

What is Software Testing Life Cycle (STLC)

What is STLC (Software Testing Life Cycle)

The Software Testing Life Cycle (STLC) is a sequence of specific actions performed during the testing process to ensure that the software quality objectives are met. The STLC includes both verification and validation. Contrary to popular belief, software testing is not just a separate activity. It consists of a series of methodological activities to help certify your software product.

The STLC is a high-quality strategy directly associated with and part of the Software Development Life Cycle (SDLC), which in turn is a framework with 6 core principles:

1. Requirements analysis;
2. Planning;
3. Engineering and design;
4. Software development;
5. Testing;
6. Deployment.



In simple terms, the SDLC is a cycle where actions at each stage are reflected in subsequent stages.

The STLC also has its stages and most closely intersects with the SDLC at the fifth stage, which I'd describe below.

What to use the STLC for

By knowing that the STLC is a set of measures, we can assume that it includes various stages, such as planning, control, implementation, standardization, and so on. All this leads us to the fact that the STLC is needed not only to test the developed product but also for the following:

- Removing its shortcomings during the earliest and most profitable stage of development;
- Increasing the quality and transparency of the development process;
- Maximizing control over the quality of the product being developed at all stages of the SDLC;
- Influencing the application of Agile, Scrum, SAFe, etc.;
- Delivering a quality product not only to the Client but also to the Users.

The role of the STLC in the SDLC

As mentioned earlier, the Software Testing Life Cycle and the Software Development Life Cycle are closely related to each other, but they simultaneously pursue different tasks of the same goal, namely:

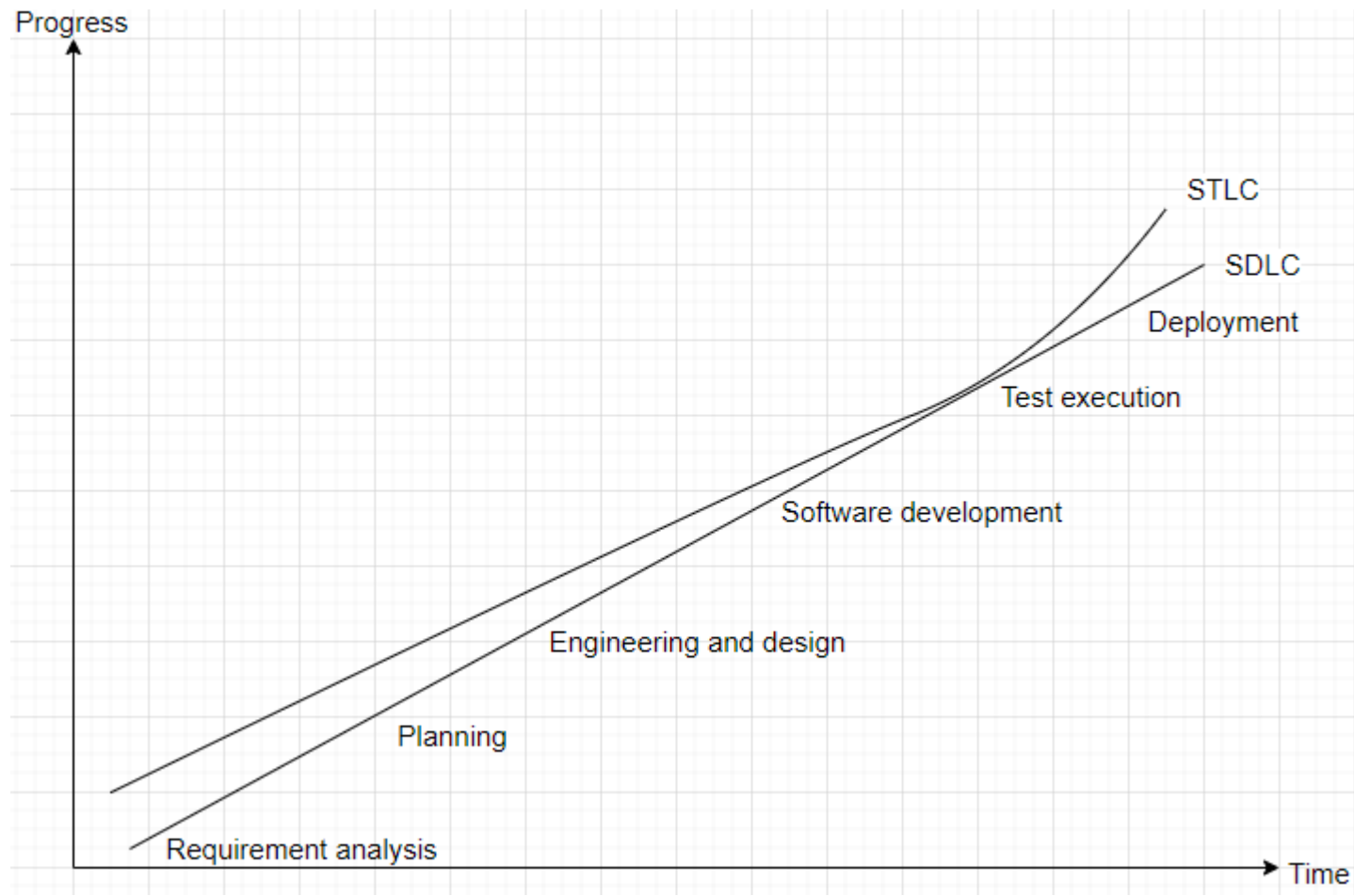
- collecting the requirements in the desired form and developing the declared functionality (as for the SDLC);



- analyzing requirements, assisting the client and the development team, and confirming the quality of the implemented functionality (as for the STLC).

The overall goal is client satisfaction and achieving the highest possible score at the Verification and Validation stages.

The role of the STLC in the SDLC can be represented with a double line graph:

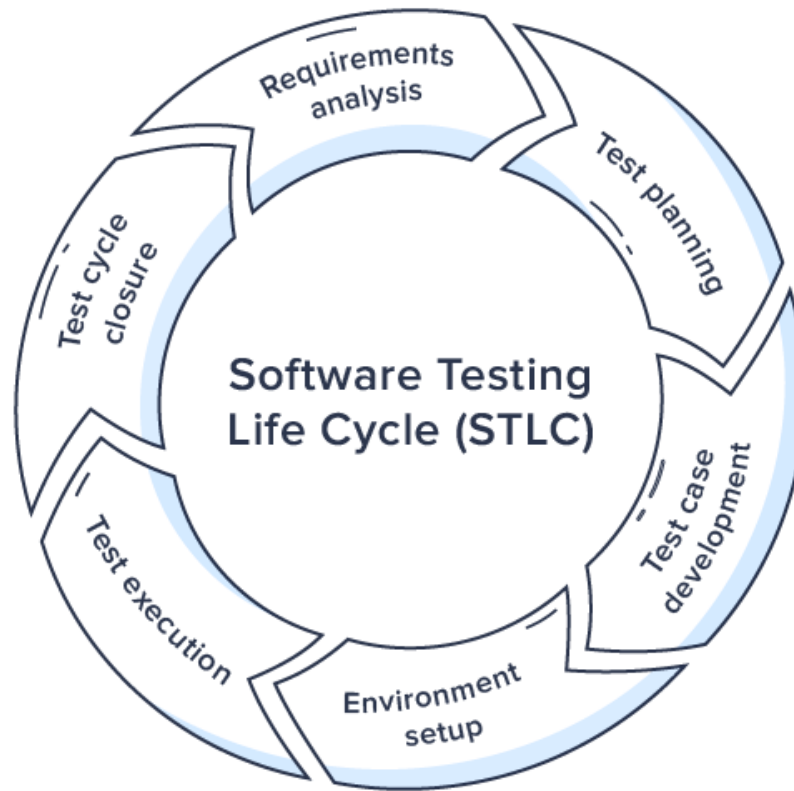


Where the SDLC and the STLC lines move in parallel for most of the project but begin to converge rapidly during the software development phase with deep synchronization during the SDLC Test phase. This graph is relevant to many different types of projects, not just large or independent ones, and will remain the same for the implementation of a task within a project and reverse for a project with a huge number of iterations (but in this case, the lines will diverge and converge more often).

What are the STLC phases

The STLC has several interconnected phases and is generally very similar to the SDLC system. These phases are sequential and are called:

1. Requirement Analysis;
2. Test Planning;
3. Test Case Development;
4. Test Environment Setup;
5. Test Execution;
6. Test Cycle Closure.



bamboo agile

Let's take a closer look at each of them.

Requirement Analysis

Requirements analysis is one of the most important phases because this is when you can fix project flaws almost completely free of charge. The requirements analysis phase also identifies the potential

need for automated testing and allows making economic calculations of labor costs based on the project estimation. This is also when the Entry Criteria and the Exit Criteria are discussed and documented.

Test Planning

At this phase, a test plan is formed. This is the concretization of all phases of the testing itself, timing, participants, and responsibilities. As a result of this, we receive data on:

- the participants and their roles in testing;
- the necessary testing tools;
- the necessary test environment.

Test Case Development

Test development implies using manual and automated testing to achieve full coverage of the software's functionality, with the process being based on the requirements set up beforehand. More often than not, the test cases for automated testing are written separately, since the cases for manual testing are described in the form of cheat sheets.

Test Environment Setup

The test plan makes it clear which test environment should be used. At this STLC phase, operating systems and virtual machines are configured, testing tools like Selenium, Katalon Studio, and the project's test environment and databases are deployed. We also make requests to DevOps and Admins if support is needed.

Test Execution

Tests are performed based on ready-made test documentation and a correctly configured test environment. All test results are recorded in the Test Management System. Negatively passed tests, where the actual result differs from the one expected, are recorded as errors and transferred to the development team for revision with subsequent rechecking after correction.

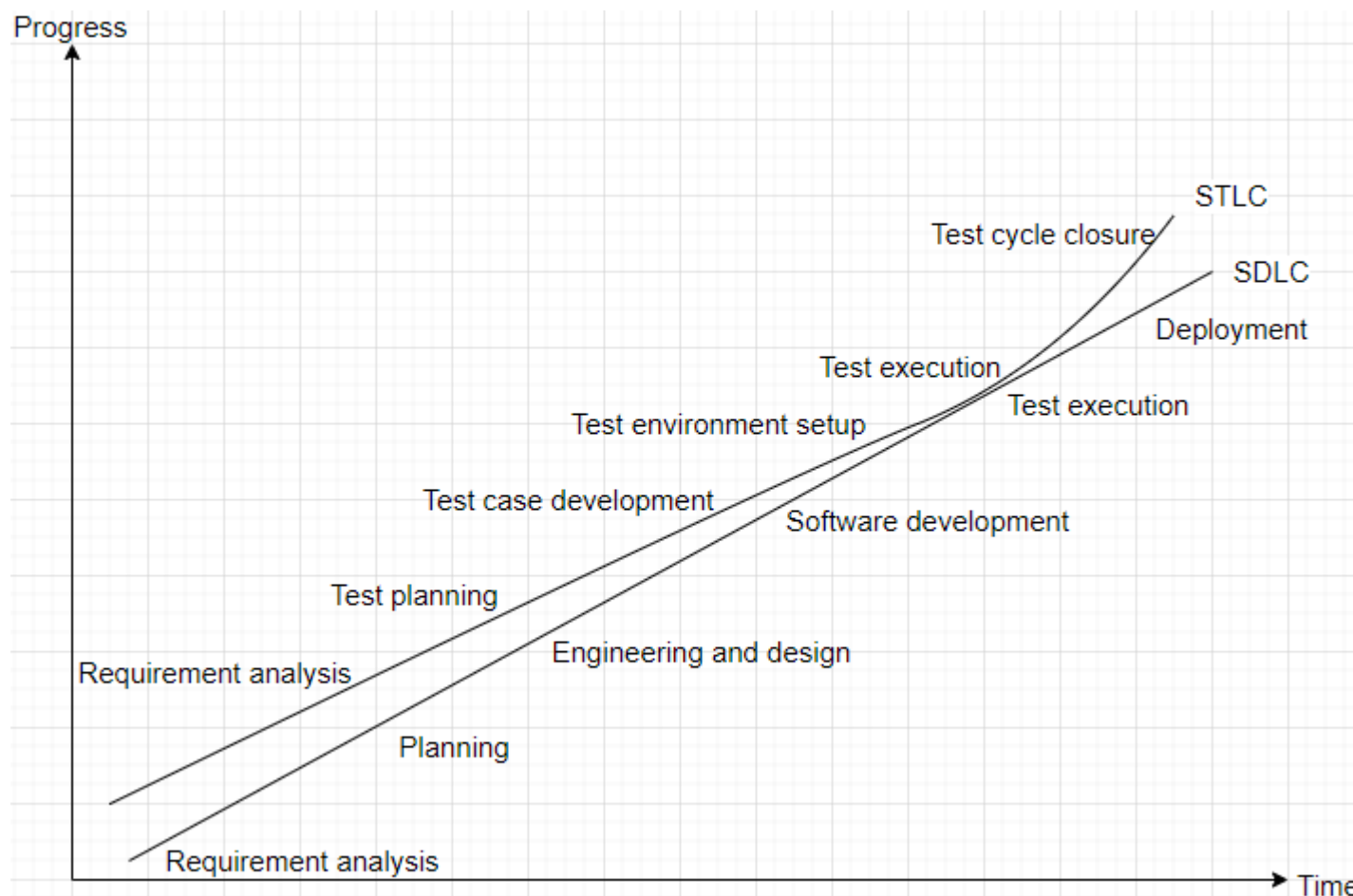
Test Cycle Closure

The last phase of the STLC is the final generation of testing reports for the client. These should include the time spent, the percentage of errors found to positive test results, the total number of errors found and fixed. As far as the testing department is concerned, this is the moment for analyzing its work, summing up the results, analyzing its productivity, and the opportunity to make suggestions for improving the quality of testing.

Conclusion

Now that we know what the STLC is and what it is for, we can say what the graph of interaction between the STLC and the SDLC will look like with full certainty:





The fact that the closest point of convergence happens during the “Test Execution” stage is direct proof of the importance of testing in software development. At the same time, we see that some phases of the STLC and the SDLC are very similar to each other both in name and in meaning, which suggests that the Software Testing Life Cycle and the Software Development Life Cycle are inevitably linked at all stages of software product development.