

Software Development Lifecycle

SDLC (Software Development Life Cycle) defines all the standard phases involved in the software development process, while STLC (Software Testing Life Cycle) defines various activities to improve the quality of the product.

- SDLC is a development life cycle, whereas STLC is a testing life Cycle.
- In SDLC, the development team creates high and low-level design plans, whereas in STLC, the test analyst creates the system and integration test plans.
- SDLC involves the actual development of code and implementation based on design documents, whereas in STLC, the testing team prepares the test environment and executes test Cases.
- SDLC covers the complete software development process, while STLC focuses only on software testing.
- SDLC consists of phases such as requirements gathering, design, build, test, deployment, and maintenance, with specific entry and exit criteria for each stage.
- STLC includes phases such as requirement analysis, test planning, test development, test environment setup, test execution, and closure.
- SDLC aims to produce a high-quality software system, meet customer expectations, and create considerable system documentation.
- STLC helps make the testing process more sophisticated, consistent, and effective by including milestones and deliverables for each step of the project.
- SDLC is suitable for projects where requirements are clear and can be completed step by step, while STLC focuses on testing and detecting errors rather than development itself.

SOFTWARE TESTING LIFECYCLE

STLC is a systematic approach to testing software to ensure that quality objectives are met. It consists of several phases and activities that are carried out in a methodical manner.

1. Analysis of Requirements:

- Determine testable requirements from a testing perspective.

- Gather information about testing priorities and emphasis.
- Create a Requirement Traceability Matrix (RTM) to map requirements to test cases.
- Analyze the feasibility of automation.
- Deliverables: RTM, report on automation feasibility.

2. Test Preparation:

- Prepare a test plan/strategy document for various types of testing.
- Select a test tool.
- Estimate the test effort.
- Organize resources and identify roles and responsibilities.
- Deliverables: Test plan/strategy document, effort estimation document.

3. Development of Test Cases:

- Design, verify, and revise test cases and test scripts.
- Select and prepare test data.
- Review and baseline test cases and scripts.
- Deliverables: Test cases and scripts, test data.

4. Setup of the Test Environment:

- Determine the required hardware and software environment.
- Set up the test environment and test data.
- Conduct a smoke test to ensure readiness.
- Deliverables: Test data setup, results of the smoke test.

5. Execution of the Test:

- Execute the tests according to the test plan.

- Document test results and log any failed instances as defects.
- Map defects to test cases in the RTM.
- Retest the fixes for defects and follow up until resolution.
- Deliverables: Completed RTM, updated test cases, defect reports.

6. Closure of the Test Cycle:

- Evaluate cycle completion criteria based on time, test coverage, cost, software quality, and business objectives.
- Prepare test metrics based on the evaluation.
- Document lessons learned from the project.
- Prepare a test conclusion report.
- Report the quality of the work output to the client.
- Analyze test results to determine defect distribution.
- Deliverables: Test completion report, testing metrics.