# REVIEW PAPER ON SOFTWARE TESTING AND LIFE CYCLE

**Manasi Rasane[1], Dr. B. V. Pathak[2]**

*Electronics & Telecommunication, MKSSS's Cummins College of Engineering for Women, Pune, Maharashtra, India*
*Electronics & Telecommunication, MKSSS's Cummins College of Engineering for Women, Pune, Maharashtra, India*

[1]manasi.rasne@cumminscollege.in , [2]bageshree.pathak@cumminscollege.in

## *Abstract*

*Testing is a method of determining the accuracy, completeness, and quality of software that has been produced. It refers to a series of actions carried out to identify software flaws so that they can be fixed before the product is released to the client. It provides metrics for determining the software's efficiency and correctness. It mostly focuses on ways to increase the software's functionality as well as its performance. We must validate and verify software before handing it on to the customer to completely test it. As a result, software methodologies and software testing procedures have been created for this aim. Program testing techniques are concerned with how specific component of the product will be tested, whereas software testing is concerned with how the entire software will be tested. This article describes software testing and its various components, such as software testing approaches. It also explains the basic life cycles that are necessary for quality testing.*

***Keywords:*** *Software Development Life Cycle, Software Testing Life Cycle, Software Bug Life Cycle, Software Testing Technique*

## 1. Introduction

Software development is the process of creating software in response to a set of requirements. Testing is required to verify and confirm that software developed meets these requirements. Software testing helps in the prevention of system problems. It refers to the process of analysing software to determine where the flaws exist. It's also used to check for things like usability, compatibility, stability, integrity, efficiency, security, capability, portability, and maintainability in software. Software testing hopes to ensure specified objectives and principles that must be committed to.

Testing, in basic terms, is the process of identifying defects in software. Software testing is the process of running software for (i) Verification, (ii) Error Detection, and (iii) Validation.

i. Verification: Verification is the process of testing objects, such as software, for conformance and consistency by comparing the outcomes to pre-determined criteria. [Are we manufacturing the product, In the right way?]
ii. Error Detection: Error Detection is the process of intentionally completing incorrect inputs to test the system's functionality.
iii. Validation: Validation examines the system's correctness in the process of ensuring that what has been stated is exactly what the user desired.

## 2. Literature Survey

In [1], compares the current state of software testing with early testing. This article provides a thorough overview of modern software testing terminology and trends.

In [2], the paper describes several software testing approaches, the need of testing, and the goals and objectives of testing.

In [3], various forms of testing are considered in this work. Different software testing methods are discussed.

In [4], various software are used in this article. Test methods are considered. Two types of testing white-box testing and black-box testing. The focus is on how to help companies identify and understand the risks of using software testing. A free look at the software provides a purpose.

## 3. Methodology

An important part of the software development life cycle (SDLC) is testing. It helps develop the developer's confidence that the program is doing what it is supposed to do. In other words, it is a way to execute a program to detect errors.

### 3.1 Software Development Life Cycle

The life cycle of testing refers to the process of executing test cases in a planned and systematic way. Simply put, life cycle can be defined as the transition from one form to another. Every living thing has a life cycle. The following are the basic life cycles that must be followed when producing software.

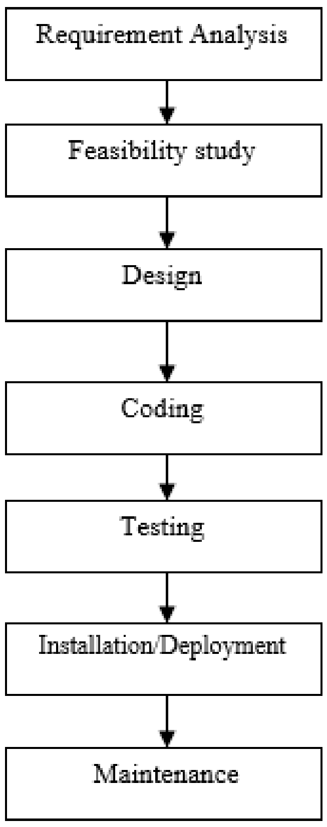

Figure 1. Software Development Life Cycle.

### 3.1.1 Requirement Analysis

The first part of the software development life cycle is requirements analysis. At this stage, all necessary or relevant information is obtained from the client. Customer interaction is done to develop products that meet customer needs. Business analysts and project managers, meeting with them and what are the goals of the product, who are the product's end users, etc. They interact with customers by gathering the necessary information to have a solid understanding of the product before installation. This will help you understand the product range. After gathering all the information, you can perform a product analysis to determine how long it will take to produce the product, how many resources are needed, limitations, product benefits, etc. Based on this research, we will understand the capabilities of the product. The necessary documentation should be completed at this stage so future developers and customers can use it.

### 3.1.2 Feasibility study

Documents are created for feasibility studies required for software development. Several checks were made to create this specification. These documents include information such as whether the project can be completed within a given budget or whether existing computer system software can support the various tasks and processes expected by customers and clients or whether the project can be completed on time. Complete documentation covering the elements of a product or program.

### 3.1.3 Design

In this stage of development, all product features, architecture, and all standards of the proposed system have been designed. There are two approaches to software development.

The high-level design provides a brief description of the architecture, specifies the features required for the module design, defines the interface specifications, and establishes the architecture information and its diagrams, database identifiers, and technical details. And the architecture diagram is created. All of these points are documented in low-level design constructs for input and output, error messages, product interfaces, database type and size, and dependency issues. This information helps developers understand the overall design of the system.

### 3.1.4 Coding

All of these points are addressed in low-level design structures for input and output items, error messages, product interfaces, database types and sizes, and dependency issues. This information helps developers understand the overall design of the system. This is the most time-consuming part of the cycle. Developers write code at this time. Developers write code in their favourite programming language. At this stage, tasks are divided into subtasks or modules. Different jobs are assigned to different developers. Developers must follow these standards when writing code. When writing code, you have to deal with some functional requirements. Developers must adhere to design documents.

### 3.1.5 Testing

At this stage, product functionality is tested by QA and testing teams. QA and test teams test the product to ensure that the current results match the expected results outlined in the requirements specification. If the QA and testing team find a bug, they report it. The bug is fixed by the developer and reassigned to tester. QA and testing teams will retest the bug. QA and the testing team will close this bug once it is fixed. If the bug is not fixed, it will be reopened by the QA and testing teams. This process will continue until the product is free of bugs. In addition, the product must meet all the specifications listed in the requirements specification.

### 3.1.6 Installation/Deployment

At this stage, when all the bugs are marked closed by the QA and testing team, the product reaches a level where the system is error-free and all the actual results are in line with the expected results. It also depends on the project manager. The feedback will determine whether the software has been installed or not. The software will be hosted by the administrator.

### 3.1.7 Maintenance

Once the product boots successfully, it enters the maintenance phase. After installing the product, customers start using it. Customers provide feedback on items. This helps the team decide which features to include. Product testing continues after deployment. Some possibilities may not have been evaluated at all. Bug fixes are done at all stages. After the bug is fixed, the product will be upgraded. During this phase, the production team may add new features.

## 3.2 Software testing objectives

- A good test may find an error that has not yet been identified.
- The test is not worth wasting time on.
- A successful test is a test that identifies an error that has not yet been detected.
- A good test should not be too easy or too hard.
- To determine if the system is running as planned.
- To determine if the system is "fit for purpose".

• To determine if the system meets the requirements and can be successfully implemented in the intended site environment.

### 3.3 Software Testing Life Cycle

Software testing is important for improving quality and better customer service, as developers can make many mistakes during software development. Some errors are minor, while others are significant because they can degrade system performance or pose a risk to the system. Therefore, to fix these problems, you need a testing method to identify the major problems that the software may cause. Therefore, software testing is necessary to correct all mistakes and errors and improve the quality of the software.
Software Testing Life Cycle (STLC) is the implementation of seven main processes for testing purposes.

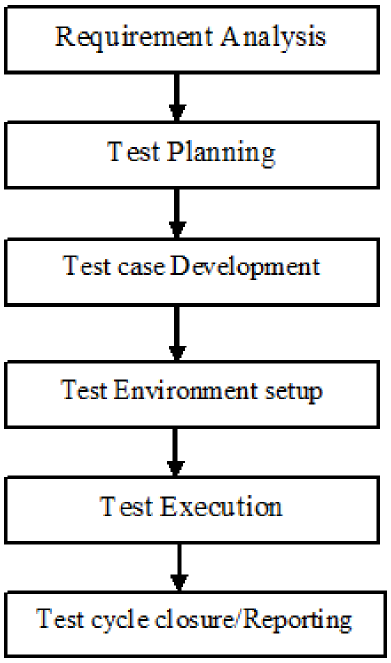

Figure 2: Software Testing Life Cycle

#### 3.3.1 Requirements Analysis

At this stage of the software testing life cycle, test teams obtain information based on the test perspective. That is, determine which product requirements require testing. Teams looking to improve product quality can communicate with customers, technical leaders, etc. to fully understand their needs. There may be functional or non-functional requirements.

#### 3.3.2 Test Planning

During this phase, the test team leader or test team manager is responsible for planning how to conduct the test. This test analysis is planned and includes some test tasks and test cases.
The test control phase is responsible for evaluating and measuring test results and controlling multiple test activities. In addition, all tasks in this phase are completed by a senior quality assurance manager who determines the costs and efforts already determined by management

at the start of the project. The senior quality assurance manager then makes the final decision on the test plan.

### 3.3.3 Test Case Development

During this phase, validation is performed to determine if all test processes that have been tested before this phase are working properly. Go through all test cases multiple times to make sure all details are correct and all features work. In this step, test cases are created to validate all the basic tests. If a test environment is available, test data is generated for verification.

### 3.3.4 Test Environment setup

This phase will determine how the product/software hardware and software are tested. It is an important part of the testing process and should be done at the same time as the test. A professional testing team is not required for the testing process if the development team provides requirements for customers to regularly test the product to determine the performance status of the product or system.

### 3.3.5 Test Execution

During this phase, several test cases are executed. Essentially, there will be a comparison between what the customer expected and what the result was. During the test execution phase, many product tests are built to test the product, whether the product tests will work properly to build a test environment

### 3.3.6 Test cycle closure/Reporting

This phase will be activated after the testing process is completed. After all, tests are completed, this phase will be completed. The closure of the test cycle has the property (depending on the type of device) that it has a specified time to perform its tasks at that time.
Test reports are very important to determine whether your product is working properly or not and what are the various defects. This information is provided in the form of a report. These reports can be presented by team leaders or senior members on a daily, weekly, monthly, or yearly basis.
Test reports are very important to determine whether your product is working properly or not and what are the various defects. This information is provided in the form of a report. These reports can be presented by team leaders or senior members on a daily, weekly, monthly, or yearly basis. This stage is mainly related to the closure of the testing process; the test cycle must be closed or completed after all requirements or expectations defined at the beginning of the testing process have been met. It is mainly used for signals or messages that are sent after the job is done.

## 3.4 Bug life Cycle

The Bug life cycle has some stapes or stages. The bug life cycle begins when the tester fined a new bug and the test bug closes it. The life cycle of a bug varies from project to project.

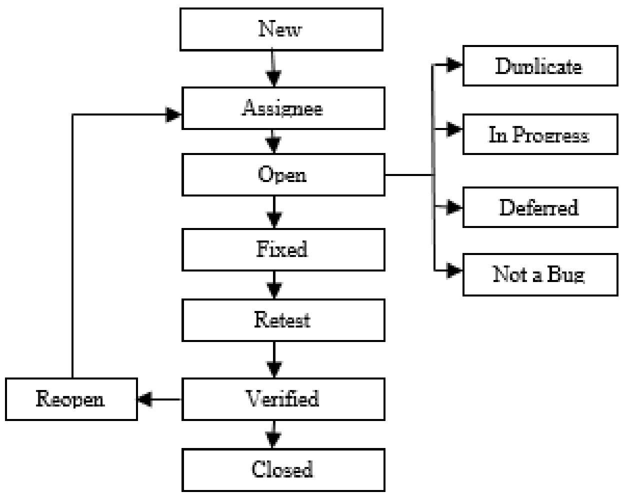

Figure 3: Bug Life Cycle

Each stage of the steam life cycle is described below:

New: When software testers find and report bugs, set the status to New.

Open: When a tester creates a new issue, the status of the issue is open.
Assignee: If a new bug is found by the tester. Tester reported this bug to developer.

Duplicate: If multiple testers report the same bug, one of the two bugs may be marked as a duplicate.

In progress: When the developer starts his work. At this point, the status is In progress.

Fixed: Once the creator has resolved the issue, it's over. Mark error status as fixed.

Retest: When a developer returns a bug to a test. Tester fixed the error.

Verified: When the tester stops the error. It ensures that all the actions shown here are performed correctly.

Closed: If the test case mentioned above works, the tester will mark the error as closed.

Reopen: If the bug does not work correctly or does not get the expected result, it marks the bug as reopened and reassigns the bug to the appropriate person.

Deferred: Bugs reported are not a priority and may be fixed in future versions. The error status is pending.

Not a bug: If the error reported is invalid or invalid. This error is then marked as a non-error.

### 3.5 Software testing techniques

After successfully developing the code, testers must test the system's functionality. Software testing approaches are divided into two categories: Static testing and dynamic testing

#### 3.5.1.1 Manual testing (Static testing)

Manual testing is the most basic sort of testing. Manual testing involves the execution of test cases by testers. Before performing automation testing on any product, it should be manually tested. Manual testing aids in the discovery of product flaws. When a project is in the early stages of development, manual testing can be chosen. When evaluating user interfaces, especially aesthetic features, and when working on a short-term project. If you have a short-term project and scripting takes a lot of time, manual testing is recommended.

**Steps of manual testing**

Step 1: For manual testing, we need to understand the requirement. So, by knowing the requirements, we can determine what has to be tested and what qualifies as a bug or defect.

Step 2: Create test cases. We can write test cases once. We understand the requirements. These test cases essentially take us through the sequence of tests, test functions, and various situations written within a software product or application.

Step 3: Once we finished writing test cases. The next step is to run the tests. It's time to start testing after the test cases have been written and the test environment has been appropriately prepared.

You can go through test scenarios and mark each one as pass, fail, or skipped. It is critical to take notes while performing manual testing. That is critical; you cannot overlook the fact simply because your test fails.

Step 4: The next step is to create a proper bug report. As a tester you are also responsible for to keep records of bugs. As soon as you find a bug, you should provide daily information to the development team.

Writing effective bug reports will benefit you and your team in the long run. Answering these troubleshooting questions will save you time.
A good bug report should include the name , strategies to overcome the bug, expected and actual results, scenarios, screenshots, etc. it should be. There must be something extra to help him understand.

Step 5: Create a detailed test report. After the test, you can get a rough idea of how many passed the test, how many failed, how many passed, how many passed, etc. The process is very easy.

#### 3.5.1.2 Automation testing (Dynamic testing)

Automated testing uses automated tools. Automation tools are used to run the test suite. Manual testing requires testers to assemble and execute test suites one by one. A tester should write a test script. Automated testing is more efficient than manual testing. Random tests are not allowed in automated tests. Tools are used for automation testing. Highly skilled testers with deep domain understanding are required in this profession. Validation testing is another term for automation testing. Testers must modify scripts to make changes. Testing can be done on different operating systems, which saves time. Automation testing requires testers to know a programming language. Other forms of testing are also suitable for automation testing. For example, regression testing, load testing and performance testing can be mentioned. This type of test is more accurate since it is done using an instrument. Automated testing is cheap. Manual tests are more reliable than automated tests. Instrumentation and automation engineers need more money. Automated testing requires a simpler test setup or simpler test environment for script testing.

Steps of Automation testing

Step 1: As testers he/she need to find automation tool.
Step 2: Choose the right test tool.
Step 3: Analyze multiple applications and determine the best tools.
Step 4: Train all team members.
Step 5: Create a framework for automation.
Step 6: Implementation of the developed plan.
Step 7: Create the script.
Step 8: Report errors.
Step 9: Maintain the script.

## 4. Results and Discussion

Software testing helps us identify bugs or defects. When defects are spotted early on, it is much easier to design good software. The goal of software testing is to make the software more stable. By completing software testing, a high-quality product can be produced. It is always concerned with the predicted outcome. Testers must prioritize customer requirements. Quality testing requires testers to understand testing techniques. In software testing, the technique of reporting issues is also highly significant. Every software scenario should be tested. This method will result in software improvisation at every level. It is also critical to select appropriate testing procedures. A tester should understand which testing approaches should be employed at various phases of software development.

## 5. Conclusion

This article describes a method for software testing. It also covers all aspects of software development life cycle, software testing life cycle and bug life cycle. This life cycle helps to improve software quality and overall software development process. By using these methods, we are able to provide a stable and reliable product to our end users. Software testers, like all software, require all levels of expertise to ensure that all parts of the system work and fail. A software testing approach helps develop test cases and test scripts for human and automated testing. This method creates methods for evaluating system components and the system as a whole. In manual testing, tests are performed by human testers, while in automated testing, tests are performed by automated technology. Automated testing is more efficient than manual testing. Automated testing requires a simpler test setup than manual testing.

## Acknowledgment

I would like to express my deep sense of gratitude towards Dr. Mrs. Bageshree Pathak my internal guide for her expert guidance and help from time to time. I would like to thanks my mentor Mr. Pankaj Badgujar for his expert guidance during preparation of this Project. I would like to thank all my team members from Xebrium Pvt Ltd for who have helped me directly or indirectly for my project preparation.

## References

[1] Arumugam, Arun. (2020). Software Testing Techniques New Trends. International Journal of Engineering Research and. V8. 10.17577/IJERTV8IS120318.

[2] Akinsola, J E T & Ogunbanwo, Afolakemi & Okesola, Olatunji & Odun-Ayo, Isaac & Florence, Ayegbusi & Adebiyi, Ayodele. (2020). Comparative Analysis of Software Development Life Cycle Models (SDLC). 10.1007/978-3-030-51965-0_27.

[3] Anwar, Nahid & Kar, Susmita. (2019). Review Paper on Various Software Testing Techniques & Strategies. Global Journal of Computer Science and Technology. 43-49. 10.34257/GJCSTCVOL19IS2PG43.

[4] Singh, Jagannath & Kassie, Nigussu. (2018). User's Perspective of Software Quality. 1958-1963. 10.1109/ICECA.2018.8474755.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[5] Jamil,Abid & Arif Muhammasd & Abubakar ,Normi Ahmad,Akhlaq (2016).Software Testing Technique:A Literature Review. 177-182. 01109/ICT4M2016.0

[6] Taley, Divyani. (2020). Comprehensive Study of Software Testing Techniques and Strategies: A Review. International Journal of Engineering Research and. V9. 10.17577/IJERTV9IS080373.