IJARIIT INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078

(Volume 8, Issue 1 - V8I1-1215)
Available online at: https://www.ijariit.com

Software testing life cycle and its efficiency

Lok Shan Chan
Independent Researcher

ABSTRACT

The software testing life cycle (STLC) is the sequence of activities that takes place within the software testing process. By following a logical STLC, an organization is able to develop a quality plan that is more likely to yield superior results. Every organization's ultimate goal is to provide its customers with a flawless product. Customers are easily turned off by bug-filled user experiences. As a result, when businesses became aware of this, they began incorporating testing. Since then, testing has evolved into a critical component of every organization's operation. Over the last few decades, there has been a shift in the competency of testing. It is important for businesses to recognize that software testing should have its own life cycle which can increase efficiency of testing.

Keywords: *Software Testing Life Cycle, Testing, and Product.*

1. INTRODUCTION

A software testing life cycle (STLC) is the sequence of changes an entity goes through from one form to another. Many concrete and obscure entities go through a series of changes from start to finish. Within the STLC, the software is an entity. The STLC is the process of executing different activities during testing.

These activities include checking the developed software to see if it meets specific requirements. If there are any defects in the product, test engineers work with the software engineering team. In some cases, they have to contact the stakeholder to gain insight into different product specs. Validation and verification of a product are also important processes of the STLC which can increase the efficiency of testing.

2. THE ROLE OF STLC

Even if a company employs the most talented software engineers, mistakes can inevitably occur. The primary function of STLC is to identify and correct any errors that have occurred. The primary purpose of performing an STLC is to ensure that product quality is maintained. Today's businesses must carry out extensive testing in order to succeed. Every phase of product testing, from planning and research to execution and maintenance, is critical to the success of the project.

In the long run, a high-quality product can result in reduced maintenance expenses than a low-quality product. When attracting new users, the reliability of an application or software is essential. Aside from that, consistently reliable products also aid in the retention of existing customers. It is critical to concentrate on each phase of the STLC in order for a product to remain viable in the business.

3. PHASES OF STLC

The validation of each module of software or application is essential for ensuring the precision and correctness of the product. Software testing can be a time-consuming procedure, and test engineers break it down into phases to ensure uniformity. If testing is not organized, it is possible for complications to arise. Unresolved issues, unreported regression bugs, or, in the worst case scenario, a module that bypassed testing because the deadline was approaching, are examples of complications.

Each phase of the STLC has a specific aim and deliverables that must be accomplished. Initiation, execution, and termination of the testing process are all covered.

3.1 Requirement Analysis Phase

The valuable software test engineers must take the time to review, understand, and analyze the specifications and requirements that have been made available. Certain requirements produce outcomes by providing them with input data. These requirements can be put to the test in controlled environment. Test engineers look at both functional and non-functional criteria while developing test plans. Following that, they must select requirements that can be tested.

In this phase, activities such as requirement analysis brainstorming, identification of test needs, and prioritization of test requirements are carried out. They also cover the process of identifying requirements for both automated and manual testing procedures. Understanding more particular data about a product's features and functions is essential during the requirement analysis process. The test engineers must learn how the product should be in its optimal state before the requirement are implemented.

An in-depth requirements report, together with study of test automation feasibility, are produced as deliverables during this phase. One of the most critical deliverables developed during this phase is a requirements traceability matrix.

Traceability refers to the ability to trace back artifacts to their corresponding requirements. For example, having traceability in the software development process means that the company should be able to trace each change in its codebase back to the requirements that were originally specified.

In the requirements traceability matrix, an organization can relate various artifacts back to the requirements that they were created for. The ability to trace back software testing activities to their original requirements is important for test engineers while performing software testing. Test engineers eliminate waste by ensuring that every testing effort is linked to a requirement that provides value for the client in this manner.

3.2 Test Planning Phase

The second phase is test planning, which is done by the test engineering team once they have analyzed all of the relevant testing requirements. After gaining an understanding of the product domain, they establish the scope and objectives. The team then conducts an analysis of the risks involved and establishes timetables and testing environments in order to develop a strategy. After that, management finalizes the tools and allocates roles and tasks to the various members of staff. In addition, an estimated time frame for completing the testing of each module should be specified.

The test plan, which is a document that describes the purpose for and details of the testing activities for a specific project, is the most important delivery generated during this step.

3.3 Test Case Design and Development Phase

Design and development of test cases are carried out in accordance with the test strategy by test engineers. Extensive and comprehensive test cases should be used to ensure that practically all possible scenarios are covered. All of the possible permutations and combinations should be put together in one place. Test engineers can prioritize these test cases by conducting research to determine which of them are the most common or which of them can have the most impact on the final product. Following that, at the documentation stage, there are the verification and validation of the established requirements. In addition, the reviewing, updating, and approval of automation scripts and test cases are critical tasks for this stage of the development process. This step also includes the definition of various test conditions, as well as the input data and anticipated outputs.

As a result, the actual test cases arranged in their respective test suites are the primary deliverables produced during this phase.

3.4 Test Environment Setup Phase

Testing activities necessitate the use of specific environmental components, such as servers, frameworks, hardware, and software, in order to execute the test cases that have been generated. The configuration of software and hardware, as well as the preparation of test data, are the primary components of this phase. Smoke testing and providing test engineers with issue reporting tools are also required. It is critical that the test environment encompasses all of the environments that the user might encounter in the real world. A feature that works in one internet browser, for example, may not operate in another internet browser. Furthermore, the functionality of features varies depending on the software and hardware requirements of the user. For example, when using 8GB of RAM, a function may run easily, but when using 2GB of RAM, it may cause problems. The test engineers can benefit from conducting research on the contexts that end users are familiar with.

At the end of this stage, the primary deliverable can be a comprehensive strategy for test environment management.

To ensure that the test environment is properly set up, it is the responsibility of the test engineering manager who is in charge of managing the team.

3.5 Test Execution Phase

Once the team has completed all of the previous processes, the application can be ready for testing to begin. Test engineers carry out the test cases in accordance with the test plan. They also locate, detect, and log defects or bugs, thereby notifying the appropriate authorities. In addition, the team is in charge of comparing expected results with the actual outcome of the test. If any bugs are identified, they must be documented so that they can be forwarded to the software engineering team for resolution.

Once a bug has been fixed by the software engineering team, regression testing can commence. The purpose of regression testing is to guarantee that the software or application continues to function properly after a change has been implemented. When testing after a bug fix, it should be made sure to test the entire product once more. A bug patch could result in the introduction of a new bug in another component of the product. The same tests must also be run repeatedly after each fix and deployment to ensure that they are reliable. The usage of scripts or automated testing tools is therefore strongly encouraged.

During this phase, the major deliverables are the test results, which, in an ideal world, should be validated and disseminated in a totally automated manner.

3.6 Test Closure Phase

The completion of test execution and delivery of the finished product indicate the beginning of the test closure phase of the project. The test engineering team examines the results of the tests and discusses them with the other members of the team. Product quality, test coverage, and project cost are some of the key factors that they take into consideration. If there is a departure from the estimated values, additional analyses can be performed to determine why items did not turn out as planned.

It is a vital procedure for test engineers to come together and discuss the results of their testing after they have completed their work. Any problems encountered during testing, as well as weaknesses in strategies, can be mentioned in this discussion. Test engineers can also work on developing a better testing approach based on the learning during the testing process, if necessary. If test engineers adhere to the release practice, testing can be done on a regular basis. When sending reports to different stakeholders, test engineers can consider how often to send reports and what details to include in the reports.

Additionally, the team takes into account the test data, the achievement of goals, and the adherence to deadlines, among other factors. They can analyze the entire testing strategy and procedure after they have a complete understanding of what happened.

The lessons learned document and the test closure report are the primary deliverables during this phase.

4. THE ENTRY AND EXIT CRITERIA FOR TESTING

Each of the six phases of a STLC has a set of criteria for entering or exiting the phase. Test engineers must complete the execution of the test cases within a specified time frame. Additionally, they must ensure that the ultimate product's quality, usefulness, and efficiency are maintained. As a result, it is essential to define entry and exit criteria.

4.1 Entry Criteria

The entry criteria specify which conditions the team must meet before beginning the testing procedure. It is mandatory to cross out all requirements before testing can begin.

It is necessary for certain ongoing activities and conditions to be present before any testing can take place. Firstly, test engineers require feedback from members of the software engineering team. The test plan, test cases and data, the testing environment, as well as their codes are all items that test engineers would like to look at.

4.2 Exit Criteria

Exit criteria are a list of requirements and actions that must be completed before the testing is completed. This means that, they include tasks to be done and processes to be completed, before the testing process is brought to an end.

The identification of high-priority problems can be included in the exit criteria. Those issues must be addressed as soon as possible by test engineers. They must pass a variety of test cases and guarantee that all functional requirements are met.

5. CONCLUSION

It is critical to make efficient use of time and resources in order to make the testing process more efficient. A systematic STLC not only results in faster issue fixing, but it also improves the overall quality of the product. It is also able to improve customer satisfaction and increase the efficiency of testing.

6. REFEREENCES

[1] Everett, G. D., & McLeod, R. (2007). *Software testing: Testing across the entire software development life cycle.* Wiley-IEEE Computer Society Press.

[2] Patton, R. (2001). *Software testing.* Sams.

[3] Perry, W. E. (2000). *Effective methods for software training.* John Wiley.

[4] Lewis, W. E. (2000). *Software testing and continuous quality improvement.* Auerbach.

[5] Wieczorek, M., Meyerhoff, D., & Baltus, R. (2001). *Software quality: State of the art in management, testing, and Tools.* Springer.