

410905: Software Testing and Quality Assurance

- Dr. Shivani Budhkar

Course Outcomes

- On completion of the course, learner will be able to –
- **CO1:** Illustrate different approaches of quality management, assurance, and quality standard to software system
- **CO2:** Create test plan, test cases and defect repository using case study.
- **CO3:** Apply the concept of white box and block box testing techniques
- **CO4:** Analyse various testing types
- **CO5:** To analyse recent automation tools for software testing.
- **CO6:** Apply software testing automation concepts using Selenium

Course Contents

- Unit I Fundamentals of Software Quality Assurance
- Unit II Essentials of Software Testing
- Unit III Software Testing Techniques
- Unit IV Testing Strategies
- Unit V Software Test Automation
- Unit VI Selenium Tool

Unit- II

Basics of Software Testing

- Dr. Shivani Budhkar

Software Testing Introduction

- What is Software Testing?
 - Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest.
 - Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements.

Why Software Testing?

- We all can make mistakes
- Software bugs could be expensive or even dangerous.
- Software bugs can potentially cause monetary and human loss
- Software testing is really required to point out the defects and errors that were made during the development phases.
- Since it makes sure of the Customer's reliability and their satisfaction in the application.
- To ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
- In order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
- Required for an effective performance of software application or product.
- To ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
- It's required to stay in the business.

Objectives of Software Testing

- Finding defects as early as possible
- To measure the quality of product
- To prevent defects.
- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS (Business Requirement Specification) and SRS(System Requirement Specifications).
- To gain the confidence of the customers by providing them a quality product.
- To have a good test coverage in order to test software application completely

Software testing principles

- Testing shows presence of defects
- Exhaustive testing is not possible
- Early testing
- Defect clustering
- Pesticide paradox
- Testing is context dependent
- Absence of errors fallacy

Verification & validation Concepts

- **Verification** makes sure that the product is designed to deliver all functionality to the customer.
- Verification is done at the starting of the development process.
- It includes reviews and meetings, walk-throughs, inspection, etc. to evaluate documents, plans, code, requirements and specifications.
- It answers the questions like: **Am I building the product right?**
- Am I accessing the data right (in the right place; in the right way).
- It is low level activity
- Performed during development on key artifacts, like walkthroughs, reviews and inspections, mentor feedback, training, checklists and standards.
- Demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

Verification & validation Concepts

Advantages of Software Verification :

- Verification helps in lowering down the count of the defect in the later stages of development.
- Verifying the product at the starting phase of the development will help in understanding the product in a better way.
- It reduces the chances of failures in the software application or product.
- It helps in building the product as per the customer specifications and needs.

Verification & validation Concepts

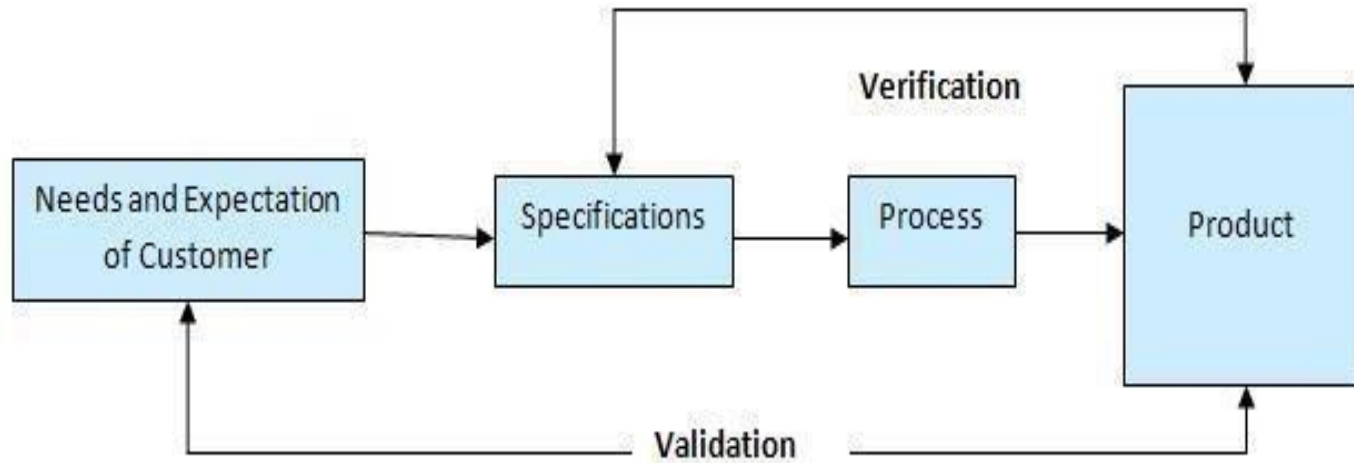
- **Validation is determining** if the system complies with the requirements and performs functions for which it is intended and meets the organization's goals and user needs.
- Validation is done at the end of the development process and takes place after verifications are completed.
- It answers the question like: **Am I building the right product?**
- Am I accessing the right data (in terms of the data required to satisfy the requirement).
- It is a High level activity.
- Performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment.
- Determination of correctness of the final software product by a development project with respect to the user needs and requirements.

Verification & validation Concepts

Advantages of Validation:

- During verification if some defects are missed then during validation process it can be caught as failures.
- If during verification some specification is misunderstood and development had happened then during validation process while executing that functionality the difference between the actual result and expected result can be understood.
- Validation is done during testing like feature testing, integration testing, system testing, load testing, compatibility testing, stress testing, etc.
- Validation helps in building the right product as per the customer's requirement and helps in satisfying their needs.

Verification & validation Concepts

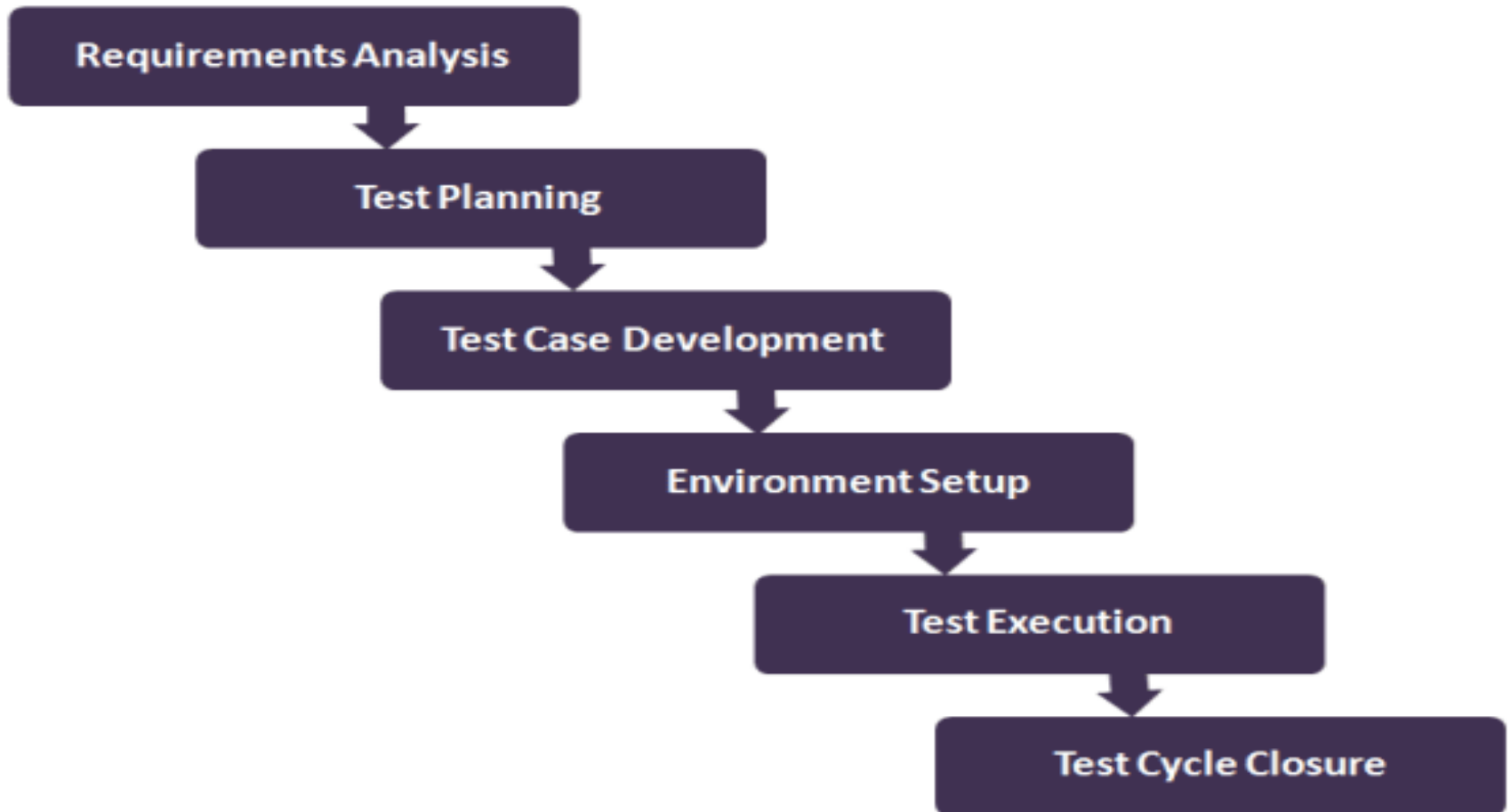


Software Testing life cycle

- Developers follow the Software Development Life Cycle (SDLC) likewise testers also follow the **Software Testing Life Cycle** which is called as **STLC**.
- It is the sequence of activities carried out by the testing team from the beginning of the project till the end of the project.
- It is a testing process which is executed in a sequence, in order to meet the quality goals.
- It consists of many different activities which are executed to achieve a good quality product.
- **Different phases in STLC**
 - Requirement analysis
 - Test Planning
 - Test case development
 - Environment Setup
 - Test Execution
 - Test Cycle Closure

Software Testing life cycle

Software Testing Life Cycle (STLC)



Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Requirement analysis	<p>Availability of Requirement document both Functional as well as non-functional</p> <p>Architectural document of the application or the product should be available</p> <p>Acceptance criteria defined and duly signed by the customers</p>	<p>Analysis of System Requirement specifications to understand the different business modules and it's functionalities</p> <p>To identify the user profile, user interface and user authentication</p> <p>Types of tests to be performed on the application or product should be identified</p> <p>Should collect the details about testing priorities</p> <p>Preparation of RTM that is Requirement Traceability Matrix</p> <p>Test Environment details should be identified in order to do testing</p> <p>Analysis of automation possibility if it is required</p>	<p>RTM should be signed off</p> <p>The customer should sign off on the test automation feasibility</p>	<p>Requirement Traceability Matrix (RTM)</p> <p>Report on Automation Feasibility if it is applicable</p>

Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Test Planning	<p>Detailed requirement document</p> <p>Requirement Traceability Matrix (RTM)</p> <p>Automation Feasibility Report</p>	<p>Preparation of Test Plan document</p> <p>Preparation of Test strategy document</p> <p>To analyze the best suited testing approach for the application or product</p> <p>To analyze the testing techniques and the types of testing to be carried out in order to maintain the quality</p> <p>Selection of the testing tool</p> <p>Estimation on the testing efforts</p> <p>Resource planning as per the skill required for testing and also assigning roles and responsibility to them</p>	<p>Approved Test Plan document</p> <p>Approved Test Strategy document</p> <p>Document of Effort estimation</p>	<p>Test Plan document</p> <p>Test Strategy document</p> <p>Effort estimation document</p>

Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Test case development	Detailed Requirement document	Creation of test cases for all the modules or features in the application or product	Reviewed Test cases	Test cases Test automation scripts Test data
	Test Plan and Test strategy documents		Reviewed Test automation scripts	
	Automation Feasibility Report	Creation of automation scripts if required	Test data creation ready for testing	
		Review of test cases and test automation scripts		
		Test data creation		

Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Test Environment setup	<p>System design documents should be available</p> <p>Architectural document of the application should be available</p> <p>Environment set-up plan document should be available</p>	<p>Understanding the design and architecture of the application</p> <p>Setting up the test environment</p> <p>Installation of required hardware and software in order to start testing the application</p> <p>Integration of any third party application (if required)</p> <p>Installation of build</p> <p>Creation of test data</p> <p>Execution of smoke testing on the build</p> <p>Accepting or rejecting the build as per the smoke test result</p>	<p>Environment setup is ready for testing</p> <p>All the required software and hardware are installed</p> <p>Build installation is complete and successful</p> <p>Test data creation is complete</p> <p>Smoke testing is done</p>	<p>Test environment along with test data</p> <p>Smoke test result</p>

Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Test Execution	<p>Documents like RTM, Test Plan, Test strategy, Test cases and Test scripts should be ready</p> <p>Test environment should be ready</p> <p>Test data should be ready</p> <p>Integration of third party application (if required) should be successful</p> <p>Smoke testing of the application should be successful</p>	<p>Execution of test cases</p> <p>Preparation of test result document</p> <p>Logging defects for the failed test cases</p> <p>Mapping of defects with the test cases</p> <p>To update the test cases and test strategy if required</p> <p>Fixed defects should be retested</p> <p>Closure of the defects if they are working as expected</p> <p>Execution of regression testing of the application or product in order to ensure its stability post defect closure</p>	<p>All test cases are executed</p> <p>Defects are logged and tracked for closure</p>	<p>Completed the test case execution</p> <p>Updated the test cases wherever required</p> <p>Defects reported</p>

Software Testing life cycle

STLC phases	Entry Criteria	Activity	Exit Criteria	Deliverables (Outcome)
Test cycle closure	All the test cases are executed and updated	Evaluation of the test completion on the basis of Test Coverage and Software Quality	Signed off Test Closure report by the client	Test closure Report
	Test results are documented	Preparation of Test Closure report		
	Defect logs are available	Analyzing the test results to find out the distribution of severe defects		

The tester's role in a software development organization

- Testing erroneously viewed as a destructive activity.
- The tester's job is to reveal defects, find weak points, inconsistent behaviour, and circumstances where the software does not work as expected.
- Given the nature of the tester's tasks, it is difficult for developers to effectively test their own code.
- To be most effective as a tester requires extensive programming experience in order to understand how code is constructed, and where and what kind of, defects are likely to occur.
- Goal as a tester is to work with the developers to produce high-quality software that meets the customers' requirements.
- Teams of testers and developers are very common in industry, and projects should have an appropriate developer/tester ratio.

The tester's role in a software development organization

- The ratio will vary depending on available resources, type of project, and TMM level.
- For example, an embedded real time system needs to have a lower developer/tester ratio (for example, 2/1) than a simple data base application (4/1 may be suitable).
- At higher TMM levels where there is a well-defined testing group, the developer/tester ratio would tend to be on the lower end (for example 2/1 versus 4/1) because of the availability of tester resources.
- The nature of the project and project scheduling issues would impact on the ratio.
- In addition to cooperating with code developers, testers also need to work alongside with requirements engineers to ensure that requirements are testable, and to plan for system and acceptance test (clients are also involved in the latter).

The tester's role in a software development organization

- Testers also need to work with designers to plan for integration and unit test.
- In addition, test managers will need to cooperate with project managers in order to develop reasonable test plans, and with upper management to provide input for the development and maintenance of organizational testing standards, policies, and goals.
- Finally, testers also need to cooperate with software quality assurance staff and software engineering process group members.
- In view of these requirements for multiple working relationships, communication and team working skills are necessary for a successful career as a tester.
- Marketing staff need to realize that testers add value to a software product in that they detect defects and evaluate quality as early as possible in the software life cycle.
- This ensures that developers release code with few or no defects, and that marketers can deliver software that satisfies the customers' requirements, and is reliable, usable, and correct.

The tester's role in a software development organization

- Low- defect software also has the benefit of reducing costs such as support calls, repairs to operational software, and ill will which may escalate in to legal action due to customer dissatisfaction.
- In view of their essential role, testers need to have a positive view of their work. Management must support them in their efforts and recognize their contributions to the organization.