**Task-2**

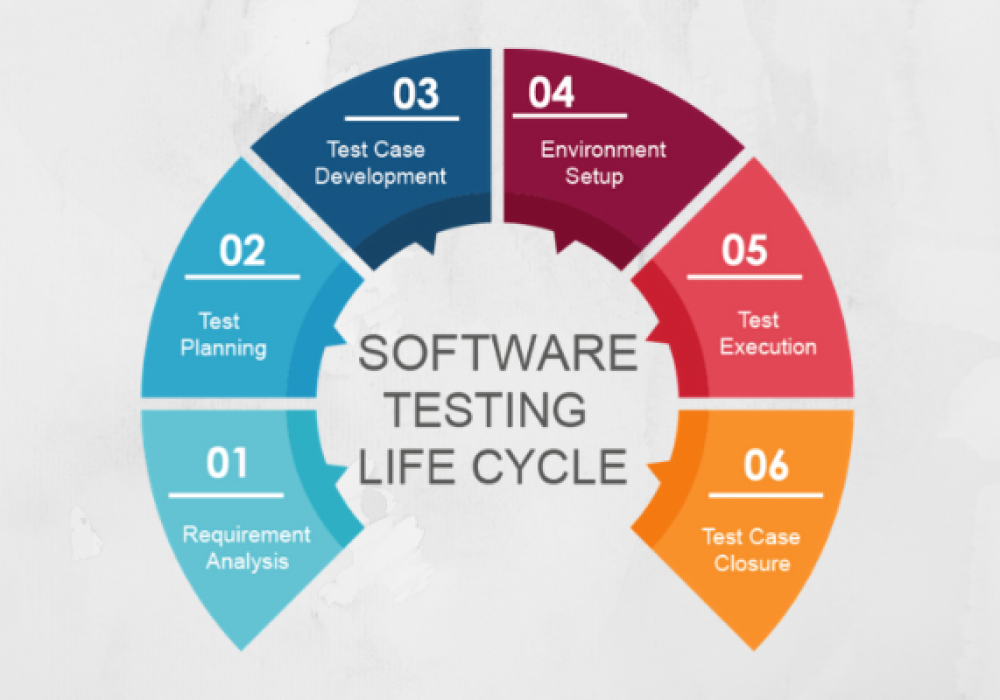
**Software testing life cycle - Benefits & Phases**

# **Software testing life cycle**

# Software Testing Life Cycle (STLC) is a testing strategy that can help you efficiently meet software quality standards. STLC enforces systematic testing, which is performed in phases. STLC is often confused with Software Development Life Cycle (SDLC), but while STLC is focused on testing, SDLC encompasses all development stages. Read on for an in-depth look at STLC and its six phases. **Benefits of Software Testing Life Cycle (STLC)**

Some of the STLC benefits include the following:

* Increased consistency and effectiveness as project requirements are analysed.
* Clear and defines goals for the product under test, which helps track the project process much easier.
* The confidence in each feature being passed testing, before additional features are added.
* Tests to be designed in a meaningful manner.
* Specifications are much clearer which in turn helps the entire team too.
* It’s a systematic approach that allows uncovering bugs/defects in the product much faster.



# **Phases of Software Testing Life Cycle (STLC)**

# **Phase 1: Requirement Analysis**

In the first phase, the STLC team understands the feature requirements, mainly what is there to be a tester? If necessary, the [testing teams](https://blog.testproject.io/2020/01/29/the-first-selenium-framework-for-team-collaboration-testproject/" \t "https://laveenaramchandani01.medium.com/_blank) can also consult at this point with stakeholders to gain more clarity on the requirements. The requirements can be functional/non-functional and the decision to [automate testing](https://testproject.io/" \t "https://laveenaramchandani01.medium.com/_blank) is also evaluated at this point.

* Entry Criteria: client requirements, acceptance criteria, and intended product architecture to be documented.
* Exit Criteria: aim for a requirement traceability matrix (RTM) and decision of automation.

# **Phase 2: Test Planning**

This phase includes a test strategy being implemented and defined in the testing plan. Also, the effort and cost of the testing team are estimated at this stage. This stage will only commence once the requirement gathering is completed.

* Entry Criteria: report the test strategy being used.
* Exit Criteria: have an approval on the test plan ( risks and costs).

# **Phase 3: Test Case Development**

During this phase, test cases are created. Each case defines test inputs (e.g. data), procedures, execution conditions, and anticipated results. Test cases should be transparent, efficient, and adaptable. Once all test cases are created, test coverage should be 100%. Any necessary [automation scripts](https://testproject.io/" \t "https://laveenaramchandani01.medium.com/_blank) are also created during this phase.

* Entry Criteria: approval of timelines to execute the test plan.
* Exit Criteria: approval of test cases and automation scripts if being used.

# **Phase 4: Test Environment Setup**

In the fourth phase, a [QA/Testing environment](https://blog.testproject.io/2020/08/26/the-importance-of-a-qa-environment/" \t "https://laveenaramchandani01.medium.com/_blank) is configured and deployed to allow testers to test the feature. Once the environments are successfully deployed, [smoke testing](https://blog.testproject.io/2020/05/31/smoke-testing-vs-sanity-testing/" \t "https://laveenaramchandani01.medium.com/_blank) can be performed to ensure the environment is working with the intended functionality and would also provide confidence in testing the new feature.

* Entry Criteria: have a system design and project architecture defined.
* Exit Criteria: have a working QA/Testing environment set up to run test cases.

# **Phase 5: Test Execution**

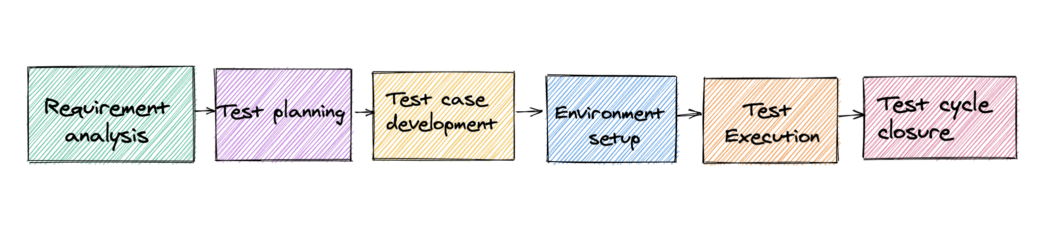
In the fifth stage, the test execution gets started as the environment is successfully set up. The test cases generated at phase three are put into action here. Your expected results can be used to compare the actual results. The actual results could also be used as a baseline. You can also report back all your interesting findings such as [bugs reports](https://blog.testproject.io/2020/11/02/tricks-to-write-a-good-bug-report/" \t "https://laveenaramchandani01.medium.com/_blank) to the development teams.

* Entry Criteria: have all the previous steps completed especially the exit criteria.
* Exit Criteria: test execution and results to be documented to report them back.

# **Phase 6: Test Cycle Closure**

In the last stage of the STLC, a [test result report](https://testproject.io/executive-test-reports/" \t "https://laveenaramchandani01.medium.com/_blank) is generated. The report should contain the entire process of testing the new requirement such as the analysis between the expected outcome and the actual outcome, whether the objectives were met or not, the time taken to test the feature, costs, test coverage and the most important thing of all — whether you found any defects and details about it.

* Entry Criteria: test results and logging from all previous phases.
* Exit Criteria: delivered planned deliverables and approved test closure test summary report.



# **STLC vs. SDLC: How Do They Differ?**

Although the STLC and SDLC (Software Development Life Cycle) might seem related, they are different in their own nature. They both have different aims and guidelines. Something quite interesting is that STLC phases can be performed in the phases of SDLC. SDLC is related to software ****development**** and STLC is related to software ****testing****. STLC is a segment/subset/part of SDLC. The activities in both lifecycles are executed one after the other.

Another difference I can point out between both approaches is that with SDLC the primary goal is to collect the client requirements and create the features accordingly. However, STLC’s goals are to generate tests that adapt to the requirements and verify that the features meet the requirements.

The goal of SDLC is to complete a successful software development with good quality software, whereas STLC aims to complete successful software testing activities and making the software defect free. Both STLC and SDLC phases have entry and exit criteria to be fulfilled before entering or leaving a phase as mentioned above.

In both of these approaches, you need your stakeholders such as DevOps, testing team, Product owners, Business analysts and developers. Full cooperation from the team means you can verify the requirements are defined accurately, tests are relevant, and the results are accurate too!

