An Analysis of Software Testing Processes

Kundan Kumar Pramanik, Assistant Professor
Department of Engineering & IT, Arka Jain University, Jamshedpur, Jharkhand, India
Email Id-kundan.parmanik@arkajainuniversity.ac.in

ABSTRACT: Software testing is the practice of carrying out a series of tests to verify that a piece of software is working correctly. Any program must be error-free in order to function correctly. Various testing methods are used to guarantee that any software or application produced is error-free, a few of which have been described in this article. When doing software testing, two essential and distinct stages are performed: verification and validation. These have been addressed in depth in this article, as well as the many actions that are carried out while performing them. There are many kinds of software testing, each requiring knowledge of the product's underlying structure, implementation, and design. Unit testing, black box testing, regression testing, user acceptability testing, integration testing, white box testing, and other functional and non-functional testing methods are among them. This paper has gone through each of them. This article explains why software tester is a good profession for students who don't have much programming experience. It explains how software testers have a promising future since new software is created every day that requires many testing methods.

KEYWORDS: Code, Error, Process, Software, Testing.

1. INTRODUCTION

Software testing is a procedure that consists of a series of actions that are carried out to ensure that a piece of software is functional. It is carried out to see whether it satisfies all of the client's or customer's needs and features. The software is tested in a variety of settings and on a variety of platforms to ensure that it functions properly and that it produces the desired results and meets the requirements of the client. Various procedures are carried out in order to discover flaws and defects in the software so that when it is given over to the customer, there are no bugs or errors that would disappoint the client. Various kinds of testing are carried out depending on the program requirements[1].

1.1 Types of Software Testing:

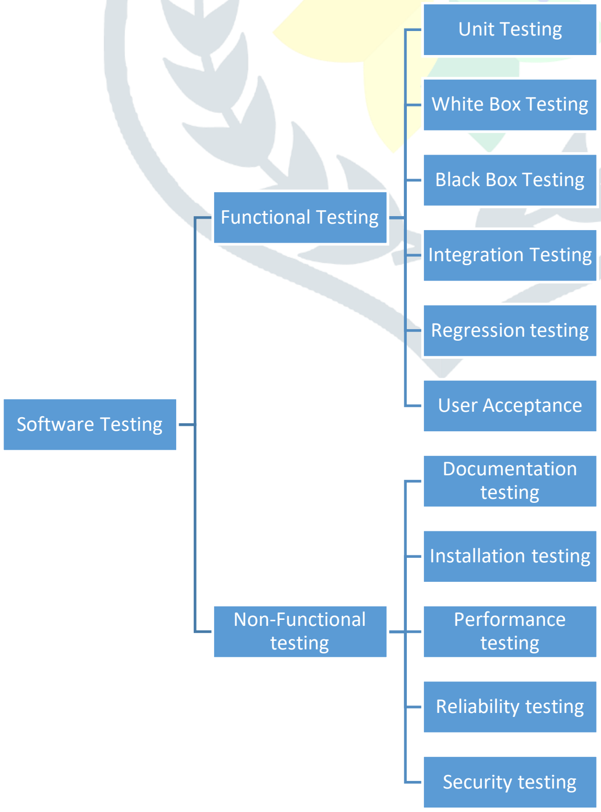

Figure 1: Flowchart showing various types of Software testing performed as per software needs.

Figure 1 shows a flowchart which contains various types of software testing which are performed as per the need of the software.

i) Functional Testing:

The process of evaluating a software application's functional characteristics is known as functional testing. Each and every aspect of the program should be evaluated during functional testing. The Software Tester should check to see whether the desired results were achieved. The primary goal of functional testing is to verify the output of a software program by giving enough input and comparing it to the functional requirements. It is unconcerned with the source code of the program. It examines the application's database, client/server connectivity, application programming interfaces (APIs), user interface, and other functions. This testing may be done manually or with the help of automation[2].

The primary focus of functionality testing is on system accessibility, mainline function testing, basic system usability testing, and error checking using different methods. Before non-functional testing, it is always done. The Business Requirements are the inputs for this testing. It explains what the product or program is capable of. The functional testing category encompasses a wide range of testing procedures. These are:

ii) Unit Testing:

For testing, the tiniest unit of code is used. A single line of code, a function, an interface, or a method may all be considered the smallest unit of code.

Programmers are in charge of this. The primary goal of this is to break down each part of the code and verify that each section works correctly, ensuring that each unit of the planned program acts as anticipated. It is carried out throughout the development stage.

Advantages:

- Because individual code units are evaluated in this phase before integration testing, problems may be discovered and corrected early on without affecting other source codes.
- It makes code maintenance easier since individual developers do it and strive to make the code as interdependent as possible, which reduces the likelihood of other source codes being impacted by a specific code[3].
- Because defects are identified and eliminated early in the development cycle, the cost of identifying and correcting mistakes is reduced.
- Since of this approach, debugging the code is simple because if any test case fails during unit testing, just the most recent changes need to be debugged.

iii) Integration Testing:

It is the process of collecting unit-tested components and constructing a program structure according to the design's specifications. This collection of components is mixed and tested to see whether they create the desired result. There are four kinds of testing: big-bang, top-down, sandwich, and bottom-up.

iv) Black box testing:

It's a kind of testing in which the tester has no knowledge of the system's underlying structure, design, or implementation. The testing is done from the point of view of the end user. It is used to test program behavior. Black box testing does not require coding skills. It is also not necessary to have prior experience with software implementation. Outer or external software testing is another name for it. It saves you time. Software testers are the ones who do it. Closed testing is another name for it. It is thought to be a very high degree of testing. It is not recommended or appropriate for algorithm testing. It generates reports with a low level of granularity. The tester interacts with the system's user interface by providing inputs and reviewing the resulting output without knowing how or where the inputs are processed[4].

Advantages:

- It has shown to be very efficient in big code segments.
- There is no requirement to access the code, and no programming expertise is needed, making it simple for people with little coding or programming experience.
- It separates the user's viewpoint from the developer's via clearly defined responsibilities.

Disadvantages:

- Because just a small number of test cases are performed, coverage is restricted.

- Because the tester does not know all of the information about the application, testing is inefficient.
- Creating test cases is very difficult.

v) White box Testing:

It is a kind of testing procedure in which, unlike black box testing, the person doing the testing is aware of the software's design and implementation. It necessitates that the tester be well-versed in coding. [6] Software developers are the ones who do it. It necessitates that the tester be familiar with software implementation. Internal testing is another name for it. It examines the system's interior workings. It's regarded as a low-level of testing. It's also known as code base testing, transparent box testing, glass box testing, or clear box testing. It concentrates on routes, branches, conditions, and the structure of the code[5]. It generates reports with a lot of detail. When compared to black box testing, it takes longer.

Advantages:

- It aids in code optimization;
- Extra lines that may lead to hidden mistakes are eliminated.
- Because the tester is familiar with internal processes and code, he or she may assist maximize coverage when designing test scenarios.
- Finding the kind of data that will assist in the appropriate testing of the software is easy.

Disadvantages:

- Costs rise as more experienced testers are needed to do this kind of testing.
- Numerous hidden faults that may create problems are often not discovered since many paths remain unexplored.
- Its upkeep is challenging due to the need for specialist tools such as code analyzers and debugging tools.

vi) Regression testing:

The program updates whenever anything new is introduced to the code or any other changes are made to the code. This testing is done to verify that the software functions correctly after all of the changes have been made. When a problem is discovered, modifications are made to the code to fix it, and regression testing is carried out[6].

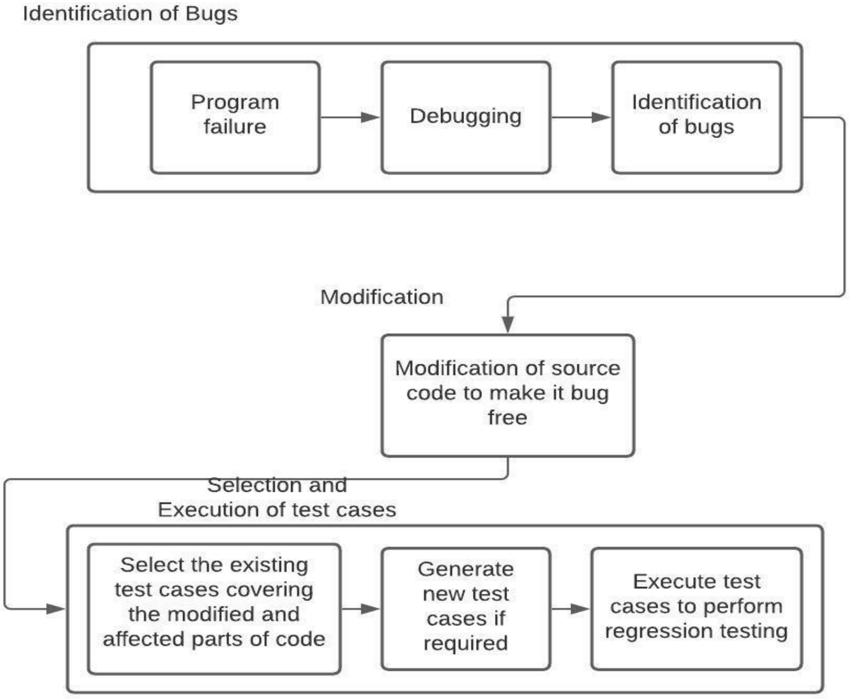
Identification of Bugs
Program failure
Debugging
Identification of bugs
Modification
Modification of source code to make it bug free
Selection and Execution of test cases
Select the existing test cases covering the modified and affected parts of code
Generate new test cases if required
Execute test cases to perform regression testing

Figure 2: Representation of Process of regression testing involving identification of bugs, Modification of code to remove bugs and then selection and execution of test cases.

Figure 2 represents the process of regression testing in which when the program fails debugging is done to find out the errors and defects then modifications are done in code to remove the bugs and then the existing test cases which got affected by the bugs and code changes are modified. New test cases are also generated if needed and then these test cases are executed to perform regression testing.

vii) User Acceptance testing:

It is carried out by the user, client, or customer who has purchased the program. It is done to verify that all of the product's criteria are fulfilled as requested by the client. It is completed at the end of the process. The primary goal of this is to verify the end-to-end business flow.

viii) Non-Functional Testing:

It is used to test a software's non-functional characteristics, such as performance, reliability, and usability. It is done to improve the product's efficiency, usability, portability, and maintainability. It is used to improve the processes for setting up, monitoring, installing, managing, and executing software. This testing is unconcerned with system testing, aesthetic flaws, or misspellings. Security, portability, availability, interoperability, efficiency, reusability, integrity, scalability, dependability, adaptability, survivability, and usability are some of the criteria addressed by non-functional testing[7]. In software testing, there are two main stages, which may be referred to as procedures or ideas. Verification and validation are two terms that are used interchangeably. These are two crucial aspects of software testing.

a) Verification:

It is carried out prior to validation. It is a procedure for verifying software, as the name implies. The design of software, code of software, documentation connected to software, and structure of software are all examined and verified throughout the verification process. There is no requirement to implement or run the software code during verification. It focuses on detecting flaws and mistakes early in the development process so that software runs smoothly and without errors. With comparison to validation, the cost of error identification and troubleshooting is minimal or free in verification. Quality Assurance (QA) engineers conduct the verification. Inspection of software code, verifying design reviews, reviewing system requirements, desk-checking, walkthroughs, checking business requirements, and so on are all part of the verification process.It is carried out to verify that the produced product or program is functioning properly. Static testing is another name for it. Humans do the task manually.

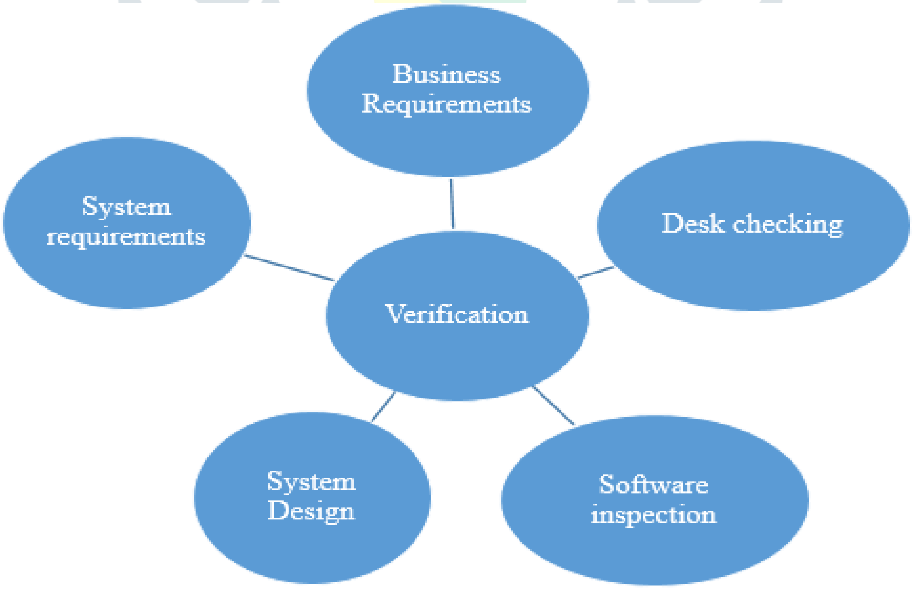

Figure 3: Different activities performed in Verification including checking of business and system requirements, system design, software inspection and desk checking.

Figure 3 shows different activities which are performed while doing verification of any software. It includes inspection of the software, checking business requirements including all the documents, checking system requirements, checking system design including both high level and low level design, walkthroughs and desk checking.

b) Validation:

It is carried out after validation. It is a procedure for validating software, as the name implies. It is carried out to see whether the actual product planned and produced meets all of the client's needs and standards. It's similar to dynamic testing. It requires coding expertise. It involves code implementation and execution. Validation detects and corrects bugs and mistakes that were missed throughout the verification phase. Both the Quality Assurance and Testing teams are in charge of validation. User acceptability testing, system testing, and functional testing are all part of the validation process integration. Validation testing includes quality control. Non-functional testing, white box testing, and black box testing are among the techniques utilized. It is carried out on a computer.

2. LITERATURE REVIEW

Sawant A et al. discussed Software Testing Techniques and Strategies in which the article discusses software testing, the necessity for software testing, and the objectives and concepts of software testing. Furthermore, it discusses various software testing methods and tactics. Finally, it compares and contrasts software testing and debugging[8].

Garousi V et al. discussed Software Testing practices in Canada in which they explained how Software testing is an essential part of the software development process. Researchers presented the findings of a regional survey of software testing methods among practitioners in the Canadian province of Alberta in a previous study published in 2009. In this post, we describe the survey's concept, implementation, and findings. The outcomes of the study show some significant and intriguing information regarding software testing methods in Canada. They compare the findings of this poll to those of other comparable studies done in the United States, Sweden, and Australia, as well as two prior Alberta-wide surveys, including our 2009 study. Testing experts in Canada and across the globe will be interested in the findings of their survey. It will also help academics in terms of monitoring current trends in the software testing business and identifying areas of strength and weakness, which should ideally inspire further industry-academic cooperation in this area[9].

Kang S et al. discussed Architecture-Based Software Testing in which they explained how Many researchers believe that the concept of test architecture should be used in software testing to deal with the task of testing complex software because it can support test architecture design and test case generation based on test architecture, as well as efficient fault detection and error locating. However, test architecture and how it may be used for software testing are not well understood in contemporary software testing theory and practice. To address this issue, this article explains the fundamental ideas and principles of architecture-based software testing and how to apply them to software testing[10].

3. DISCUSSION

Software developer, Software Tester, Web developer, Front-end developer, Backend developer, Project Manager, Data Scientist, Software Engineer, Quality Assurance engineer, Database Manager, Business Analyst, Database Developer, Applications Developer, Game developer, Cloud engineer, Multimedia programmer, Full stack developer are some of the job titles available in the field of computer science. Each of these tasks requires familiarity with a variety of tools, technology, and applications. Strong coding abilities, for example, are needed for the job of software developer. Students who excel in programming are chosen for the position of software developer. It is necessary to have knowledge of the cloud, its kinds, functioning mechanisms, and many other methods in order to perform the job of cloud engineer. Analytical and aptitude abilities are also important for the job of software tester.

Various testing methods are used in software testing to ensure that the final product provided to the client meets his requirements and exceeds his expectations. Positive testing, negative testing, functional testing, system testing, performance testing, integration testing, black box testing, non-functional testing, white-box testing, regression testing, security testing, efficiency testing, reliability, scalability testing, load testing, volume testing, and user acceptance testing are all examples of software testing techniques. In this article, a few of these testing methods have been described, along with their processes, benefits, and drawbacks. Any of these testing methods or techniques are critical for the effective completion of any software project since they ensure that all mistakes, bugs, or flaws are identified and corrected. These methods ensure that the final product is bug-free and ready to provide optimal outcomes.

4. CONCLUSION

Program testing is a professional role that entails creating numerous test cases and utilizing them to determine whether or not a software or application is functioning correctly at various phases of its development. There are many testing methods that are utilized or conducted according to their requirements while creating any product.

In this article, just a few of them are addressed. In this article, the primary goal and methods of software testing are discussed. This article discusses both functional and non-functional testing. This article discusses several kinds of functional testing, including unit testing, black box testing, user acceptability testing, regression testing, integration testing, and white box testing. Two significant testing methods are white box testing and black box testing. These are described in depth, along with their benefits and drawbacks. There are many concepts that should be kept in mind while conducting any testing procedure, and these ideas have been described in this article. As more software and applications are created and produced each day, the developers and whole project team want them to be error-free and as optimized as possible according to client requirements, software testing is becoming an increasingly important profile. Students that choose to become software testers will have a bright career since no project can be finished without these procedures. Thus, students who are not excellent at programming but have strong analytical abilities are suitable for this position and may guarantee their future with a little hard effort.

REFERENCES

[1] D. Spinellis, "State-of-the-Art Software Testing," *IEEE Software*. 2017, doi: 10.1109/MS.2017.3571564.

[2] K. Wnuk and T. Garrepalli, "Knowledge management in software testing: A systematic snowball literature review," *E-Informatica Software Engineering Journal*. 2018, doi: 10.5277/e-Inf180103.

[3] M. Kassab, J. F. Defranco, and P. A. Laplante, "Software Testing: The State of the Practice," *IEEE Softw.*, 2017, doi: 10.1109/MS.2017.3571582.

[4] S. Nidhra, "Black Box and White Box Testing Techniques - A Literature Review," *Int. J. Embed. Syst. Appl.*, 2012, doi: 10.5121/ijesa.2012.2204.

[5] ~ Wwjmrd, "A Research Paper on White Box Testing," *Int. J. Peer Rev. J. Ref. J. Index. J. UGC Approv. J. Impact Factor*, 2018.

[6] G. Rothermel, R. H. Untcn, C. Chu, and M. J. Harrold, "Prioritizing test cases for regression testing," *IEEE Trans. Softw. Eng.*, 2001, doi: 10.1109/32.962562.

[7] S. Dave, H. Patel, and A. Prajapati, "Non-Functional Testing In Agile Development," *Int. J. Innov. Res. Technol.*, 2018.

[8] A. A. Sawant, P. H. Bari, and P. . Chawan, "Software Testing Techniques and Strategies," *J. Eng. Res. Appl.*, vol. 2, no. 3, 2012.

[9] V. Garousi and J. Zhi, "A survey of software testing practices in Canada," *J. Syst. Softw.*, 2013, doi: 10.1016/j.jss.2012.12.051.

[10] J. Lee, S. Kang, and C. Keum, "Architecture-Based Software Testing," *Int. J. Softw. Eng. Knowl. Eng.*, 2018, doi: 10.1142/S0218194018500031.