



A REVIEW PAPER ON LEVELS, TYPES & TECHNIQUES IN SOFTWARE TESTING

Ms. Ankita Sethi
Assistant Professor
IPEM Group of Institutions

Abstract-Software testing is a discipline of software engineering whose aim is to find the presence of errors in the software. Software testing starts with the initial phase of software development life cycle. It is a myth that software testing shows the absence of errors. It is actually performed to show the presence of errors. The software testing leads to minimize errors lower down maintenance and cut down software costs. There are various techniques through which the process can be achieved. Testing is done to ensure that the product which is going to be produced will achieve the success. This paper focuses on the review and comparison of software testing levels, types & techniques in order to find out the most suitable techniques for Software Testing.

Keywords: Software Testing, SDLC, Level of Testing, Testing Technique, Testing Process.

I. INTRODUCTION

According to the definition [1], Software testing is the process of executing the programme with the intent of finding the error. According to *ANSI/IEEE 1059* standard [2, 3], A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item is known as Testing.

It is myth that testing is one of the phase in SDLC, but the fact is that testing starts with very first phase of SDLC. In simple words, testing is process of locating errors in the programme.

Software Testing is executing the software in order to (1) perform verification (2) to detect errors, and (3) to achieve validation.

1. Verification is the process of checking the software with respect to the specification. [Verification: Are we making the product right?]

2. Error Detection: It is the process of deliberately performing the wrong inputs in order to check the system's performance.

3. Validation is the process of checking the software with respect to the customer's expectation. [Validation: Are we making the right product?]

Basic Terminology

Mistake, Error: Human make mistake. A good synonym is error. It difference between the actual output and the expected output.

Bug: When developer makes the error while coding, it is known as bug. It is the state which is responsible for the failure of the specific function.

Fault : It is the representation of the error, where representation is the mean of expression that may be

diagrams, flow charts etc. It doesn't stop the execution of the programme instead it produces wrong output.

Failure : Failure occurs when fault is executed.

Software Testing Objectives :

The primary objective of the software testing is to provide quality product with reference to the reliability estimation and entire verification and validation of the product.

The secondary objective of testing includes :

- Executing a program with the intent of finding errors.
- To produce a test case which is capable of finding the undiscovered error as yet..
- Better the software can be controlled more the testing can be automated and optimized.

II. LITERATURE REVIEW

In this section, I will summarize the work which has already done in the field of Software Testing. According to "The Theory of Software Testing" Adtha. Lawanna Department of Information Technology, Faculty of Science and Technology Assumption University, Bangkok, Thailand (Jul. 2012)[4], Testing is the means of showing the presence of errors in the programme which can either be performed manually or automatically. It also includes the basic terminology of testing such as automated testing, failure, testing team, wrong test case selection.

This paper focuses on the process that should be followed to test the performance of new software and the entire system.

The conclusion of the paper is the complete view of three phases for software testing: preliminary testing, testing and user acceptance testing.

Abhijit A. Sawant et al [5] stated different Software testing techniques and different software testing strategies along with the difference between software testing and debugging. InMohd.EhmerKhan,"Different Forms of Software Testing Techniques for Finding Errors,"IJCSI International Journal

of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010[6], the author stated that it is very difficult to find out all the errors in the programme.

According to Jovanovic and Irena, "Software Testing Methods and Techniques," May 26, 2008.[7], there are several testing methods and techniques which can be used in different situations to get a suitable outcome.

J. A. Whittaker[8] stated a four phase approach to see the process through which a bug hides from testing.

III. TESTING TECHNIQUE

The importance of software testing to software quality cannot be overemphasized. After the development of the code, it is mandatory to test the software to identify all the errors and they must be debugged before the release of the software. Although it is impossible to identify and debug all the errors in the large software but at every phase it is tried to remove all the errors as possible. Testing helps in finding the bugs, it cannot conclude that the software is bug free.

We broadly categorized testing techniques into two parts:

1. Static Testing
2. Dynamic Testing

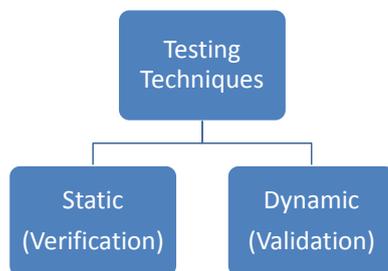


Fig 1. Testing Techniques

A. Static Testing: It refers to the method of testing where the code is not executed. It does not require highly skilled professionals since actual execution of the code is not done in this process. It starts with the initial phase of SDLC hence it is also known as verification testing. The main objective of static testing is to enhance the quality of software products by helping software professionals to identify and resolve their own errors early in the software development process. Static testing is performed on the documents like SRS, design documents, source code, test suites, and web page content. It is performed before code deployment. As a result it provides the evaluation of code as well as of documentation. The Static testing techniques include:

Inspection: It is primarily done to locate defects. Moderators conduct the code walkthrough. In this type of formal review a checklist is prepared in order to check the work document..

Walkthrough: It is not a formal process. Authors lead this process. To achieve a common understanding and to gather feedback, the Author guide the participants through the document according to his or her thought process. It especially useful for higher level documents like requirement specification, etc.

Technical reviews: In this type of static testing a technical round of review is conducted to check if the code is made according to technical specifications and standards. Generally the test plans, test strategy and test scripts are reviewed here.

Informal reviews: Static testing technique in which the document is reviewed informally and informal comments are provided.

B. Dynamic Testing : Dynamic Testing is a kind of software testing technique in which the dynamic behavior of the code is analyzed. In dynamic testing, also known as validation where the actual code is considered. It requires the highly skilled professional with the proper domain knowledge. Dynamic testing involves testing the software for the input values and output values are analyzed.

The Dynamic Testing are divided into two categories:

- Functional Testing or Black box testing
- Structural testing or White Box testing

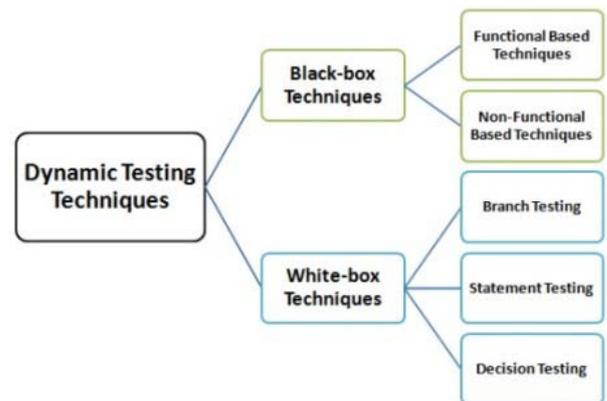


Fig 2: Dynamic Testing Techniques

	Black Box Testing	White Box Testing
1	In this type of testing, the tester need not to have the knowledge of particular programming language.	In this type of testing knowledge of programming is must .
2	In this type the tester are independent.	In this type, the testing is performed by the developer or the programmer.
3	It is basically concerned with output and input. It checks the	It is basically concerned with the code. It checks the code and try to find

	output with respect to given input .	error in it.
4	In can be performed even if the tester is not technically sound .	software developers perform this type of testing as the domain knowledge is a must
5	In this type of testing testers emphasize on the functionality of the system.	In this type of testing programmers emphasize on the structure or code of the system.
6	The code of the structure is hidden that is why it is known as black box testing. It is also known as Functional Testing.	The code of the structure is visible to the programmer that is why it is known as White box testing. It is also known as Structural Testing. In this the entire structure is visible to the developer and it check the code and generate the result.

C. Non-Functional Testing Techniques:

It is a type of testing which is performed to test various attributes of a system like stress, load etc.

Non-Functional testing is performed at all test levels. It is basically concerned with the non-functional requirements and is modeled specifically to assess the readiness of a system according to the various criteria which are not covered by functional testing[9]



Fig 3. Few Examples of Non-Functional Testing Techniques

- a) **Performance Testing:** It assesses the entire performance of the system. It is used to evaluate the system's performance under a specific work load.
- b) **Load testing:** A load test is performed to ensure the load taking capacity of a particular system. Load testing is performed to determine the behavior of the system under normal as well as peak load condition.
- c) **Endurance Testing.** It is the type of testing performed to determine the system's behavior after a particular

time. For eg. A system is working perfectly fine in initial first hour but the performance decreased after three hours of execution.

- d) **Stress Testing:** It is performed to determine the system performance beyond normal operational capacity, often to a breaking point. It is concerned with system's load taking capacity.
- e) **Security Testing:** Security testing is done to assess that the system is safe or not. It is a process which is concerned with the fact that an information system protects data and maintains functionality as intended.
- f) **Recovery Testing:** Recovery testing is performed to check the recovery of the system after the crash or hardware failure. In this type the software is forced to fail under a given circumstance and then finally the recovery is tested.
- g) **Compatibility Testing:** Compatibility testing is concerned with checking the system's compatibility with rest of the environment. It checks the compatibility of the developed system with the various other components such as hardware, other software, DBMS and operating system etc.

IV. CONCLUSION

After review I would like to conclude that Software testing is a basic activity of SDLC. We can never say that a product is "Perfect". Testing is a never ending process. Testing only shows the presence of errors not the absence. Manual testing is quite difficult and hence cost effective whereas testing cost can be reduced using automated testing tools.

V. REFERENCES

- [1]. Glenford J. Myers, "The Art of Software Testing, Second Edition" Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
1.1. P. Mathur, "Foundation of Software Testing", Pearson/Addison Wesley,2008.
- [2]. IEEE Standard 829-1998, "IEEE Standard for Software Test Documentation",pp.1-52, IEEE Computer Society, 1998.
- [3]. "The Theory of Software Testing" Adtha. Lawanna Department of Information Technology, Faculty of Science and Technology Assumption University, Bangkok, Thailand
- [4]. "Software Testing Techniques and Strategies" Abhijit A. Sawant¹, Pranit H. Bari² and P. M. Chawan³ Department of Computer Technology, VJTI, University of Mumbai, INDIA.
- [5]. Mohd. Ehmer Khan,"Different Forms of Software Testing Techniques for Finding Errors,"IJCSI International Journal of Computer Science Issues,Vol. 7, Issue 3, No 1, May 2010
- [6]. Jovanovic and Irena,"Software Testing Methods and Techniques," May 26,2008.
- [7]. J. A. Whittaker, "What is Software Testing? And Why Is It So Hard?" IEEE Software, January 2000, pp. 70-79[hard software testing]
- [8]. <http://reqtest.com/testing-blog/functional-vs-non-functional-testing/>