

SOFTWARE TESTING LIFE CYCLE

People make software, and they make mistakes. Unfortunately, mistakes in software development can become too expensive. That is why testing deserves its own phase in the SDLC.

A software testing lifecycle (STLC) is a process within the software development lifecycle that helps ensure the highest possible quality of the end product and its adherence to client and industry requirements. Its main steps are as follows:

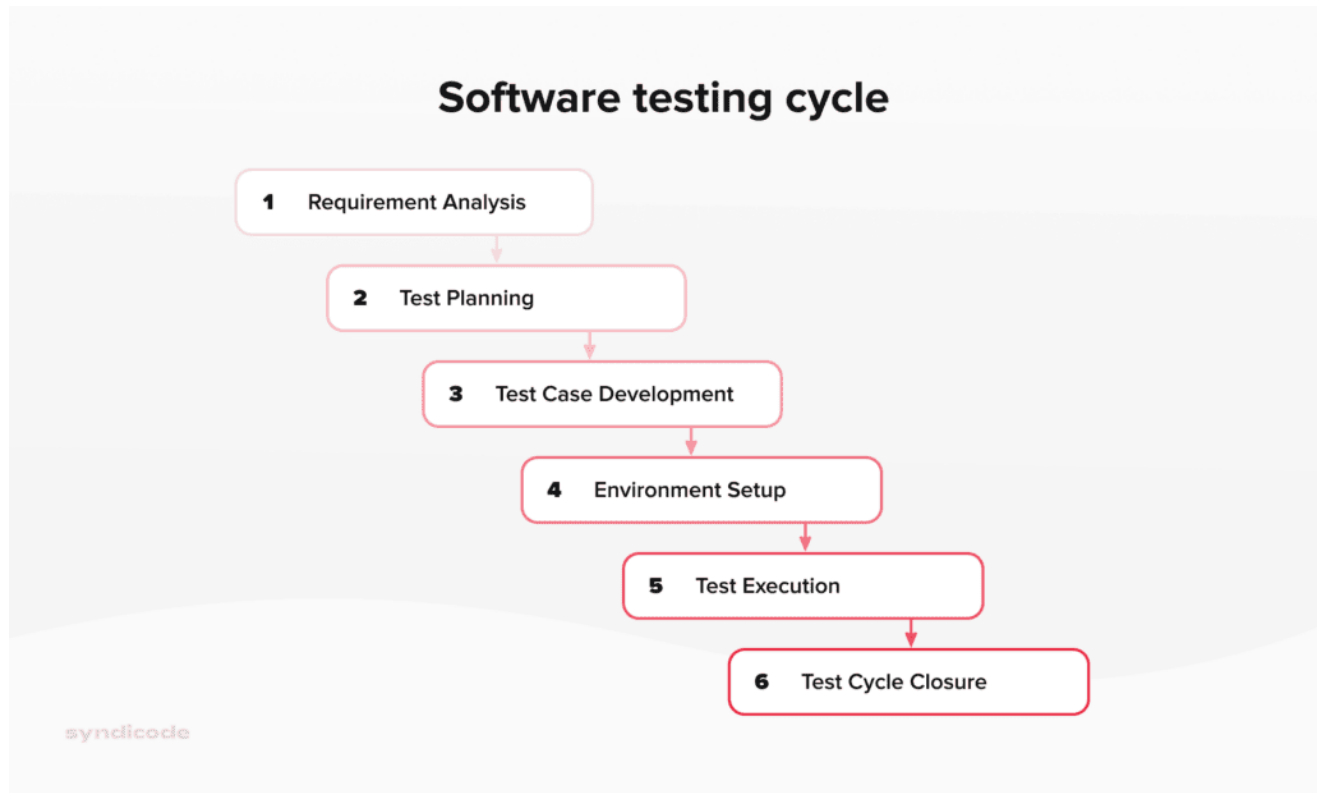
1. Requirement analysis
2. Test case planning
3. Test case development
4. Test environment setup
5. Test execution
6. Test closure

If you are eager to learn what STLC is, where it belongs in the software development life cycle, and how to do it right, make sure to read this post until the end.

What is the software testing life cycle (STLC)?

A software testing life cycle is a series of steps to ensure that the product meets the quality objectives. It is an integral part of the software development life cycle that starts as soon as requirements are defined. A strategy based on STLC enforces a systematic and documented approach to testing and increases the chance of producing a better software product.

There are six main phases of the testing life cycle. Some can be split into smaller steps or skipped depending on the project's nature, the available resources, and the chosen SDLC model.



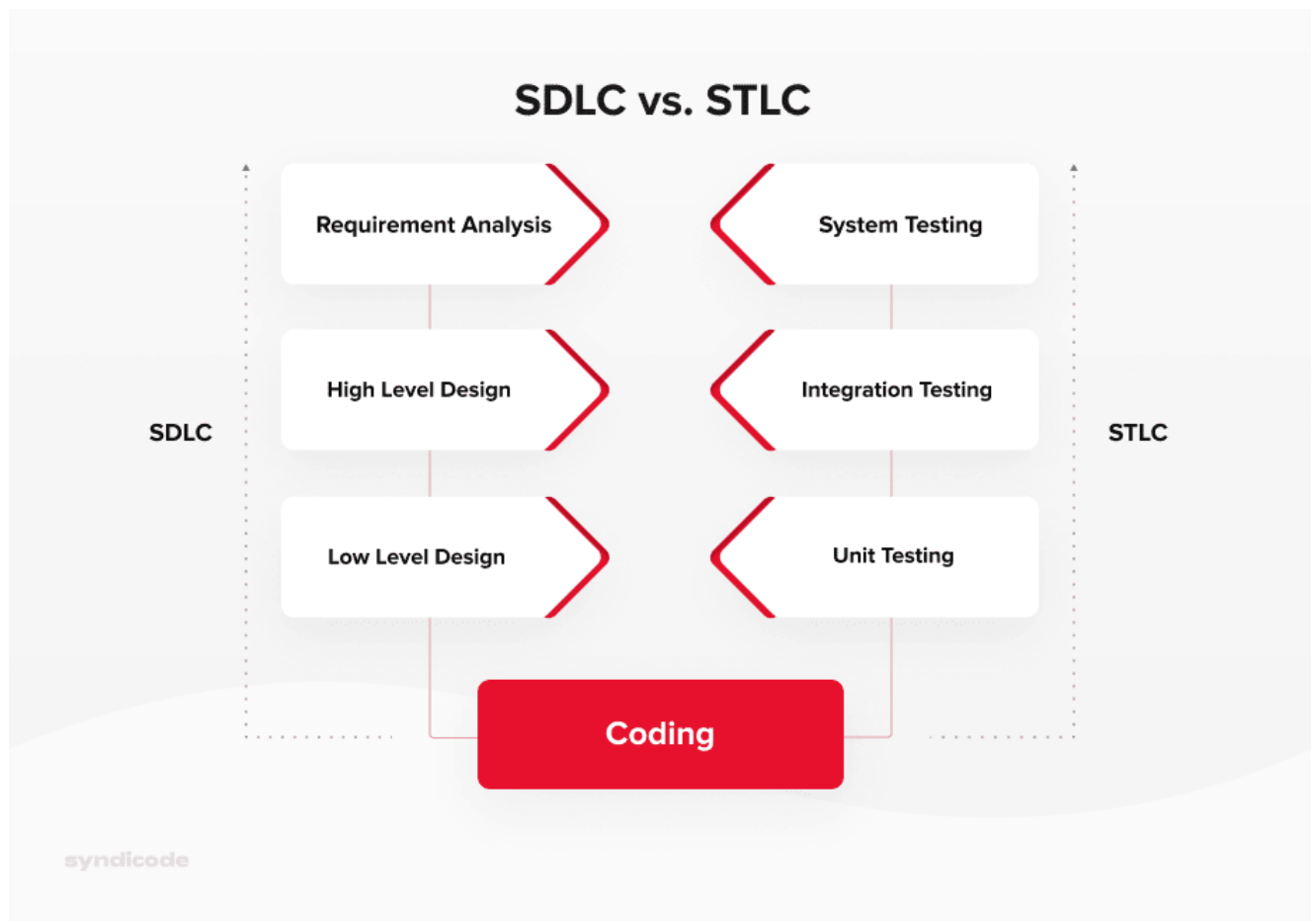
Software testing cycle

Why do you need STLC in the software development lifecycle?

Both the software development life cycle and software testing life cycle relate to software application development and often overlap their timelines. However, they pursue different objectives. SDLC aims to deploy high-quality software, whereas STLC is there to identify any parts of the software that are not working as expected.

Teams that adhere to the STLC methodology begin testing operations early and validate the quality of digital products as they are developed. This way, they minimize the chance of errors at the product deployment.

The software testing life cycle can be performed as a series of steps within the SDLC or alongside its stages. Yet, while the development cycle can happen without the testing cycle (which is highly discouraged), the opposite is not possible.



SDLC vs. STLC

Benefits of software testing life cycle

Teams that follow STLC achieve better consistency and effectiveness when testing a software application. They also have goals for each project aspect defined more clearly, and all the project requirements verified. Since all testing activities are planned beforehand, the time constraints for testing are more precise. Finally, the software testing life cycle ensures that each feature is tested before adding any additional ones.

For a business owner, the advantages of the software testing life cycle mentioned above result in the following:

- STLC can enhance your product vision;
- STLS maximizes your chances of getting the end product that fully meets your expectations;
- Systematic testing leads to faster bugs identification and fixing;
- Well-defined goals and phases help to track project progress.

Key STLC phases explained

Going through a sequence of STLC steps guarantees a high-quality software release. It happens due to methodical validation through ideation to design and execution, ensuring the goals are set and clear and the necessary deliverables are present.

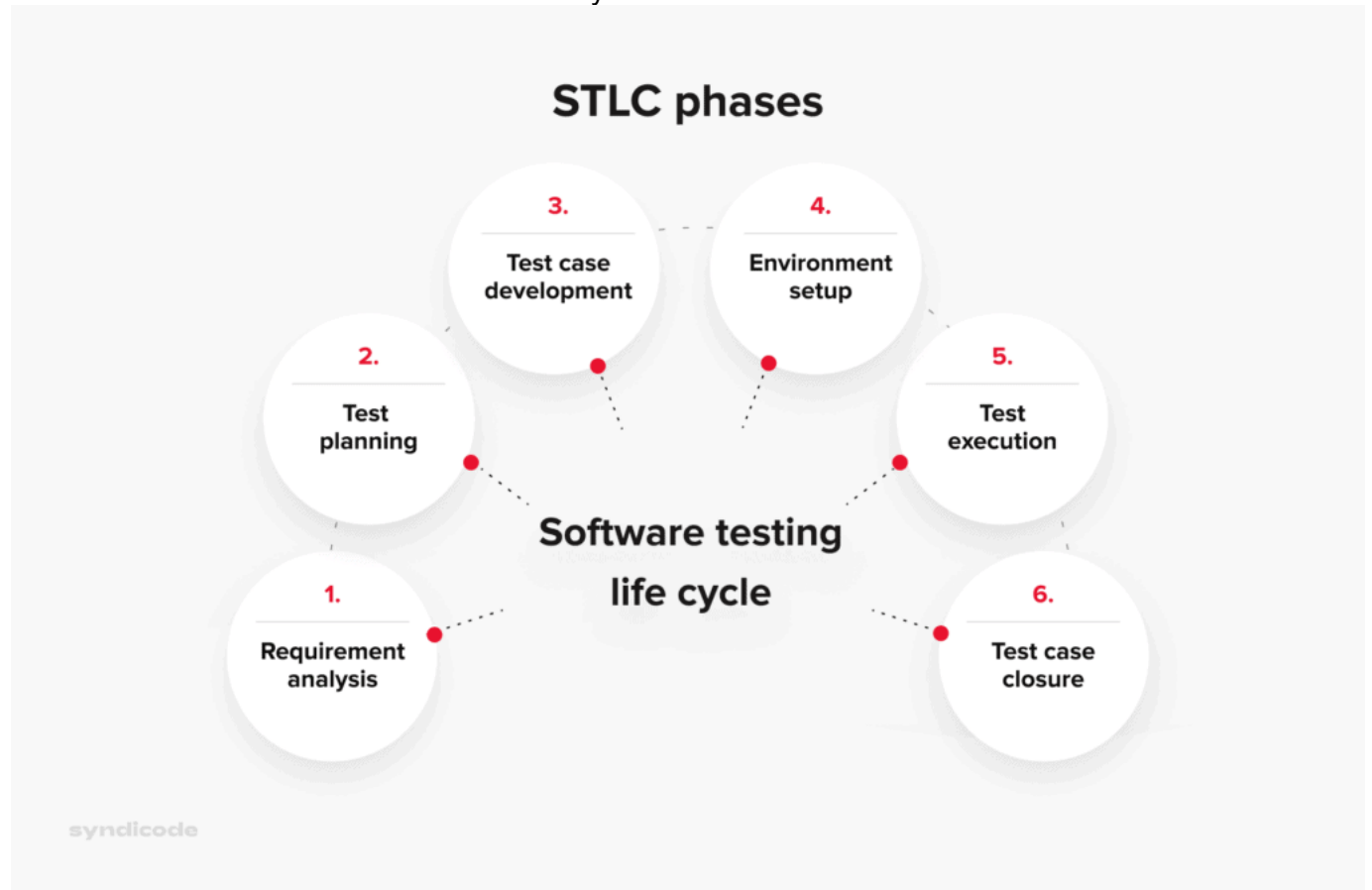
There are six main phases of the software testing life cycle. However, following them all is not mandatory, as we said earlier. Likewise, you can split some phases into smaller steps if that aligns with your development cycle better.

Phases of STLC:

1. Requirement analysis
2. Test planning
3. Test case development
4. Test environment setup
5. Test execution
6. Test

cycle

closure



STLC phases

Let's go over each phase in more detail.

1. Requirement Analysis

[QA and software testing services](#) usually begin with reviewing and analyzing available requirements and specifications. Project requirements often include software requirements and system requirements.

Software requirements cover high-level business needs, the system architecture, and details on each feature's design and maintenance. Functional and non-functional requirements describe system requirements.

The testing team starts identifying the requirements that produce outcomes by feeding them with specific input data. These are testable requirements. Ideally, all the requirements should be testable. Otherwise, it's impossible to determine acceptance criteria and validate a feature's quality.

Next, testers work within their teams and cross-functionally to decide how to test software and prioritize their activities. They also determine if automated testing is feasible, clarify vague requirements, and consider an appropriate test environment.

In a nutshell, the Requirement Analysis stage aims for the following:

- To understand the expected output from the product
- To identify blind spots or ambiguities in the specifications
- To prioritize testing activities
- To determine the automated testing feasibility

The Requirement Analysis stage results in these deliverables:

- Requirements report
- Requirements traceability matrix (RTM)

2. Test Planning

After analyzing the requirements, the QA engineers outline the scope and objectives for testing. They outline the actions and resources required to achieve the goals, as well as the testing measures and methods for gathering and tracking those metrics.

Now, the requirements received and validated at the previous step turn into a testing strategy. This document will keep the testing and development teams on the same page. This is especially helpful when testing occurs in parallel with development.

Based on the created plan, the team analyzes risks and estimates time and effort. Then, management decides on testing tools, assigns roles and responsibilities, and puts together the deliverables.

Briefly, the goal of the Test Planning stage is:

- To create a plan for the testing activities
- To estimate time and efforts
- To pick the most appropriate tools
- To assign tasks to teams and individuals
- To identify the need for training

The Test Planning phase deliverables are:

- Test strategy
- Risk analysis
- Effort estimation

3. Test Case Development

Based on the test strategy, the testing team develops test cases. This is the most creative part of the process, as testers have the areas to validate specified in the requirements, but they are free to choose *how* they achieve this task.

In addition, a test has to cover all possible use cases of the same module. Every test case includes test inputs, processes, execution circumstances, and anticipated outcomes to guarantee clarity and transparency.

Next, test cases are prioritized due to their effect on the product. If automation is applicable, the scripts are written, too. Eventually, all the test cases are reviewed, updated, approved, and combined in test suites.

To sum up, the Test Development phase aims to:

- Create test cases
- Prioritize test cases

The deliverables of the Test Development step are listed below.

- Test cases
- Test data

4. Test Environment Setup

This step can be done in parallel with test case development. The QA team refers to the test plan and decides which test environment they should use.

The test environment represents software and hardware conditions, frameworks, configurations, test data, and networks. These should mimic the devices of the product end-users. For instance, most product users are on Android devices, and some of them also use Chrome. Therefore, a test environment should include the parameters of an average Android device and the most popular Chrome version.

The QA team could perform smoke tests if some software modules were already built. These are very basic checks to see if the product is ready for more comprehensive testing.

Briefly, the goal of the Test Environment Setup phase is:

- To understand what software and hardware are required
- Prioritize test environments (if multiple)
- Setup test environments
- Smoke test the software build

Test Environment Setup step deliverables:

- Environment ready
- Smoke tests results

5. Test Execution

Testers follow the test plan and carry out tests prepared in the previous step. The goal is to execute the test cases within the allotted time. If there is a bug, it goes to the bugs report to be addressed by the development team.

As developers fix the defects, the QA team often retests the module to ensure no new bugs. Test automation tools help greatly speed up repetitive testing tasks and save time and resources.

Let's outline the Test Execution phase objectives:

- Execute tests as specified in the test plan
- Log defects and map them to test cases in RTM
- Retest the defective modules
- Track the defects in closure

Deliverables of the Test Execution step:

- Results of test cases
- Defect reports
- Updated requirements traceability matrix

6. Test Closure

Test closure is the process of test execution completion and the last phase of the STLC. The QA team verifies the testing objectives were met and reports any unexpected or unusual software behavior revealed during testing. They also archive all the resources used for testing, such as tools, scripts, and environments, for the client.

Further, the testing team meets to discuss and analyze the testing work and results at this step. They calculate test coverage, take lessons from the current cycle and identify ways for improvement to implement them in the future.

The main goals of the Test Closure phase:

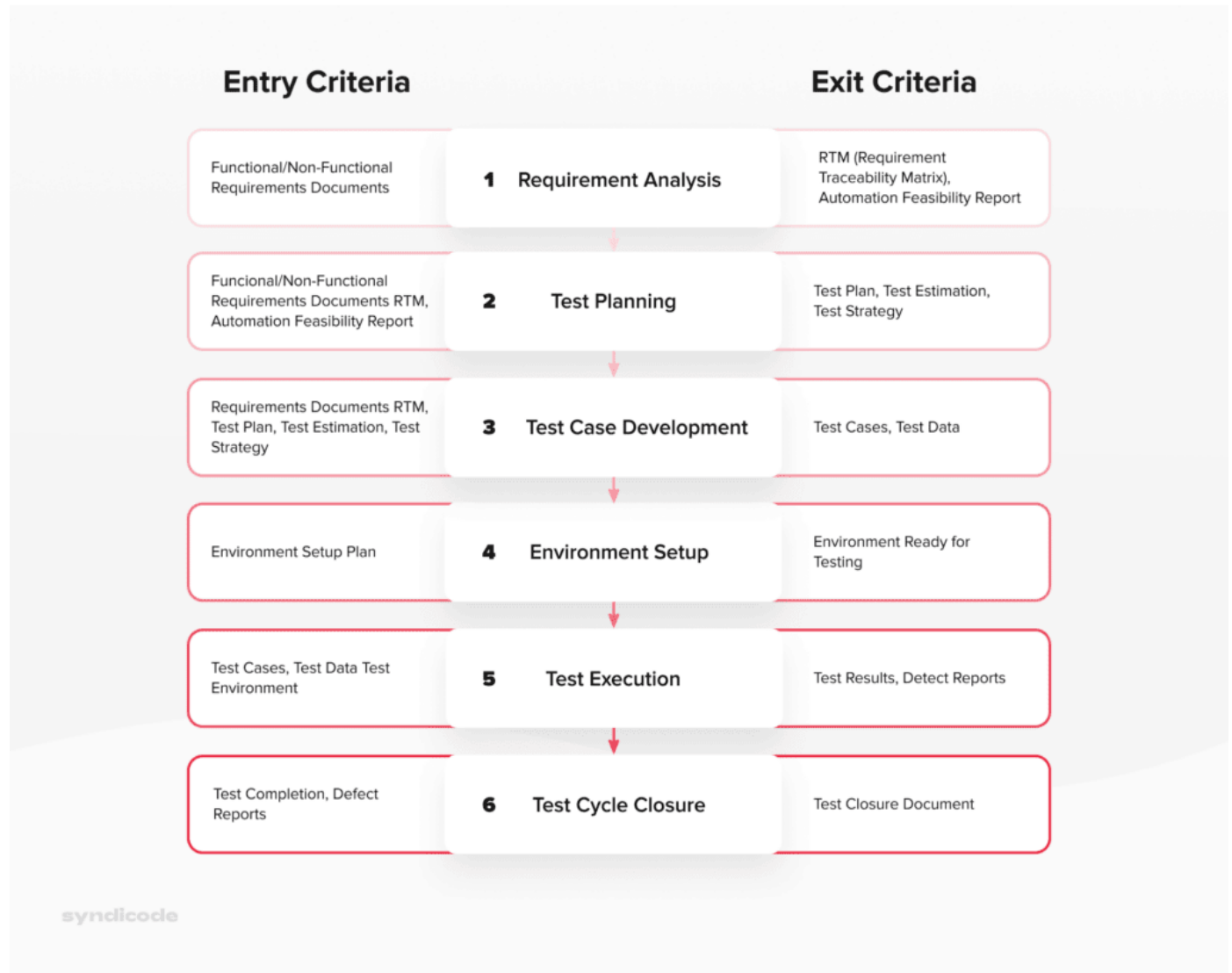
- To evaluate testing completion criteria
- To prepare the test closure report
- To prepare the qualitative and quantitative product quality reports
- To analyze test results

Test Cycle Closure deliverables:

- Test metrics
- Test closure report

What Are the Entry and Exit Criteria for Testing?

Entry and exit criteria in STLC are the conditions for a phase to start and end. They help limit the testing time and achieve the required quality, functionality, and efficiency. Different phases of testing have specific criteria.



Entry and exit criteria

Entry Criteria

Entry criteria state what a tester should do before starting the testing. Below, we listed the activities and documents that should be completed before beginning an STLC phase:

- Requirement analysis phase: Requirements document, Acceptance criteria, Application architecture
- Test planning phase: Requirements document, requirement traceability matrix, test automation feasibility document
- Test case development phase: Requirements document, RTM, test plan, automation analysis report
- Test environment setup phase: system design and architecture, environment setup plan

- Test execution phase: RTM, test plan, test scripts and cases, ready test environment, set up test data, unit/integration test report for every build to be tested
- Test closure phase: completed testing, test results document, defect logs.

Exit Criteria

Exit criteria specify the activities and documents that a tester should complete before starting the next phase. Ideally, the next stage cannot begin until all the exit criteria of the previous step have been met.

In practice, the entry criteria for the next phase do not always require documentation from the previous step. So testers can complete them later to speed up the process.

The exit criteria for the software testing life cycle phases are:

- Requirement analysis phase: RTM and test automation feasibility report signed off by the client
- Test planning phase: test plan/strategy and effort estimation document approved
- Test case development phase: test cases and test data reviewed and signed off
- Test environment setup phase: environment setup works as planned, test data setup completed, successful smoke test
- Test execution phase: all the planned tests were executed, defects logged and tracked to closure
- Test closure phase: test closure report signed off by the client.