International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2019): 0.28 | SJIF (2019): 7.583

Software Testing Techniques

Bhupinder Paul Singh Sahni

Email: *bhupinder.sahni[at]gmail.com*

Abstract: ***Software Testing involves experimentally and systematically checking the correctness of software with respect to the defined specifications or requirements. It is an activity that is performed for evaluating software quality and also for improving it. There are different Software Testing techniques that can be used at different stages to improve the quality of the software. This paper discusses different functional and non - function software testing techniques that can be used to make the software optimal and bug - free according to the requirements.***

Keywords: Software Testing; Software Quality Assurance; Software Development Life Cycle; Functional Testing; Integration Testing; System Testing; Non - Functional Testing; Performance Testing; Security Testing; Cost of Testing

1. Introduction

In Software Development Life Cycle (SDLC), Software development involves developing software against a set of requirements. In order to verify and validate that the software that has been built meets these specifications, Software testing is performed. Software testing refers to the process of evaluating the software against business requirements, with the intention to find out issues in it [1]. With the growing complexity of today's software applications, Software testing is an inevitable part of the Software Development Lifecycle.

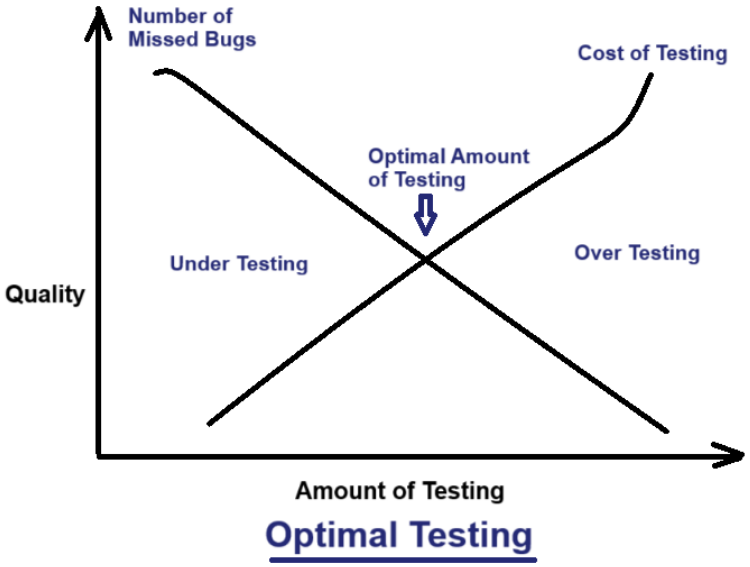
Software Testing is not just about Testing the software based on new requirements, but also that the new requirements coded should not impact the existing software application. As the amount of maintenance and upgrade of existing systems grows, a significant amount of testing is also needed to verify systems after changes are made [2]. This is to ensure that the quality of the existing system is also intact with the introduction of new features or requirements.

There are different software testing techniques that can be used at different stages of software development life cycle and based on customer and technical requirements. In this paper, we will study the importance and goal of Testing, when can we say that we have done enough testing that it will not lead to critical bugs in production, and different functional and non - functional testing techniques.

1.1 Goal of Testing

Software Testing is the process of executing a program or a system with the intent of finding errors or Bugs. Bugs in the Software does not mean that computer programmers have been careless or not adhering to the standards and thus resulting in bugs in the Software, but in most scenarios, there are very complex applications or complex requirements which makes it very hard to think about all the possible scenarios and its impact on the system. So, software testing is done by an independent group or individuals which will just think about what is being developed and match it with different scenarios that the requirements can lead to. Software bugs may almost always exist in any software with a complex application or project.

Software Testing is potentially endless. We cannot test till all the defects are unearthed and removed. At some point, we must stop testing and ship the software. The question is when to stop Testing. So, Software Testing can also be defined as a risk - based activity. The testing cost and errors can be found in a relationship as shown in the figure below. It is apparently exhibited in the figure below that as amount of testing is increased, cost of Testing also rises. On the contrary, missed number of Bugs is higher if the amount of Testing done is less. So, according to risk - based Testing, the Testing goal is to do the optimal amount of testing so that extra testing efforts can be minimized [3].



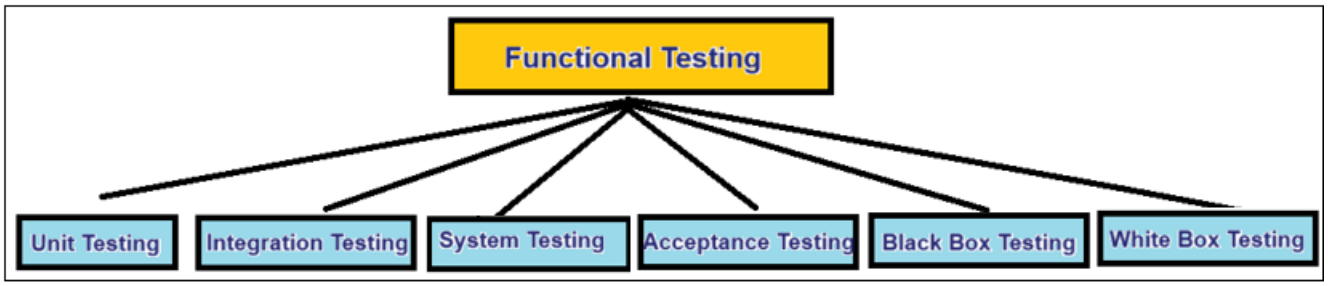
Anything below optimal Testing would be under testing leading to critical or high priority bugs, and anything above optimal testing would lead to a smaller number of bugs but would cost more. Realistically, testing is a trade - off between budget, time, and quality.

1.2 Software Testing Techniques

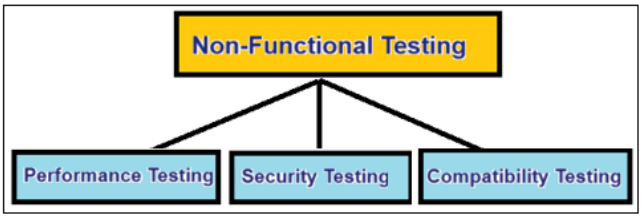
There can be many ways that Software Testing techniques can be characterized into. But the most common way of characterizing software testing technique is:

- Functional Testing
- Non - Functional Testing

Functional Testing focusses on matching software behavior with the user requirements. The main quality factor in software is to meet its required functionality and behavior. There are different types of Functional testing techniques which could be performed at various levels of testing. Some of the Functional testing techniques are shown in the figure below.



Similarly, Non - Functional Testing is where testers are testing non - functional requirements, which may not be mentioned in the use cases, but are essential for software to perform under different conditions. Some of the Non - Functional Testing techniques are shown in the figure below.



In the next sections, we will study different types of Functional and Non - Functional Testing Techniques.

a) Functional Testing Techniques

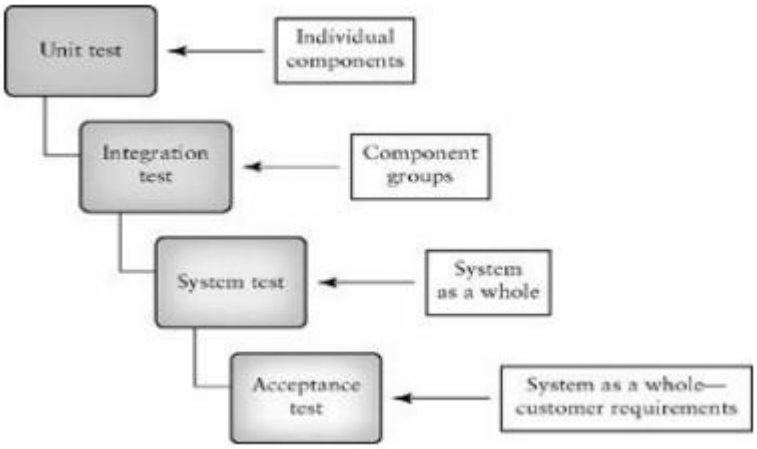
Unit Testing: Unit Testing is a simple testing technique because the smallest testable unit of the code is tested. It tests the basic unit of software, which can be a module or component and individual parts can be tested individually, when necessary. Unit testing is done by the developer as proper knowledge about the core programming design is required [4].

Integration Testing: The objective of integration testing is to test the communication and integration between individual units or components of a system to make sure data is flowing across various components correctly. This is done following either a top - down approach or bottom - up approach [4]. In a top - down approach, higher - level components are integrated first, followed by lower - level components. In a bottom - up approach, lower - level components are integrated first, followed by higher - level components.

System Testing: System testing is a level of software or hardware testing where testing is conducted on a complete, integrated system to assess the system's compliance with its specified requirements. The purpose of system testing is to evaluate that the overall software system developed meet the specified requirements [3].

Acceptance Testing: Acceptance Testing, also known as User Acceptance Testing, is done when the completed system is handed over from the developers to the customers or users. The purpose of acceptance testing is to give confidence that the system is working rather than to find errors [5]. User acceptance testing is considered to be one of the most important testing techniques by the user before the system is finally delivered or handed over to the end user.

Different Functional Testing techniques can also be easily characterized as below [6]:

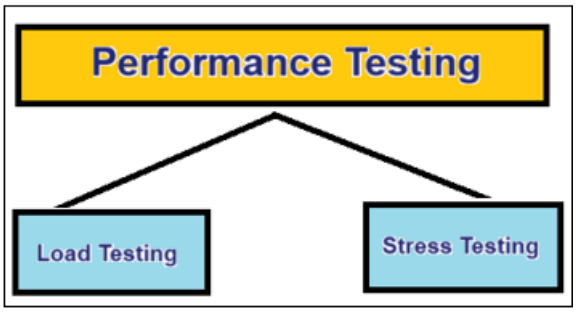


Black Box Testing: Black box testing is performed to ensure output of application is as correct for all various types of positive and negative inputs. There are various types of Black box testing types like Equivalence Class partitioning, Boundary value analysis, error guessing etc. [4]. In black box testing, only the outside of the system under test is known to the tester [7].

White Box Testing: White box testing deals with the internal working of code to ensure there is no redundant code written in software. This involves testing of line of code, program, flow, logic, loop, structure, functions, class communication testing and other internal testing of program [4]. In white box testing, the internal structure of the system is also known, and this knowledge can be used by the tester [7].

b) Non - Functional Testing Techniques

Performance Testing: Evaluation of the performance of any software system includes resource usage, throughput and stimulus response time [8]. Performance Testing tests performance of software under all favorable and non - favorable conditions. The goal of performance testing can be performance bottleneck identification, performance comparison and evaluation, etc. The typical method of doing performance testing is using a benchmark program, workload or trace designed to be representative of the typical system usage [9]. Performance Testing can be divided into 2 categories – Load Testing and Stress Testing.



Load testing is performed to determine whether the given system is able to handle the anticipated number of users or not. Load testing is also used for checking an application against heavy load or inputs such as testing of website in order to find out at what point the website or applications fails or at what point its performance degrades [8].

Stress testing is performed to find the upper limits of capacity within the system. Extreme load is given to the application to determine the robustness of the system [4].

Load and Stress Testing is often used to test the whole system, rather than the software alone.

Security Testing: Software quality, reliability and security are tightly coupled. With the development of the Internet, software security problems are becoming even more severe. Issues in the Software can be exploited by intruders for their benefits. Security testing of software is important as to protect the information, services, skills and resources of adversaries and the cost of potential assurance remedies. The system is tested for areas like authentication, authorization, information leakage and different kinds of threats. Security testing makes sure that only the authorized personnel can access the program and only the authorized personnel can access the functions available to their security level [10]. Simulated security attacks can be performed to find vulnerabilities [11].

Compatibility Testing: Compatibility testing is a kind of non - functional testing technique. This testing is conducted on the application to evaluate the application's compatibility with the computing environments, platforms, browsers, devices, and operating systems [1]. It checks the compatibility of the developed system with the various other components such as hardware, additional software, DBMS and operating system, etc. [3].

2. Conclusion

Quality is critical to any software development project. Complete Testing can never be feasible for complex projects, but we need to consider risk - based testing to have the optimal level of testing. The intent of this paper was to study different Software Testing techniques that can be performed at different phases of software development life cycle to ensure quality of the end product at an optimized cost.

References

[1] A. Sawant, P. Bari, P. Chawan, “Software Testing Techniques and Strategies”, International Journal of Engineering Research and Applications (IJERA), vol.2, issue 3, pp.980 - 986, 2012.

[2] J. Marciniak, “Encyclopedia of software engineering”, John Wiley & Sons, Inc., 2002.
[3] N. Anwar, S. Kar, “Review paper on various software testing techniques & strategies.”, Global Journal of Computer Science and Technology, vol.19, issue 2, pp.43 - 49, 2019.
[4] Hooda, Itti, and Rajender Singh Chhillar. "Software test process, testing types and techniques. " International Journal of Computer Applications 111.13 (2015).
[5] Luo, Lu. "Software testing techniques. " Institute for software research international Carnegie mellon university Pittsburgh, PA 15232.1 - 19 (2001): 19.
[6] Chauhan, Rasneet Kaur, and Iqbal Singh. "Latest research and development on software testing techniques and tools. " International Journal of Current Engineering and Technology 4.4 (2014): 2368 - 2372.
[7] Tretmans, Jan. "Testing techniques. " Lecture notes, University of Twente, The Netherlands 1 (2002): 1.
[8] M. Khan, “Different Forms of Software Testing Techniques for Finding Errors”, IJCSI International Journal of Computer Science Issues, vol.7, issue 3, no.1, pp.11 - 16, May 2010.
[9] Filippos I. Vokolos, and Elaine J. Weyuker, “Performance testing of software systems”, Proceedings of the first international workshop on Software and performance, pp.80 - 87, 1998.
[10] Software testing - Brief introduction to security testing by Nilesh Parekh published on 14 - 07 - 2006 available at http: //www.buzzle. com/editorial/7 - 14 - 2006 - 102344. asp
[11] Pan, Jiantao. "Software testing. " Dependable Embedded Systems 5.2006 (1999): 1.