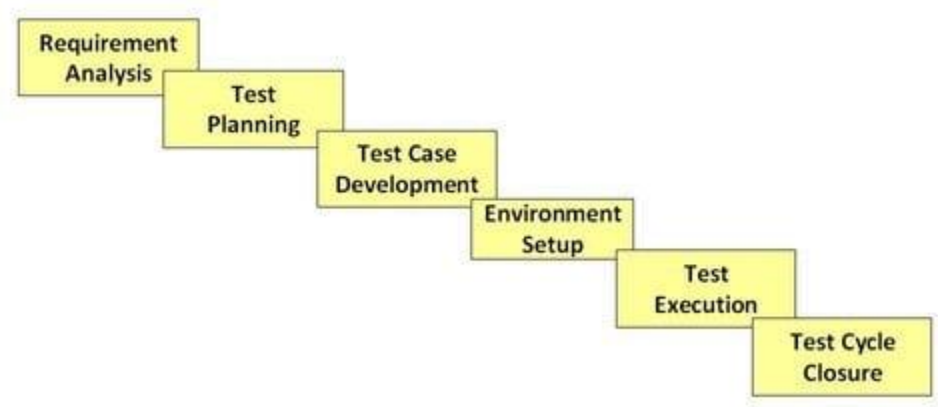
-- What is Software Testing Life Cycle (STLC)?

Software Testing Life Cycle (STLC) is a sequence of specific activities conducted during the testing process to ensure software quality goals are met. STLC involves both verification and validation activities.

-- STLC Phases

There are **six major** phases in every Software Testing Life Cycle Model (STLC Model):



- - What is Entry and Exit Criteria in STLC?

- **Entry Criteria:** Entry Criteria gives the prerequisite items that must be completed before testing can begin.
- **Exit Criteria:** Exit Criteria defines the items that must be completed before testing can be concluded

We have **Entry and Exit** Criteria for all levels in the Software Testing Life Cycle (STLC)

In an Ideal world, we will not enter the next stage until the exit criteria for the previous stage is met. But practically this is not always possible.

-- Requirement Phase Testing

Requirement Phase Testing, also known as Requirement Analysis in which test team studies the requirements from a testing point of view to identify testable requirements and the QA team may interact with various stakeholders to understand requirements in detail. Requirements could be either functional or non-functional. Automation feasibility for the testing project is also done in this stage.

Activities in Requirement Phase Testing

- **Identify Testable Requirements** – Review software requirements to determine what can be tested.
- **Identify Testing Types** – Decide whether functional, performance, security, etc., testing is needed.
- **Traceability Matrix (RTM) Creation** – Map test cases to requirements for full coverage.
- **Stakeholder Discussion** – Collaborate with developers and business analysts to clarify doubts.

Deliverables of Requirement Phase Testing

- RTM
- Automation feasibility report. (if applicable)

-- Difference between functional and non-functional requirements

Functional Requirements

Functional requirements describe ***what a system should do****. They define the system's behavior, features, and functions based on user needs.*

Examples:

- *User authentication (login/logout)*
- *Data validation (e.g., email format checking)*
- *Payment processing*
- *Report generation*
- *Role-based access control*

Non-Functional Requirements

Non-functional requirements specify ***how a system should perform****. They focus on quality attributes and constraints rather than specific behaviors.*

Examples:

- *Performance (response time should be <2 seconds)*
- *Security (data encryption, two-factor authentication)*
- *Usability (intuitive UI)*
- *Reliability (99.99% uptime)*
- *Scalability (support 10,000 concurrent users)*

In simpler terms: *Functional requirements are about the system's capabilities, while non-functional requirements are about its qualities.*

-- Test Planning in STLC

Test Planning in STLC is a phase in which a Senior QA manager determines the test plan strategy along with efforts and cost estimates for the project. Moreover, the resources, test environment, test limitations and the testing schedule are also determined. The Test Plan gets prepared and finalized in the same phase.

Test Planning Activities

- **Define Test Strategy** – Identify scope, types, and levels of testing.
- **Estimate Effort & Resources** – Determine time, team members, and tools required.
- **Define Entry & Exit Criteria** – Set conditions for when testing starts and ends.
- **Risk Analysis & Mitigation** – Identify possible risks and plan how to handle them.
- **Test Plan Documentation** – Prepare a formal test plan for approval.

Deliverables of Test Planning

- Test plan/strategy document.
- Effort estimation document.

-- Test Case Development Phase

The **Test Case Development Phase** involves the creation, verification and rework of test cases & test scripts after the test plan is ready. Initially, the Test data is identified then created and reviewed and then reworked based on the preconditions. Then the QA team starts the development process of test cases for individual units.

Test Case Development Activities

- **Write Test Cases** – Create step-by-step instructions to validate functionality.
- **Prepare Test Data** – Identify required input data for testing.
- **Review & Validate** – Ensure test cases meet requirements and cover edge cases.
- **Create Test Scripts (For Automation)** – Write automation scripts if applicable

Deliverables of Test Case Development

- Test cases/scripts
- Test data

-- Test Environment Setup

Test Execution is the phase in the Software Testing Life Cycle (STLC) where actual testing is performed based on the prepared test cases.

Test Environment Setup Activities

- **Set Up Required Software & Hardware** – Install the application and necessary tools.
- **Configure Test Data** – Ensure data is available for different test scenarios.
- **Check Environment Readiness** – Validate database, network, and system settings.
- **Test Environment Validation** – Perform a smoke test to check if everything works before test execution.

-- Test Execution Phase

Test Cycle Closure is the final phase of the Software Testing Life Cycle (STLC), where the testing process is formally completed and evaluated.

Test Execution Activities

- **Test Case Execution –** Run test cases manually or using automation tools.
- **Compare Expected vs. Actual Results –** Verify if the software behaves as expected.
- **Defect Reporting –** Log defects in a bug tracking tool if actual results don't match expected results.
- **Defect Retesting –** After developers fix bugs, testers re-execute the failed cases.
- **Regression Testing –** Ensure new changes don't break existing functionality.
- **Test Execution Report** – Track pass/fail status, defects found, and test coverage.

-- Test Cycle Closure

Test Cycle Closure phase is completion of test execution which involves several activities like test completion reporting, collection of test completion matrices and test results. Testing team members meet, discuss and analyze testing artifacts to identify strategies that have to be implemented in future, taking lessons from current test cycle. The idea is to remove process bottlenecks for future test cycles.

Test Cycle Closure Activities

- **Test Summary Report** – Document test results, defect statistics, and overall test coverage.
- **Defect Analysis** – Review resolved and unresolved defects.
- **Test Case & Script Archiving** – Store test cases for future reference.
- **Lessons Learned** – Identify what worked well and what needs improvement.
- **Test Closure Meeting** – Discuss findings with stakeholders and ensure all objectives are met.
- **Sign-Off** – Obtain approval from stakeholders to officially close testing.