The Software Testing Life Cycle (STLC). Your Step by Step Guide to Quality

👤 Rashmi Ruwanthika Athukorala   4 min read · 1 hour ago

👏   💬   🔖   ▶   ⬆   •••

In software development, creating an application is not enough it must work correctly, be user-friendly, and have fewer bugs. That's where the **Software Testing Life Cycle (STLC)** comes in.

The STLC is a **step-by-step process** that guides testers on **how to plan, design, execute, and finish testing** properly. It helps ensure that software is tested in an organized and effective way, rather than randomly checking for bugs.

💡 STLC vs. SDLC: What's the Difference?

Many people get confused between **SDLC** and **STLC**, but they are not the same.

Let's make it simple

- **SDLC (Software Development Life Cycle)** is the **overall process** of developing software.
  It includes requirement gathering, design, development, testing, deployment, and maintenance.
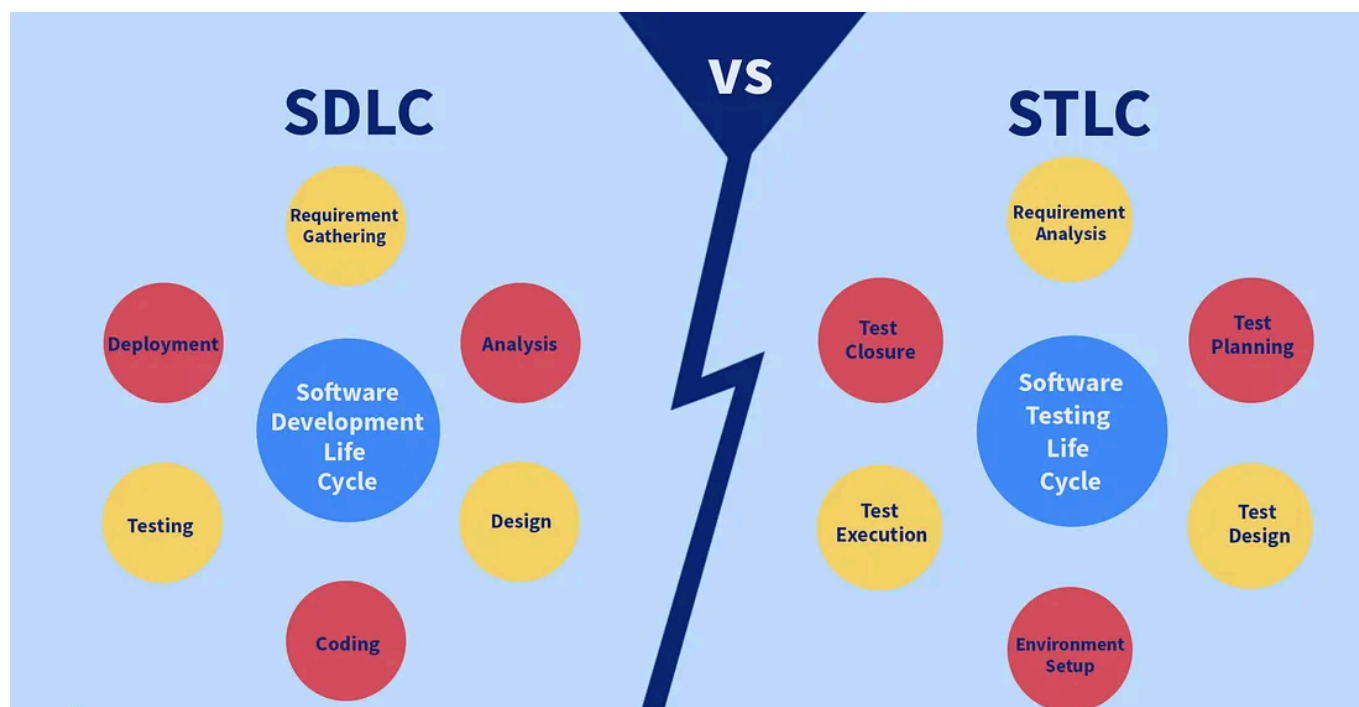 *In short: SDLC = Building the software.*

- **STLC (Software Testing Life Cycle)** is a **part of SDLC** that focuses only on the **testing process**.
  It includes planning tests, writing test cases, setting up the environment, running tests, and closing testing.
  In short: STLC = Testing the software to make sure it works well.

Example:
Think of SDLC like **baking a full cake** , and STLC is the process of **checking the taste, texture, and quality** before serving it.

The 6 Phases of the STLC (Explained in Simple Words)

The STLC has **six important phases,** and each one has a clear starting point (entry criteria) and ending point (exit criteria). This helps the team move step by step without missing anything.

❌ 1. Requirement Analysis Phase

This is the **first and most important** phase of testing.
Here, the QA team carefully studies the **software requirements** given by clients or business analysts.

Goal:
Understand *what needs to be tested* and *what the user expects from the system.*

Activities in this phase:

Read and analyze requirement documents.

Identify which features can be tested.

Ask questions to clarify doubts.

Find missing or unclear requirements.

Main Output:
Requirement Traceability Matrix (RTM) — a document that links each requirement to its test case to ensure nothing is missed.

2. Test Planning Phase

Once the requirements are clear, the next step is to plan *how the testing will be done.*

Goal:
Create a clear **testing strategy** and decide the testing method, tools, and timeline.

Activities in this phase:

Prepare the ***Test Plan document****.*

Decide whether to use ***manual testing, automation testing,*** *or both.*

Select testing tools (like Selenium, JIRA, etc.).

Estimate effort, cost, and time needed for testing.

Assign responsibilities to the team.

Main Output:
An approved **Test Plan** and estimation sheet that guides the whole testing process.

3. Test Case Development Phase

This is where testers start writing **test cases** based on the requirements.

Goal:
Create **step-by-step instructions** for testing each feature of the application.

Activities in this phase:

> *Write detailed test cases.*
>
> *Create **test data** that will be used during testing.*
>
> *Review test cases with peers to ensure accuracy.*
>
> *If automation is used, start creating test scripts here.*

Main Output:

A complete set of **reviewed and approved test cases** and **test data** ready to use.

4. Test Environment Setup Phase

In this phase, the environment (software, hardware, and tools) required for testing is prepared.

Goal:

Set up a **testing environment** that is similar to the real production environment.

Activities in this phase:

Install necessary software, databases, and servers.

Configure browsers, operating systems, or devices.

*Perform a **smoke test** (a quick basic test) to check that the system is ready for full testing.*

Main Output:
A **stable test environment** and a successful **smoke test result**.

5. Test Execution Phase

Now, the real action happens! Testers execute the test cases to check if the software works as expected.

Goal:
Find bugs or defects by running test cases.

Activities in this phase:

Run manual and automated test cases.

*Compare **expected results** vs **actual results**.*

*If something fails, create a **defect report** with clear details.*

*Developers fix the issue, and testers perform **re-testing**.*

*Perform **regression testing** to confirm old features are not broken.*

Main Output:
Defect reports, **test results**, and an updated **RTM** showing pass/fail status.

6. Test Cycle Closure Phase

This is the **final phase** of the STLC. After testing is completed, the QA team reviews everything and documents the results.

Goal:
Summarize the testing process, results, and lessons learned.

Activities in this phase:

- Prepare a **Test Closure Report**.

- Share overall testing results, number of bugs found, fixed, and pending.

- Conduct a **retrospective meeting** to discuss what went well and what can be improved.

- Archive all test documents for future reference.

Main Output:
A **Test Closure Report** and improvement points for future projects.

The Role of Automation in STLC

Automation testing plays an important role in modern testing.

Where it fits:

- During **Test Case Development** (to write automated scripts).

- During **Test Execution** (to run scripts automatically).

When to Automate:

- Repetitive test cases (like regression testing).

- Smoke and sanity tests.

- Performance and load tests.

Benefits of Automation:

- Saves time by running tests faster.

- Reduces human errors.

- Gives quick feedback on every new build (especially in Agile and CI/CD environments).

Final Summary

The **Software Testing Life Cycle (STLC)** helps QA teams follow a proper path for testing.
It ensures every requirement is tested, defects are caught early, and software quality is maintained.

Phase	Purpose	Main Output
Requirement Analysis	Understand what to test	RTM
Test Planning	Plan how to test	Test Plan
Test Case Development	Write test cases	Reviewed Test Cases
Test Environment Setup	Prepare environment	Validated Setup
Test Execution	Run tests & report bugs	Defect Reports, RTM
Test Closure	Summarize results	Closure Report

STLC gives a **clear roadmap** to testers so that testing is not random it's **structured, complete, and effective.**

Written by Rashmi Ruwanthika Athukorala

1 follower · 1 following

BSc Hons Computer Science and Technology | Undergraduate at Uva Wellassa University, Sri Lanka | QA Enthusiast | Software Tester

Edit profile

No responses yet

Rashmi Ruwanthika Athukorala

What are your thoughts?