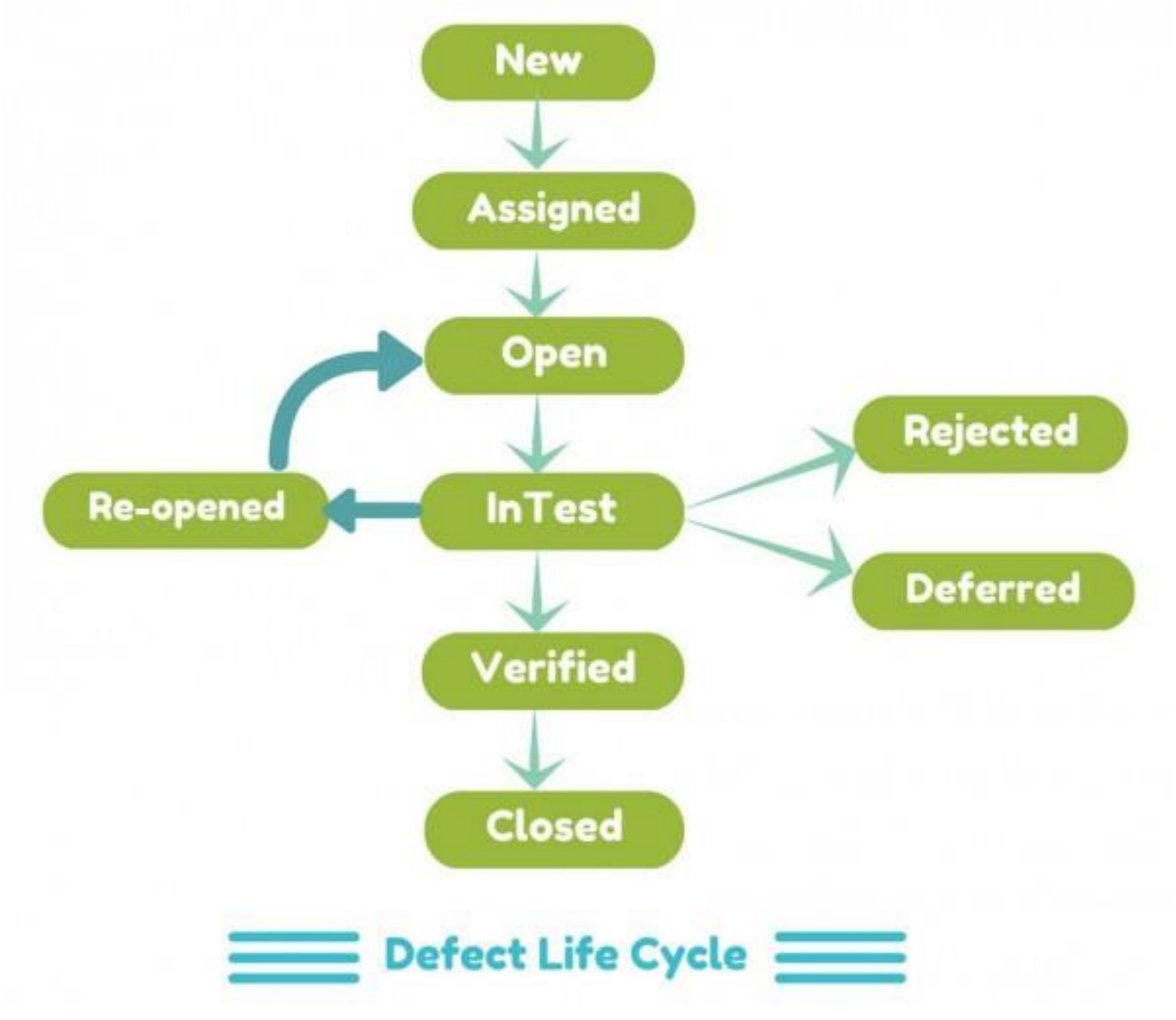
- A defect or a bug is an error in a program that causes the application to perform in an unintended manner, deviating from its requirements. Based on the urgency of fixing the defect, we can classify them on a scale of P0 to P3, with P0 defect having the most urgency to fix. Also, the defects can be classified based on their criticality or the impact to the functionality.
- To report a bug we have different Defect Management Tools like - Jira, Mantis, Bugzilla etc.

Next, we will see the different components of a Defect Report.

Defect Reporting Template:

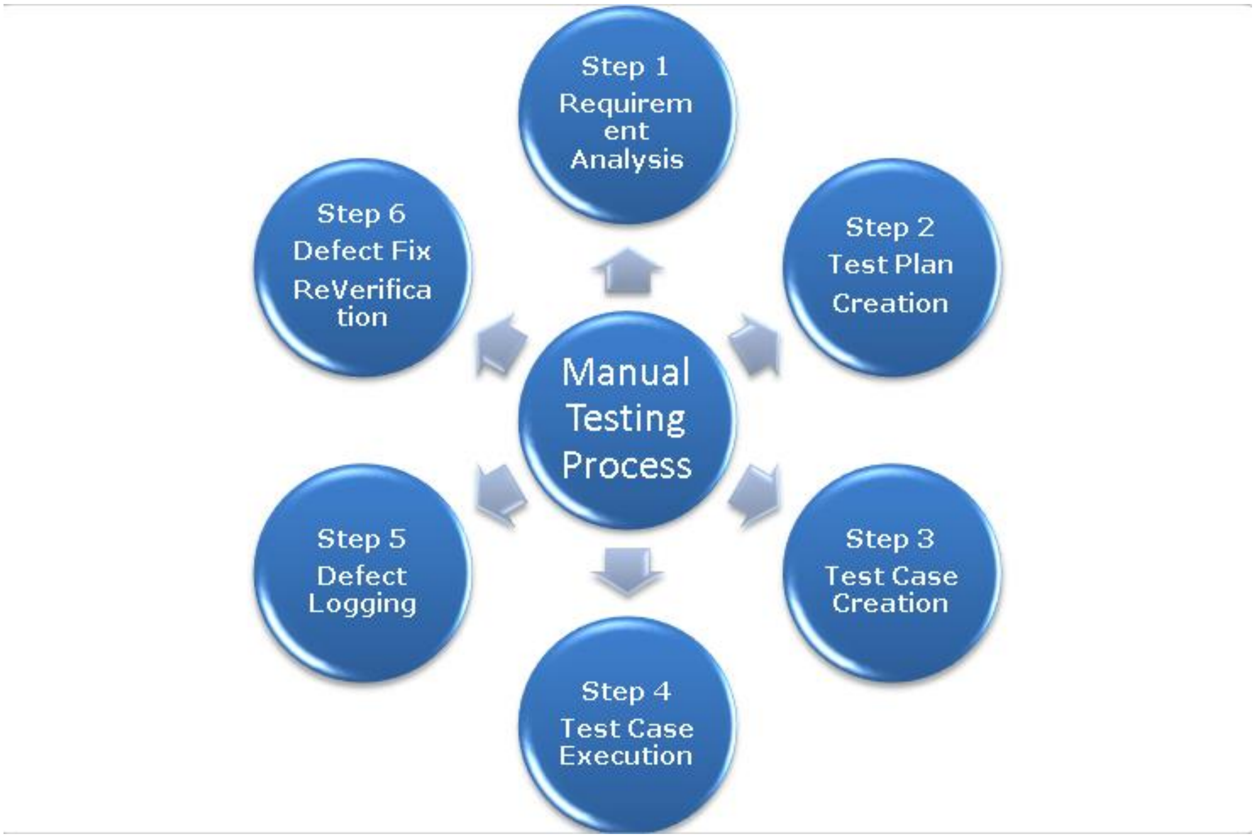
- DefectId - A unique identifier of the defect.
- Summary - A one line summary of the defect, more like a defect title.
- Description - A detailed description of the defect.
- Build Version - Version of the build or release in which defect is found.
- Steps to reproduce - The steps to reproduce the defect.
- Expected Behavior - The expected behavior from which the application is deviating because of the defect.
- Actual Behavior - The current erroneous state of the application w.r.t. the defect.
- Priority - Based on the urgency of the defect, this field can be set on a scale of P0 to P3.
- Severity - Based on the criticality of the defect, this field can be set to minor, medium, major or show stopper.
- Reported By - Name of the QA, reporting the defect.
- Reported On - The date on which the defect was raised.
- Assigned To - The person to whom the defect is assigned in its current state. It can be the developer fixing the defect, the QA for verification of the fixed defect or the manager approving the defect.
- Current Status - The current status of defect (one of the states of the defect life cycle).
- Environment - The environment on which the defect is found - release, staging, production etc.

What is a defect life cycle?

- A defect life cycle is the movement of a bug or defect in different stages of its lifetime, right from the beginning when it is first identified till the time is marked as verified and closed.
- Depending on the defect management tool used and the organisation, we can have different states as well different nomenclature for the states in the defect life cycle.
- 

- **New** - A bug or defect when detected is in New state
- **Assigned** - The newly detected bug when assigned to the corresponding developer is in Assigned state
- **Open** - When the developer works on the bug, the bug lies in Open state

- **Rejected/Not a bug** - A bug lies in rejected state in case the developer feels the bug is not genuine
- **Deferred** - A deferred bug is one, fix of which is deferred for some time(for the next releases) based on urgency and criticality of the bug
- **Fixed/InTest** - When a bug is resolved by the developer it is marked as fixed and assigned to the tester
- **Reopened** - If the tester is not satisfied with issue resolution the bug is moved to Reopened state
- **Verified** - After the Test phase if the tester feels bug is resolved, it is marked as verified
- **Closed** - After the bug is verified, it is moved to Closed status.



What is Test Strategy?

- A Test Strategy is a high-level document describing the way testing will be carried out.
- In a test strategy document, we document the test objectives and set of guidelines for achieving those objectives.
- It is presented by the project manager to all the stakeholders in the testing process. It can have a scope of an entire organization or a particular project.

Test Strategy Approaches:

The are different apporaches to test strategy are-

- Analytical Approach - Based on the risk analysis
- Model-based Approach - Based on the various statistical models
- Consultative Approach - Based on the consultation with technology or domain experts
- Methodical Approach - Based on the experience
- Heuristic Approach - Based on the exploratory techniques
- Standard-compliant Approach - Based on the industry standards and processes

Test Strategy Document Template:

A test strategy document can contain the following fields-

- **Test Strategy Id** - An identifier of the test strategy document and its various versions.
- **Introduction** - A brief introduction to the purpose and scope of the document.
- **Standards to use** - The different standards or set of guidelines to be followed.
- **Risks and Mitigations** - The different risks associated with in testing and their mitigation strategies.
- **Entry Critieria** - The set of pre-requisite that must be performed before testing can start.
- **Exit Critieria** - The criteria defining when the testing can be stopped.
- **Test design techniques** - The test design techniques to be used like - equivalence partitioning, boundary value analysis etc.
- **Test environment** - The test environment specifications.

- **Configuration management of testware** - Specification of the right version of testware for testing.
- **Test process improvement** - The apporaches to use for improving the test process.
- **Approvals** - The persons approving the test strategy document.

What is a Test Plan?

- A Test Plan is a formal document derived from requirement documents (Software Requirement Specification, Use Case documents etc.), describing in detail the scope of testing and the different activities performed in testing.
- It is generally prepared by a test manager and approved by the different stakeholders of the application.

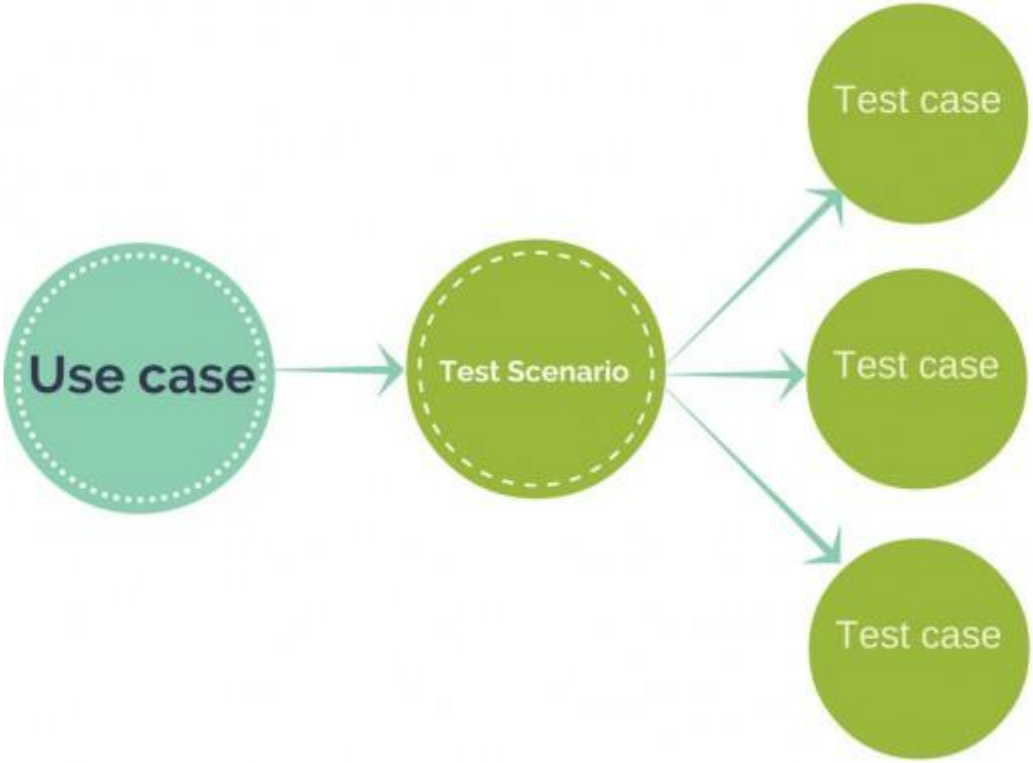
Features of a Test Plan

A Test Plan needs to address the following-

- Overall scope of testing
- Risk Analysis
- Test estimate
- Resource Requirement
- Tools used
- Scheduling, review and analysis of test design activities
- Creation of test cases and test data
- Identification of test monitoring and test control activities

What is a Test Scenario?

- A Test Scenario is generally a one line statement describing a feature of application to be tested. It is used for end to end testing of a feature and is generally derived from use cases.
- A single test scenario can cover one or more test cases i.e. it has a one to many relationship with test cases.



What is Scenario testing?

- Scenario testing is a type of testing carried out using scenarios derived from the use cases. Using scenario testing, complex application-logic can be tested using easy to evaluate test scenarios.

Some advantages of test scenarios are -

- Test scenarios can serve as basis for lower level test case creation.
- Testing using test scenarios can be carried out relatively faster than the one using test cases.
- Saves a lot of time, better with projects having time constraints.

What is a Test Case?

- A test case is a set of conditions for evaluating a particular feature of a software product to determine its compliance with the business requirements.

- A test case has pre-requisites, input values and expected results in a documented form which cover the different test scenarios.

  A test case can have following attributes-
- TestCaseId - A unique identifier of the test case.
- Test Summary - Oneliner summary of the test case.
- Description - Detailed description of the test case.
- Prerequisite or pre-condition - A set of prerequisites that must be followed before executing the test steps.
- Test Steps - Detailed steps for performing the test case.
- Expected result - The expected result in order to pass the test.
- Actual result - The actual result after executing the test steps.
- Test Result - Pass/Fail status of the test execution.
- Automation Status - Identifier of automation - whether the application is automated or not.
- Date - The test execution date.
- Executed by - Name of the person executing the test case.