

6: Software Testing Life Cycle

IT6206 – Software Quality Assurance

Level III - Semester 6

Overview

- Software Testing Life Cycle (STLC) is the collection of activities carried out by the testing team to ensure the quality of the final product or software.
- It is an integral part of the Software Development Life Cycle (SDLC). But it specifically deals only with Testing.
- There are several steps that should be followed in a proper order to ensure the software quality completely.
- All those steps or phases in STLC will be discussed in this Chapter.

Intended Learning Outcomes

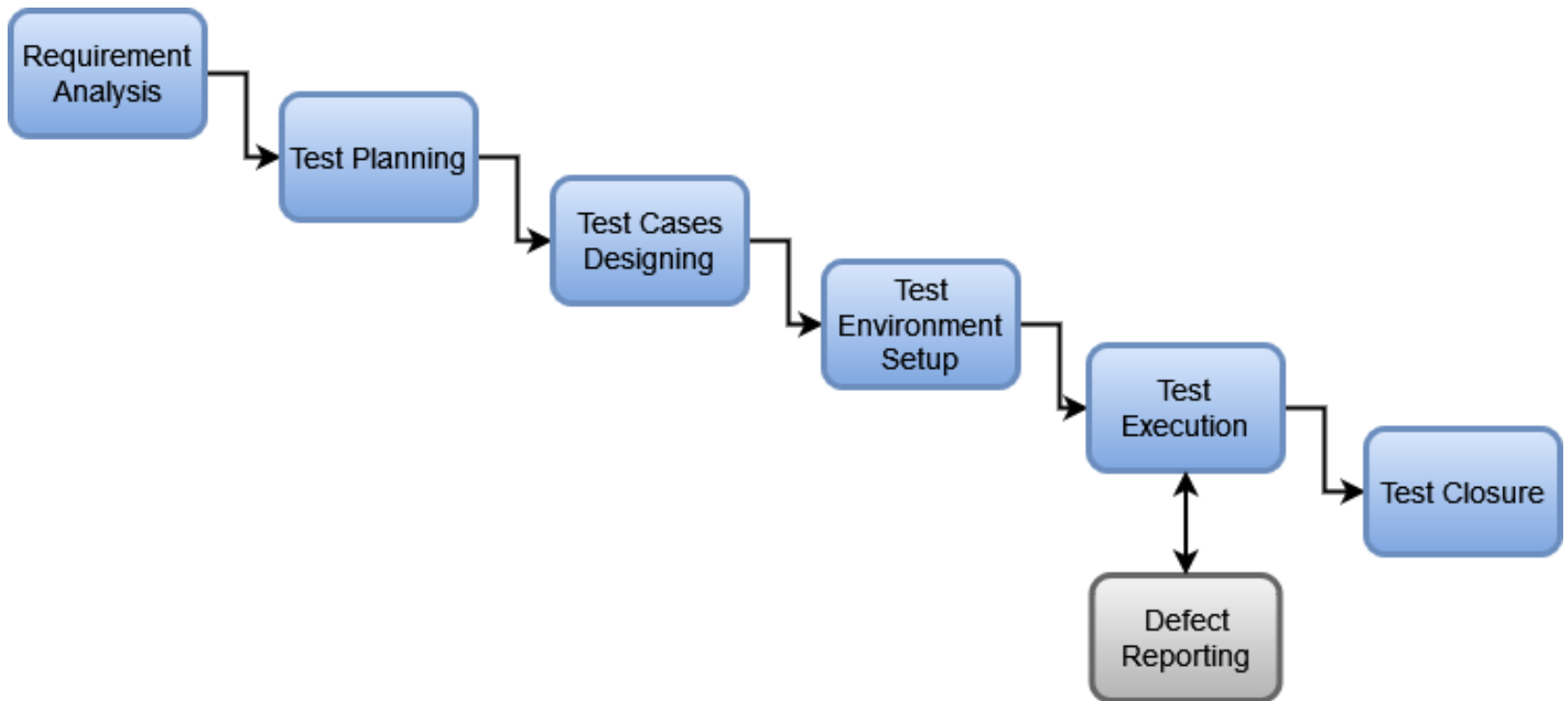
At the end of this lesson, you will be able to;

- Identify the phases of Software Testing Life Cycle.
- Explain each phase of STLC.
- Explain what is an entry criteria and what is an exit criteria.
- Identify entry and exit criteria of different phases.
- Understand defects life cycle.
- Understand and identify defect classifications.

List of sub topics

- 6.1 Requirements Analysis/Design
- 6.2 Entry Criteria and Exit Criteria
- 6.3 Test Planning
- 6.4 Test Case Design and Development
- 6.5 Test Environment Setup
- 6.6 Defect Life Cycle
- 6.7 Defects Classification
- 6.8 Test Execution
- 6.9 Test Closure

Software Testing Life Cycle Phases



6.1 Requirements Analysis/Design

- First phase of STLC
- High level requirement analysis
 - Starts when the Software Requirements Document is ready and shared with the stakeholders
- Testing team identifies the testing requirements
 - Scope
 - Verification and Validation key points
 - Time

6.1 Requirements Analysis/Design

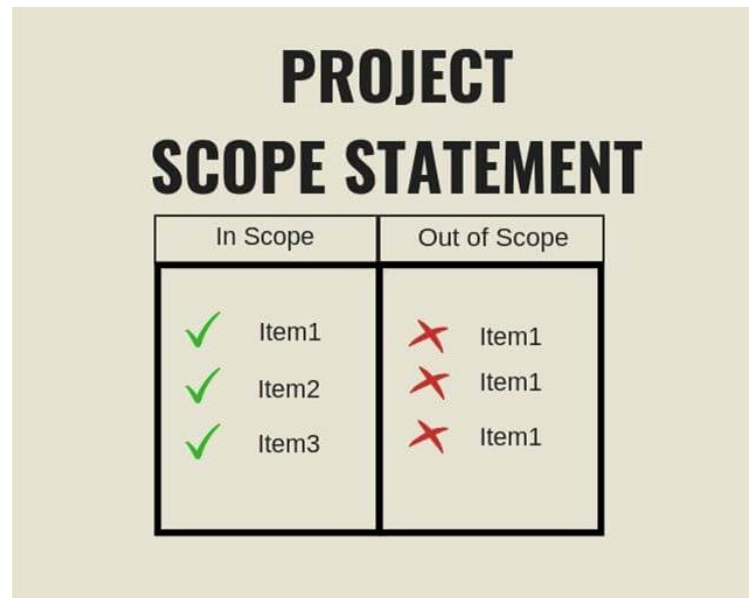
- Scope – identifying the scope of testing
 - Testing scope means the areas of the product that are needed to be tested or the functionalities that are needed to be focused on.
- Verification and Validation key points – gathering testing priorities.
- Time – identifying a draft amount of time that will be needed for the testing process.
 - Better to identify the expected time frame for each functionalities too.

6.1 Requirements Analysis/Design

- Functional and logical relationships between modules are identified
- Discussion with other stakeholders will improve the requirement understanding
- Identifies functional and/or non-functional requirements
- Three main activities performed in this phase are:
 - Scope definition
 - Requirement Traceability Matrix(RTM) preparation
 - Automation analysis

6.1 Requirements Analysis/Design

- Scope definition
 - Identifies the scope of testing at high level and breaks down into several functional modules
 - Identifies the needed types of testing
 - Analyses and identifies the prerequisites and testing environment details



<https://www.scholar99.com/what-is-a-scope-statement-in-project-management/>

6.1 Requirements Analysis/Design

- Requirement Traceability Matrix (RTM) preparation
 - Requirement tracing is a process of documenting the links between the requirements and the work products developed to implement and verify those requirements.
 - RTM is a document that maps and traces user requirements with test cases

Req No	Req Desc	Testcase ID	Status
123	Login to the application	TC01,TC02,TC03	TC01-Pass TC02-Pass
345	Ticket Creation	TC04,TC05,TC06, TC07,TC08,TC09 TC010	TC04-Pass TC05-Pass TC06-Pass TC06-Fail TC07-No Run
456	Search Ticket	TC011,TC012, TC013,TC014	TC011-Pass TC012-Fail TC013-Pass TC014-No Run

<https://www.guru99.com/traceability-matrix.html>

6.1 Requirements Analysis/Design

- Automation analysis
 - Analysis of scope of automation for regression testing.
 - Automation tool, functionalities to be automated, time-frame and resource allocation for the automation are planned.

6.2 Entry Criteria and Exit Criteria

- Entry Criteria
 - Conditions or documents required to begin a particular phase of STLC.
 - Conditions that allow a task to perform and in case of their absence, the task will not be performed.
 - Defining a time frame while setting the entry criteria would be ideal.
- Common entry criteria which are used to mark the beginning of the testing:
 - Requirements are defined and approved
 - Complete or partially testable code is available
 - Availability of test data
 - Availability of test cases
 - Completely set up testing environment

6.2 Entry Criteria and Exit Criteria

- Common entry criteria determined for unit testing:
 - Planning phase is completed
 - Properly reviewed and approved system design, technical design and other documents are available
 - Business and functional requirements are defined and approved
 - Availability of testable codes or units
 - Availability of test environment

6.2 Entry Criteria and Exit Criteria

- Exit Criteria
 - Items/documents/actions/tasks required to be completed before concluding a STLC phase and moving on to the next phase.
 - A set of expectations from a particular phase.
 - Defining a time frame while setting the entry criteria would be ideal.
- Common exit criteria which are used to mark the conclusion of the testing:
 - Deadlines meet.
 - All the test cases are executed.
 - Sufficient coverage of the requirements and functionalities under the test.
 - All the identified defects are corrected and closed.
 - No critical bug has been opted out.

6.2 Entry Criteria and Exit Criteria

- Common exit criteria which are defined for the unit testing:
 - Successful execution of the unit tests.
 - All the identified bugs have been fixed and closed.
 - Completed the project code.

6.3 Test Planning

- Defining the test specifications in order to achieve the project requirements.
- Main objective is to prepare a Test Plan document.
 - A test plan outlines the strategy that will be used to test an application, the resources that will be used, the test environment to perform testing, limitations and the schedule of the testing.
 - Usually prepared by Quality Assurance Team Lead(Test Analyst) or the Manager.
- Analysis is mainly focused on the data/information related to application.
- This phase finds the answers for the following questions:
 - WHAT is to be tested?
 - WHAT RESOURCES are required to test?

6.3 Test Planning

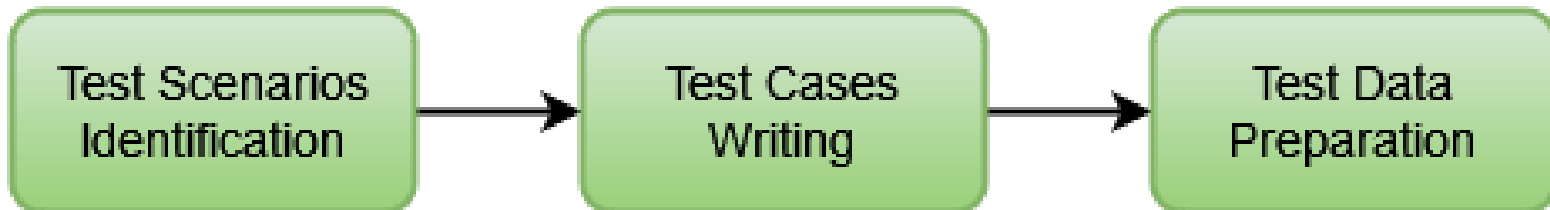
- Basic Entry Criteria for Test Planning phase:
 - Provision of Requirements Documents along with Requirements Traceability Matrix (RTM).
 - If automation is in the scope, Automation Feasibility Report should be prepared to enter this phase.
- The exit criteria of this phase:
 - Completion of Test Plan Document and Test effort Estimation document.
- Major aspects of the Test Planning Phase
 - Scope of Deliverables
 - Effort Estimation
 - Resource Plan

6.4 Test Case Design and Development

- The main objective is to prepare test cases for an individual unit.
- Test cases cover the functionality, points of verification and validation that are mentioned in the Test Plan.
- Written by QA team and signed off by a Business Analyst after the reviewal.
- Then test data is prepared based on preconditions.
- Entry criteria for this phase:
 - Activities in Test Planning should be finished
 - Test Plan should be ready.
- Exit criteria for this phase:
 - Test cases should be signed off
 - Test data should be ready and test scripts prepared if automation is in scope.

6.4 Test Case Design and Development

- Activities in the Test Case Development Phase:
 - Test Scenarios Identification
 - Test Cases Writing
 - Test Data Preparation



6.4 Test Case Design and Development

- Activities in the Test Case Development Phase:
 - Test Scenarios Identification
 - Test Scenarios ease the testing and evaluation of a complex system.
 - Following strategies help in creating good scenarios:
 - Enumerate possible users, their actions and objectives.
 - Evaluate users with hacker's mindset and list possible scenarios of system abuse.
 - List the system events and how does the system handle such requests.
 - List benefits and create end-to-end tasks to check them.
 - Read about similar systems and their behavior.
 - Studying complaints about competitor's products and their predecessor.

6.4 Test Case Design and Development

- Activities in the Test Case Development Phase:
 - Test Cases Writing
 - A test case is a document which includes test data, preconditions, expected results and post conditions.
 - Developed for a particular test scenario in order to verify compliance against a specific requirement.
 - Test case acts as the starting point for test execution.
 - Features of good test cases:
 - Simple and transparent.
 - Have 100% coverage.
 - Must be identifiable.
 - Should be according to the minds of the clients.
 - Likely to be revised and updated regularly.
 - Likely to be used by developers and developers.

6.4 Test Case Design and Development

- Activities in the Test Case Development Phase:
 - Test Cases Writing
 - Example:
 - Test scenario: Check Login Functionality
 - Test cases:
 - Check results on entering valid User Id & Password.
 - Check results on entering Invalid User ID & Password
 - Check response when a User ID is Empty & Login Button is pressed

6.4 Test Case Design and Development

- Activities in the Test Case Development Phase:
 - Test Data Preparation
 - Used to execute the tests on test platform.
 - Test data should be precise and exhaustive to uncover the defects.
 - This can be created manually or using some automation methods.
 - The coverage of the test data must be high, and it should be updated regularly.

6.5 Test Environment Setup

- Test Environment contains the elements that support test execution with software, hardware and network configured.
- A combination of hardware and software environment to execute the tests.
- This phase includes:
 - Hardware configuration
 - Operating system settings
 - Software configuration
 - Test terminals
 - Other support
- Smoke testing can be used to validate the readiness of the test environment.

6.5 Test Environment Setup

- Activities performed for the Test Environment Setup



- Design Test Environment
 - Check whether archiving is needed to take back-ups
 - Verify the network configuration
 - Identify the required server operating system, databases and other components
 - Identify the number of license required by the test team

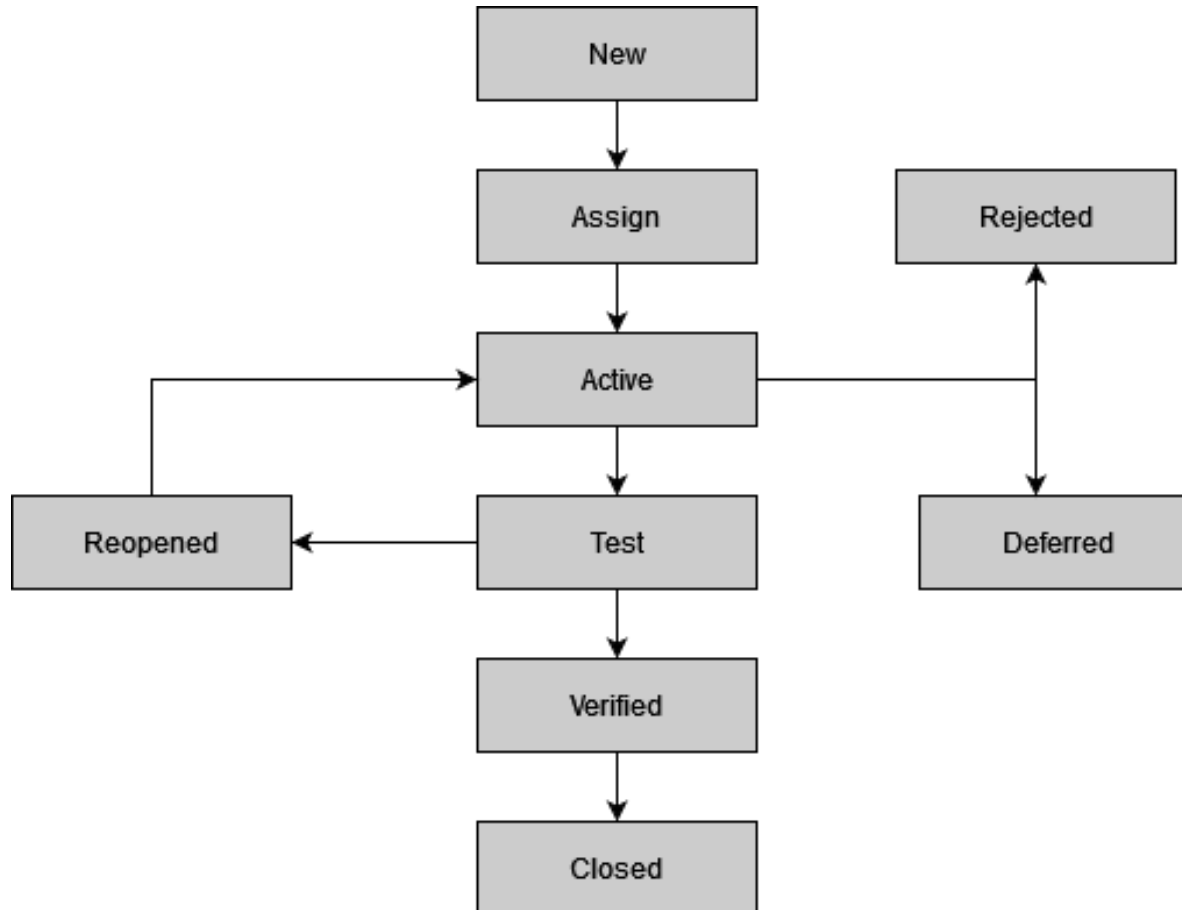
6.5 Test Environment Setup

- Activities performed for the Test Environment Setup
 - Environment Set Up
 - Analysis of setup requirements
 - Preparation of software and hardware requirements for the setup
 - Get official confirmation for setup of the test environment
 - Smoke Testing
 - Must be performed for validation of test environment build stability.
 - It is performed to prevent the needless expense and time waste of deeper tests on a broken product.

6.6 Defect Life Cycle

- Also known as Bug Life Cycle. It is the journey of a defect, the cycle which a defect goes through during its lifetime.
- May vary from organization to organization and from project to project too.
- States of a Defect Life Cycle
 - New
 - Assigned
 - Active
 - Test/Fixed/Ready for Retest
 - Verified
 - Closed
 - Reopened
 - Deferred
 - Rejected

6.6 Defect Life Cycle



Different states of a Defect Life Cycle

6.6 Defect Life Cycle

- States of a Defect Life Cycle
 - New: Potential defect that is raised and yet to be validated
 - Assigned: Defect that is assigned to a development team to be checked
 - Active: The Defect is being addressed by the developer and investigation is under progress.
 - Possible outcomes – deferred or rejected
 - Test/Fixed/Ready for Retest: Defect is fixed and ready for testing
 - Verified: The defect that is retested and the test has been verified by QA.

6.6 Defect Life Cycle

- States of a Defect Life Cycle
 - Closed: The final state of the defect that can be closed after the QA retesting or can be closed if the defect is duplicate or considered as NOT a defect.
 - Reopened: Unfixed defect that is reactivated or reopened by the QA.
 - Deferred: When a defect cannot be addressed in that particular cycle it is deferred to future release.
 - Rejected : A defect can be rejected for any of the three reasons – duplicate defect, NOT a Defect, Non-Reproducible.

6.7 Defects Classification

- QA team perspective – Priority
- Development perspective – Severity (Complexity of code to fix it)
- Priority is the order in which the defects or bugs should be resolved.
- QA team sets the priority status and a time frame to fix the defect based on the end users' requirements.
- Categories are as follows:
 - Low
 - Medium
 - High
 - Urgent

6.7 Defects Classification

- Categories are specified based on the priority of the defects:
 - Low – the defects that can fixed after fixing the critical defects.
 - Medium – these defects should be resolved in the subsequent builds.
 - High – these defects should be solved immediately because the defect affects the application to a considerable extent and the relevant modules cannot be used until it is fixed.
 - Urgent - These defect must be resolved immediately because this kind of a defect affects the application or the product severely and the product cannot be used until it has been fixed.

6.7 Defects Classification

- Severity is the impishness of defect on the application and the complexity of code to fix it from the development perspective.
- It is decided based on how crucial is the defect for the system.
- Severity status gives an idea about the deviation in the functionality due to the defect.
- Severity Listings are as follows:
 - Critical/ Severity 1
 - Major/ Severity 2
 - Medium/ Severity 3
 - Low/ Severity 4

6.7 Defects Classification

- Let us see how these Severity Listings are categorized:
 - Critical/ Severity 1 - Defect impacts most crucial functionality of Application and the QA team cannot continue with the validation of application under test without fixing it.
 - Example: Frequent App/Product crash
 - Major/ Severity 2 - Defect impacts a functional module; the QA team cannot test that particular module but continue with the validation of other modules.
 - Example: flight reservation is not working

6.7 Defects Classification

- Let us see how these Severity Listings are categorized:
 - Medium/ Severity 3 - Defect has issue with single screen or related to a single function, but the system is still functioning. The defect here does not block any functionality.
 - Ticket# is a representation which does not follow proper alpha numeric characters like the first five characters and the last five as numeric.
 - Low/ Severity 4 - It does not impact the functionality. It may be a cosmetic defect, UI inconsistency for a field or a suggestion to improve the end user experience from the UI side.
 - The background color of the Submit button does not match with that of the Save button.

6.8 Test Execution

- Objective is to make sure that there are no any bugs or issues in the application, and it is completely “fit for purpose”.
- Unit testing and integration testing are carried out and the expected and actual results are compared.
- Identified defects or bugs will be corrected and the development continues until all the tests pass.
 - Testers execute the tests, log any defects in the defect tracking tool and communicate the progress to the test manager.
 - Then test status is communicated to management and the development team resolves the identified defects and produce new releases.
 - Then testing team should retest the previously failed test blocks and perform the regression test to confirm that the main functionality is maintained.

6.8 Test Execution

- Entry criteria for this phase:
 - Completion of the Test Plan
 - Completion of Test Cases Development Phase
 - Test data should be available
- Exit criteria for this phase:
 - Successful validation of all the Test cases
 - Defects should be deferred
 - Test case execution
 - Defect Summary Report should be available

6.8 Test Execution

- Main activities in Test Execution phase:
 - System integration testing
 - System Integration Testing (SIT) is a black box testing technique that evaluates the system's compliance against specified requirements/ test cases prepared.
 - System Integration Testing is usually performed on subset of system.
 - This can be performed with minimum usage of testing tools, verified for the interactions exchanged and the behavior of each data field within individual layer is also investigated.
 - After the integration, there are three main states of data flow –
 - Data state within the integration layer
 - Data state within the database layer
 - Data state within the Application layer

6.8 Test Execution

- Main activities in Test Execution phase:
 - Defect reporting
 - A software bug arises when the expected result doesn't match with the actual result. It can be an error, flaw, failure, or fault in a computer program.
 - While performing SIT testing, the QA team finds these types of defects and these need to be reported to the concerned team members.
 - Defect Reporting is a process of finding defects in the application under test or product by testing or recording feedback from customers and making new versions of the product that fix the defects based on the client's feedback.
 - One of the most challenging factors is managing, evaluating and prioritizing these defects.

6.8 Test Execution

- Main activities in Test Execution phase:
 - Defect mapping
 - Once defect is reported and logged, it should be mapped with the concerned failed/blocked test cases and corresponding requirements in Requirement Traceability Matrix.
 - It helps to make a proper defect report and analyze the impishness in product.
 - Once the test cases and requirements are mapped with the defect, stakeholders can analyze and take a decision on whether to fix/defer the defect based on priority and severity.

6.8 Test Execution

- Main activities in Test Execution phase:
 - Re-testing
 - Re-testing is executing a previously failed test against AUT to check whether the problem is resolved. After a defect has been fixed, re-testing is performed to check the scenario under the same environmental conditions.
 - Regression testing
 - Regression testing is a black box testing technique that consists of re-executing those tests that have had an impact due to code changes.
 - Types of Regression Tests:
 - Final Regression Tests – to validate the build that has not changed or modified for a period of time.
 - Regression Tests – to verify if the build has not broken any other parts of the application by the latest code changes

6.9 Test Closure

- Final phase of the Software Testing Life Cycle
- Completeness of the test is assured by the check against a test exit criteria and by the product quality measurement against the test completion criteria.
- Entry criteria for this phase:
 - Execution of the test case is complete
 - Test results are available
 - Defects report is available
- Exit criteria for this phase:
 - Provision of test closure reports
 - Preparation of matrices (to be signed by the clients)

6.9 Test Closure

- Main activities in the Test Cycle Closure phase:
 - Test completion reporting
 - Test completion reporting is a process, whereby the test metrics are reported in summarized format to update the stakeholders.
 - This enables them to take an informed decision.
 - Test Completion Report contains the following information.
 - Test Summary Report Identifier
 - Summary
 - Variances
 - Summary Results
 - Evaluation
 - Planned vs Actual Efforts
 - Sign off

6.9 Test Closure

- Main activities in the Test Cycle Closure phase:
 - Test completion matrix reporting
 - Upon completion of testing, various matrices are collected to prepare the test reports.
 - Availability of Test results articulates - screenshots, database query results, recording, log files
 - While executing a test case, re-testing defects and performing regression test case, **Test results articulate** should be saved.
 - These can be produced along with the test cycle closure documents to support the completion of test execution.

Summary

Requirements Analysis/ Design

- High level requirement analysis
- Scope definition, RTM preparation and Automation analysis

Entry Criteria and Exit Criteria

- Conditions to be completed before starting a phase
- Conditions to be completed before ending a phase

Test Planning

- Defining test specifications
- Test plan

Summary

Test Case Design and Development

- Test cases for individual units

Test Environment Setup

- Preparation of hardware and software environment

Defect Life Cycle

- Cycle defect goes through during its lifetime

Summary

Defects Classification

- Priority
- Severity

Test Execution

- Executing test to ensure there are no bugs
- Ensure that the software is “fit for purpose”

Test Closure

- Final phase
- Test completion criteria