# A Review Paper on Software Testing

Pankaj Saraswat

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Email Id- pankajsaraswat.cse@sanskriti.edu.in

***ABSTRACT: Software testing is essential for reducing software faults, maintenance, and total expenses. How to acquire an appropriate collection of test cases to test a software system is one of the key issues in the software-testing field. A handful of principles that every software engineering student and faculty member should be familiar with are identified. For creating test cases, there are currently a variety of testing methodologies accessible. With the fewest amount of test cases feasible, this collection should assure maximum efficacy. The major objective of this work is to analyse and compare testing techniques in order to determine which one is superior at detecting software errors. When someone think about building software, they constantly focus on making it bug-free and as dependable as possible. At this phase, testing is being used to ensure that the programme is bug-free. There are several test cases that aid in the discovery of defects; thus, in this paper, the author will discuss the most widely used test cases and error detection methodologies. In the future, advanced software testing techniques will change the performance of the software.***

***KEYWORDS: Black Box Testing, Code, Gray Box Testing, Software Testing, White Box Testing.***

## 1. INTRODUCTION

Software testing is described as a discipline for creating a high system that deals with the creation of software products using well methodologies, techniques, subroutines, and processes. Software Engineering, according to the IEEE definition[1], is "the application of a systematic, well-defined, disciplined, and quantifiable approach to the development, and maintenance of software, as well as the study of these approaches, which is considered the application of engineering to software." Software engineering is the process of creating, testing, and documenting computer programmes[2].

Software testing is the purpose of analysing software and identifying of the difference between the existing and the required conditions and to evaluate the features of the software. Testing measures the quality of the developed software. This fact supposes that there are defects in the functionality of the software waiting to be discovered. Testing should get a high priority during any software development effort[3].

### *1.1 Objective of testing:*

Objective of testing are, first checks if the system meets the requirements and can be executed in the intended environment. At every execution of a program, testing has the intent to find a possible error. Successful testing are that find a yet undetected error[4]. The main purpose of software testing is as follows:

- *Validation Testing:*

Validation testing states that the software meets the requirements of the system customer. It is also known as dynamic testing.

- *Verification Testing:*

Verification testing ensures that the software system meets all the functionality. It includes checking for code, documentation etc. It is also known as static testing.

- *Defect Testing:*

The goal of defect testing is to discover defect in the developed software. Tests shows the presence not the absence of defects.

### *1.2 Most common software problems are:*

- Incorrect implementation of the business rules.
- Weak software performance.
- Incorrect results of data searches.
- Incorrect matching of data.
- Inadequate security controls.
- Confusing or misleading data.
- Incorrect file handling, etc.

Testing can be accomplished manually or by automation. In manual testing the tester has the role of the end-user and try to find bugs or wrong behaviour of the software. Manual testing has some testing levels[5]:

- Unit testing
- Integration testing
- System testing
- Acceptance testing

Testing is described as a method of determining whether a given system satisfies its initial criteria. It is mostly a validation and verification procedure that determines if the produced system satisfies the user's needs. Because of this action, there is a discrepancy between the actual and intended results. Finding flaws, mistakes, or missing requirements in a developed system or programme is referred to as software testing. As a result, this inquiry provides stakeholders with precise information on the product's quality[6].

Software testing may be viewed as a high endeavour. The most essential thing for software testers to grasp during the testing process is how to condense a huge number of tests into a manageable test set and make informed judgments about which risks are necessary to test and which are not[7].

Figure 1 depicts the cost of testing and the number of mistakes discovered. Figure 1 clearly illustrates that the cost of evaluating both functional and non-functional items increases substantially. When deciding what to test or how many tests to run, many bugs might be missed. The aim of effective testing is to run the least number of tests possible so that extra testing work is minimised. Software testing is an important part of software quality assurance. The value of testing may be shown in life-critical software (for example, flight control) testing, which can be extremely costly due to the possibility of schedule delays, cost overruns, or outright cancellation[8].

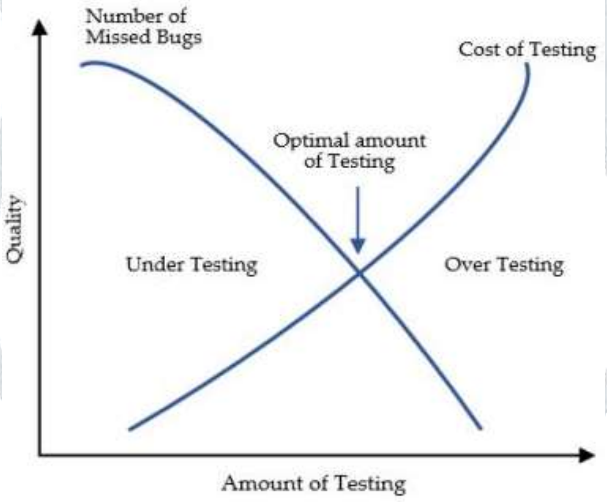

**Figure 1: The Above Figure Shows the Cost of Testing and The Number of Mistakes Discovered.**

Testing includes several levels and steps, and the individual who performs the testing varies from level to level. Unit testing, Integration testing, and System testing are the three core phases in software testing. The software developer or the quality assurance engineer, often known as a software tester, tests each of these processes[9].

### *1.3 Software Testing principles:*

Testing a software means trying to make a software fail. Faults have to be found not to be corrected with the testing. Tests have not to replace specifications. Specifications can produce tests but a not generate specifications. The process of determining the success or failure of the tests must be automatic process. Manually investigating the results in less practical. The testing process must include both manually and automatically produced test cases[10].

### *1.4 Software Testing Techniques:*

This technique is done without executing the code. The requirements, design and testing cases are reviewed and the possible errors are eliminated. Tools analyze the code in order to find defects. This method can evaluate both web or non-web applications and can detect that cannot be seen through dynamic web testing. Figure 2 shows the approaches to software testing.

Example of defects found by tools:

- Unreachable code.
- Programming standard violation
- Duplication
- Complexity

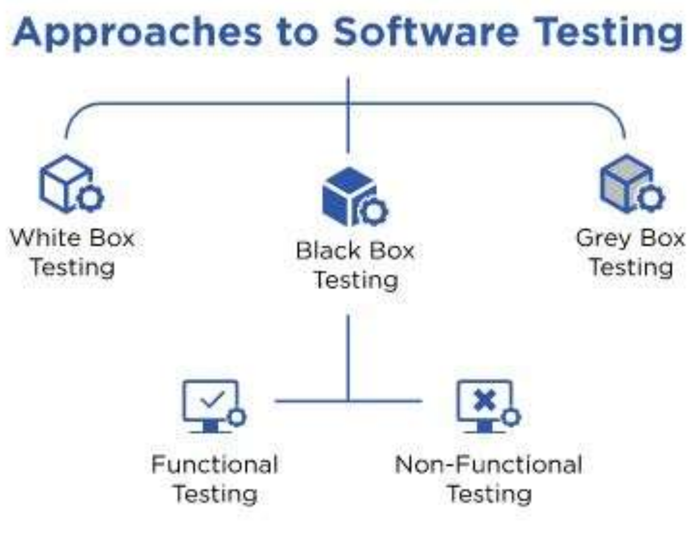

**Figure 2: The Above Figure Shows the Approaches to Software Testing [leewayhertz].**

### *1.4.1 White Box Testing:*

Testing is conducted on the source code by developers to check if the source code is working as expected or not. This type of testing is very economical and very helpful to reduce the defects as early as possible. Advantages and disadvantages of the white box testing is highlighted in the Table 1.

**Table 1: Advantages and Disadvantages of White Box Testing.**

| Advantages | Disadvantages |
|---|---|
| It can be performed at the initial stages | It can be complicated |
| It is more thorough | It required highly skilled resources |
| It allows us to find hidden defects | Tools to perform white box testing may not be readily available |
| It helps in code optimization | |
| Maximum coverage is ensured due to knowledge of code | |

### *1.4.2 Black Box Testing:*

Test engineers or experts from end-user perspective conduct this type of testing in order to check whether the application is working according to customer requirements. Functional requirements of the developed software are covered in the black box testing. The objective of the black box testing is the completeness of the customer business requirements coverage. Advantages and disadvantages of the white box testing is highlighted in the Table 2.

**Table 2: Advantages and Disadvantages of Black Box Testing.**

| Advantages | Disadvantages |
|---|---|
| It exposes inconsistencies in specifications | Test may be tough to design |
| There is no need to understand programming | Many bugs can go under detected |

### *1.4.3 Grey Box Testing:*

Grey Box Testing represents the method of combination of both white box and black box testing. The responsible tester for gray tester for gray box testing has limited access of code. For e.g. for a web application where the validation is made on the client side the systems can be affected when the JavaScript is disabled and the server get an invalid email. Advantages of Grey box testing is highlighted in the Table 3.

**Table 2: Advantages Associated with Grey Box Testing.**

| | |
|---|---|
| **Advantages** | It provides combined benefits of white box testing and black box testing |
| | Grey box testers can develop more intelligent tests |
| | Developers and testers have clear goals while testing |
| | The overall quality of the software is enhanced |

### *1.5 Software Development Lifecycle:*

The Software Development Life Cycle (SDLC) is a technique for developing high-quality software that includes well-defined procedures. The SDLC approach focuses on the phases of software development as follows:

- Required analysis
- Planning
- Software design such as architectural design
- Software development
- Testing
- Deployment

The Software Development Life Cycle, or SDLC, is a method for producing high quality, low-cost software in the least amount of time. SDLC is a well flow of stages that enables a company to swiftly generate high software that has been thoroughly tested and is ready for production.

It also outlines the new system's needs. It goes through the steps of analysis, planning, design, programming, testing, and deployment to generate the software. SLDC may reduce needless rework and after adjustments by predicting costly mistakes like forgetting to ask the end-user or customer for input. Figure 3 shows the software development lifecycle.

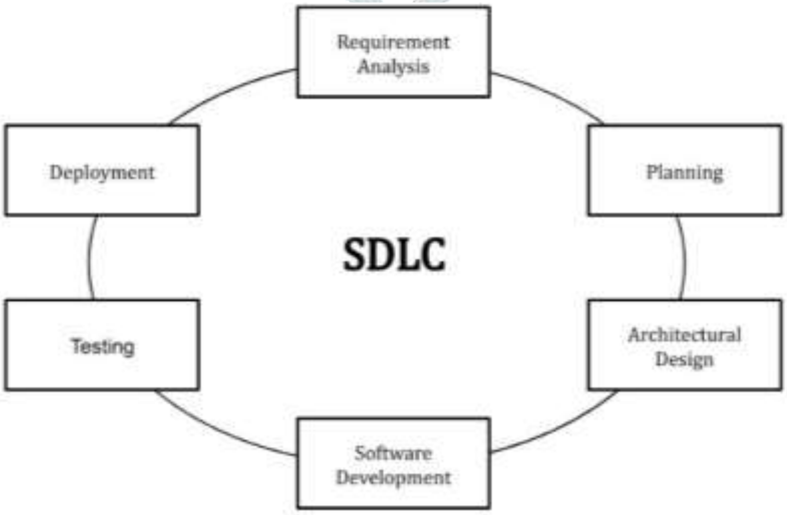

**Figure 3: The Above Figure Shows the Software Development Lifecycle [stackify].**

## 2. DISCUSSION

The author has discussed about the software testing, Software testing is the purpose of analysing software and identifying of the difference between the existing and the required conditions and to evaluate the features of the software. Testing measures the quality of the developed software. This fact supposes that there are defects in the functionality of the software waiting to be discovered. Testing should get a high priority during any software development effort. Software testing is an important part of software quality assurance. The value of testing may be shown in life-critical software (for example, flight control) testing, which can be extremely costly due to the possibility of schedule delays, cost overruns, or outright cancellation. A handful of principles that every software engineering student and faculty member should be familiar with are identified. For creating test cases, there are currently a variety of testing methodologies accessible. With the fewest amount of test cases feasible, this collection should assure maximum efficacy. The major objective of this work is to analyse and compare testing techniques in order to determine which one is superior at detecting software errors. The most essential thing for software testers to grasp during the testing process is how to condense a huge number of tests into a manageable test set and make informed judgments about which risks are necessary to test and which are not.

## 3. CONCLUSION

The author has concluded about the software evaluation, Software testing is the process of analysing programme and determining the differences between the current and desired circumstances, as well as evaluating the software's characteristics. Testing assesses the software's overall quality. This implies that there are flaws in the software's functionality that have yet to be uncovered. During every software development project, testing should be given top attention. Testing software is an essential aspect of software quality assurance. Life-critical software testing (for example, flight control) can be highly costly due to the risk of schedule delays, cost overruns, or outright cancellation. There are a few key concepts that every software engineering student and faculty member should be aware. There are presently a number of testing techniques available for building test cases. This collection should provide optimum efficacy with the fewest number of test cases possible. The author has also discussed about the approaches used by the testing techniques those are white box testing, black box testing and grey box testing.

### REFERENCES

[1] M. Kaur and R. Singh, "A Review of Software Testing Techniques," *Int. J. Electron. Electr. Eng.*, vol. 7, no. 5, pp. 463–474, 2014.

[2] P. Yadav and P. Kumari, "Review paper on software testing," vol. 1, no. 12, pp. 588–592, 2015.

[3] M. A. Jamil, M. Arif, N. S. A. Abubakar, and A. Ahmad, "Software testing techniques: A literature review," *Proc. - 6th Int. Conf. Inf. Commun. Technol. Muslim World, ICT4M 2016*, no. November, pp. 177–182, 2017, doi: 10.1109/ICT4M.2016.40.

[4] H. Babbar, "Software Testing : Techniques and Test Cases," *Int. J. Res. Comput. Appl. Robot.*, vol. 5, no. 3, pp. 44–53, 2017.

[5] A. A. Sawant, P. H. Bari, and P. . Chawan, "Software Testing Techniques and Strategies," *J. Eng. Res. Appl.*, 2012.

[6] K. Sneha and G. M. Malle, "Research on software testing techniques and software automation testing tools," 2018, doi: 10.1109/ICECDS.2017.8389562.

[7] M. E. Khan, "Different approaches to white box testing technique for finding errors," *Int. J. Softw. Eng. its Appl.*, 2011, doi: 10.5121/ijsea.2011.2404.

[8] S. Nidhra and J. Dondeti, "14 - Black Box and White Box Testing Techniques," *Int. J. Embed. Syst. Appl.*, 2012.

[9] R. H. Rosero, O. S. Gómez, and G. Rodríguez, "15 Years of Software Regression Testing Techniques - A Survey," *International Journal of Software Engineering and Knowledge Engineering*. 2016, doi: 10.1142/S0218194016300013.

[10] T. L. S. Brown, "Principles of software testing," *CrossTalk*, 2017.