



www.SyntaxTechs.com

© 2017 Syntax Technologies. All Rights Reserved.

What is Manual Testing?

Goal of Manual Testing

Software Testing Life Cycle

What is Manual Testing?



Manual Testing is :

- A process of finding out the defects or bugs in a software program by manually executing test cases.
- A process in which testers play role as end users to compare the behavior of a developed piece of code (software, module, feature, etc.) against expected behavior (Requirement)
- Preliminary testing that must be carried out prior to start automating the test cases. If manual test case fail we cannot automate until it pass, so **Manual Testing is important.**

What is the Goal of Testing ?

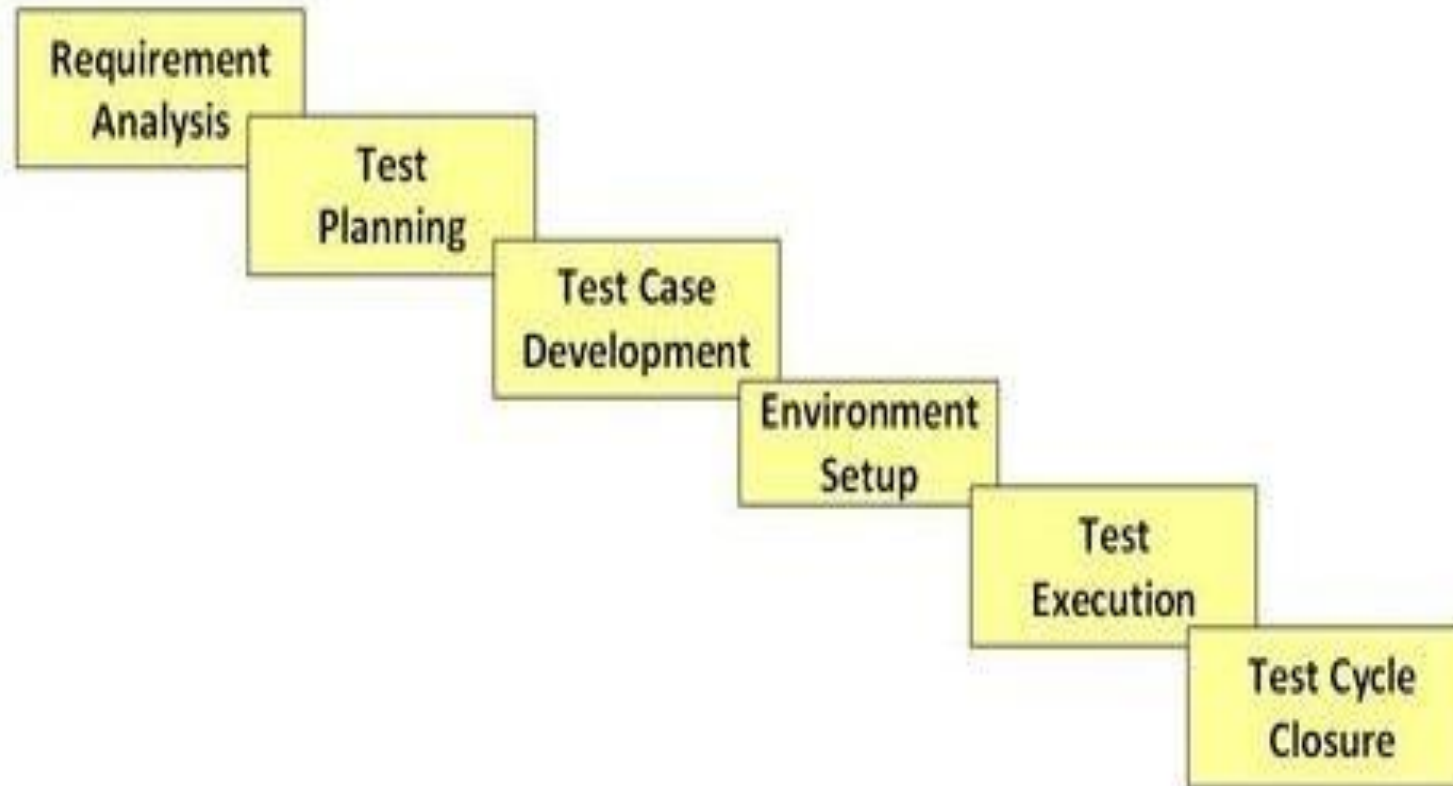


Main goal of testing is to make sure that the application under test is defect free and software application is working as per the requirement specification document.

How do we start Testing?



Software Testing Life Cycle

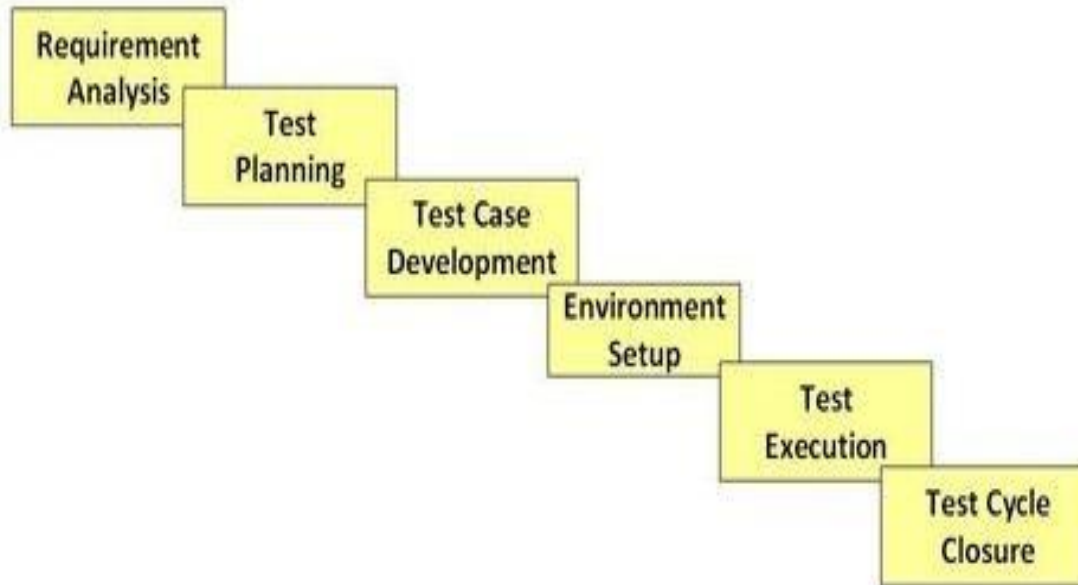


Software Testing Life Cycle



- **STLC** is a process of sequence activities conducted to perform Software Testing.
- In STLC process, different activities are carried out to improve the quality of the product.

Software Testing Life Cycle



Following steps are involved in Software Testing Life Cycle (STLC). Each step is have its own Entry and Exit Criteria.

Requirement Analysis



Test team studies the requirements from a testing point of view to identify the testable requirements.

Entry Criteria: Requirements Specification document should be available

Exit Criteria: List of questions with all answers to be resolved from business i.e. testable requirements.

What is Requirement?



- Requirements are description of features/ functionalities of the target system.
- Requirements are user expectation from the software product.

Test Planning



In this Stage usually **QA Manager or Test Lead** (based on the company) create Test Plan.

Entry Criteria: Requirement Documents.

Exit Criteria: Test Plan is prepared

What is Test Plan ?



- Test Plan is a detailed document that outlines the test strategy, testing objectives, resources (manpower, software, hardware) required for testing, test schedule, test estimation and test deliverables.
- Test Plan serves as a blueprint to conduct software testing.

Test Case Development



This phase involves creation, verification and rework of test cases & test scripts. Test data is identified/created and is reviewed and then reworked as well.

Entry Criteria: Requirements Documents.

Exit Criteria: Test Case/Test Scripts, Test Data

Test Environment Set Up

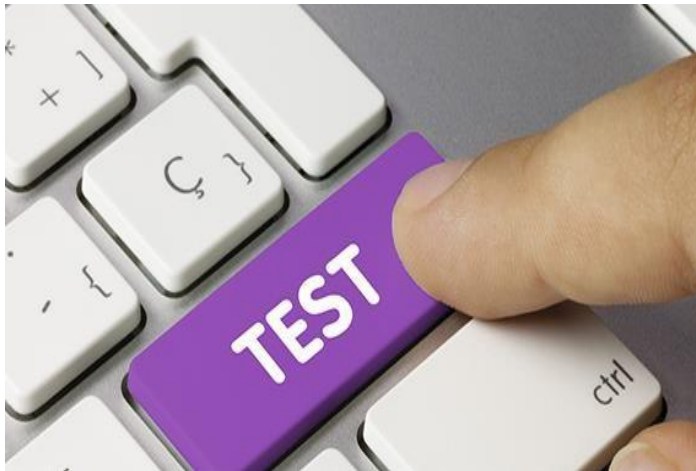


Testing Team is **not involve** in setting up the environment; however, team should prepare the smoke test cases to check the readiness of the test environment setup.

Entry Criteria: Test Plan, Smoke Test cases, Test Data are available.

Exit Criteria: Test Environment is available, results of Smoke Test

Test Execution

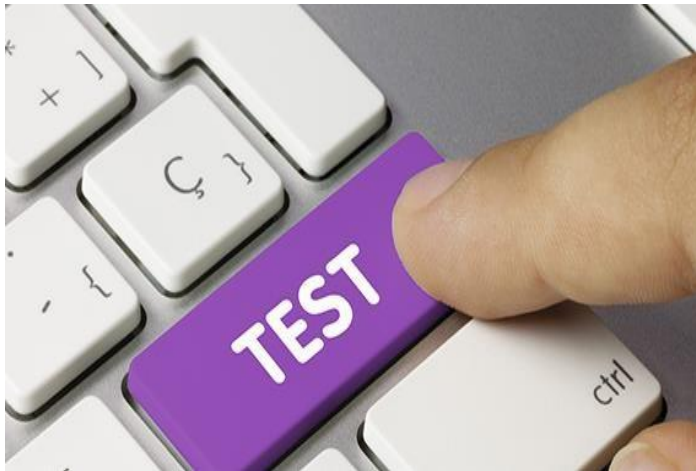


Testing team start executing test cases based on prepared test planning & prepared test cases . Bugs will be reported back to the development team for correction and retesting will be performed.

Entry Criteria: Test Cases and Test Data available

Exit Criteria: Test Case execution report, Defect Report, Completed RTM with execution results

Test Cycle Closure

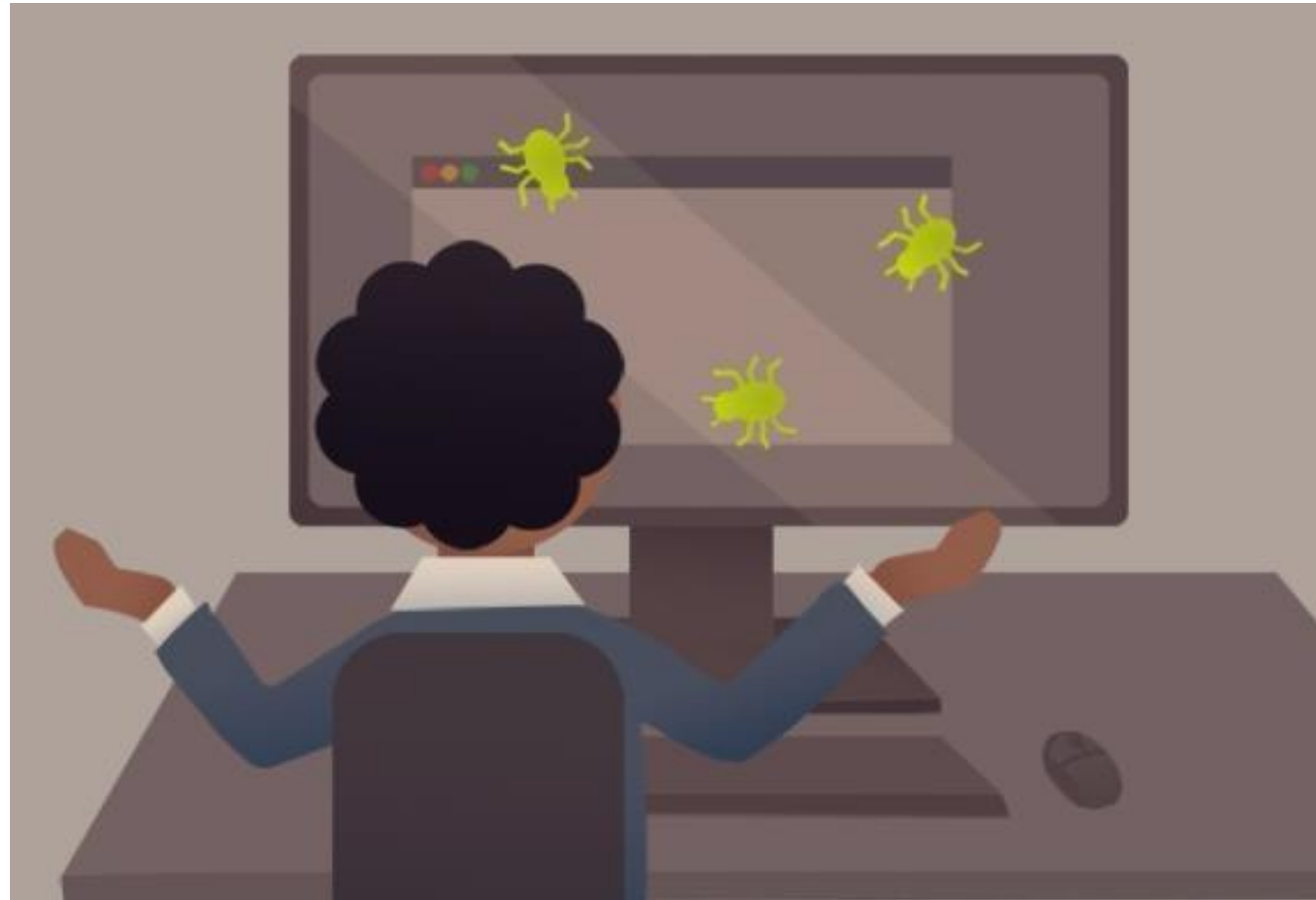


Testing team will meet , discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from the current test cycle.

Entry Criteria: Test Case
Execution Report, Defect Report

Exit Criteria: Test Closure report,
Test Metrics

What should we do after a bug is found?



What should we do after a bug is found?

- 1) ALWAYS: Try to **reproduce** the BUG
- 2) Check requirements (requirements update or functionality removed)
- 3) Once bug is reproducible you need to report it



Priority and Severity of the Bug

- Severity is defined as the level of impact that bug has on the functionality of application/system.
- Severity indicates the seriousness of the defect on the product functionality
- Priority is defined as the order in which a defect should be fixed. Higher the priority the sooner the defect should be resolved.
- Priority indicates how soon the bug should be fixed



Priority categorization



Severity categorization

- **Show Stopper:** Not able to test application further.
- **Major:** Major functionality not working but able to test application.
- **Minor:** The defect that does not result in the termination and does not damage the usability of the system
- **Cosmetic:** Issues in location of the object or the look and feel issue.

- **Low:** The defect is an irritant but repair can be done once the more serious bugs have been fixed
- **Medium:** During the normal course of the development activities defect should be resolved. It can wait until a new version is created
- **High:** The defect must be resolved as soon as possible as it affects the system severely and cannot be used until it is fixed

Examples:



- **High Priority & High Severity:** User is unable to login to their Bank of America account
- **High Priority & Low Severity:** Bank of America logo on the front page misspelled as 'Bank of Amerika'
- **High Severity & Low Priority:** On Syntax Technologies website under 'Campus' section not able to scroll to view pictures
- **Low Priority and Low Severity:** Any cosmetic issues (Image is displaying on top right instead of center right) or spelling issues (when user trying to make purchase there is always acknowledge message)

Bug Life Cycle

- is the journey of a defect cycle, which a defect goes through during its lifetime.
- is the specific set of states that a Bug goes through from discovery to defect fixation.
- It varies from organization to organization, from project to project and also depends upon the tools used.

Bug Life Cycle

- **New:** When a new defect is logged and posted for the first time. It is assigned a status NEW.
- **Assigned:** Once the bug is posted by the tester, the lead of the tester approves the bug and assigns the bug to developer team.
- **Open:** The developer starts analyzing and works on the defect fix
- **Fixed:** When developer makes necessary code change and verifies the change, he or she can make bug status as "Fixed."

Bug Life Cycle

- **Retest:** Tester do the retesting to check whether the bug got fixed or not.
- **Reopened:** If the bug still exists even after the bug is fixed , the tester changes the status to “reopened”. The bug goes through the life cycle once again.
- **Rejected:** If the developer feels that the bug is not genuine, developer rejects the bug. Then the state of the bug is changed to “rejected”.
- **Closed:** Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, tester changes the status of the bug to “closed”. This state means that the bug is fixed, tested and approved.

Overview:

Bug Life Cycle:

