© INTERNATIONAL JOURNAL FOR RESEARCH PUBLICATION & SEMINAR
ISSN: 2278-6848 | Volume: 07 Issue: 04 | July - September 2016
Paper is available at www.jrps.in | Email : info@jrps.in

STUDY AND ANALYSIS OF TESTING TECHNIQUES

Neha[1], Abid khan[2], Ruchi Parashar [3]
[1] Research Scholar, Department of CSE, Al-Falah University, FARIDABAD Haryana, INDIA, silent.angel02@gmail.com
[2]Research Scholar, Department of CSE, Al-Falah University, FARIDABAD Haryana, INDIA, rohu123123@gmail.com
*[3] Research Scholar, Department of CSE, Al-Falah University, FARIDABAD Haryana, INDIA, ruchiparashar4*00*@gmail.com*

Abstract- Software testing provides a mean of reduce errors, cut maintenance and overall software costs. Numerous software **development and testing methodologies, tools, and techniques have emerged over the last few decades promising to enhance software quality. While it can be argued that there has been some improvement it is apparent that many of the techniques and tools are isolated to a specific lifecycle phase or functional area. One of the major problems within software testing area is how to get a suitable set of cases to test a software system. This set should assure maximum effectiveness with the least possible number of test cases. There are now numerous testing techniques available for generating test cases. In this paper, the three most prevalent and commonly used software testing techniques for detecting errors are described and compared, they are: white box testing, black box testing and hybrid testing.**

Keywords- Software testing, level of testing, testing techniques, testing process.

I. INTRODUCTION

Testing is the process of executing the program with the intend of finding faults. It is different from debugging. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. It is an expensive and challenging activity. The purpose of testing can be quality assurance, verification and validation. Testing can be used as generic metric as well. An important point to always keep in mind is that what makes a test a “good” one does not have a unique answer, but it changes depending on the context, on the specific application, and on the goal for testing. Software testing is an important phase of software development cycle. It is the process of finding the errors while executing a program. Software testing is a process of verifying and validating that a software application or program works as per the user expectations. It is used to find the important defects, flaws or errors in the application code that must be fixed. Software testing is the process to testing the software product. Effective software testing will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, more accurate, and reliable results. Ineffective testing will lead to the opposite results, low quality products, unhappy users, increased maintenance costs, unreliable and inaccurate results. Software testing was considered only a debugging process for removing errors after the development of software. Software testing consists of the dynamic verification of the verification of the behavior of a program on a finite set of test cases, suitably selected from the usually infinite executions domain, against the specified expected behavior.

In 1978, G. J. Myers realized the need to discuss the techniques of software testing in a separate subject. He wrote the book *THE ART OF SOFTWARE TESTING* which is a classic work on software testing. Myers discussed the psychology of testing and emphasized that testing should be done with a mindset of finding errors and not to demonstrate that errors are not present.

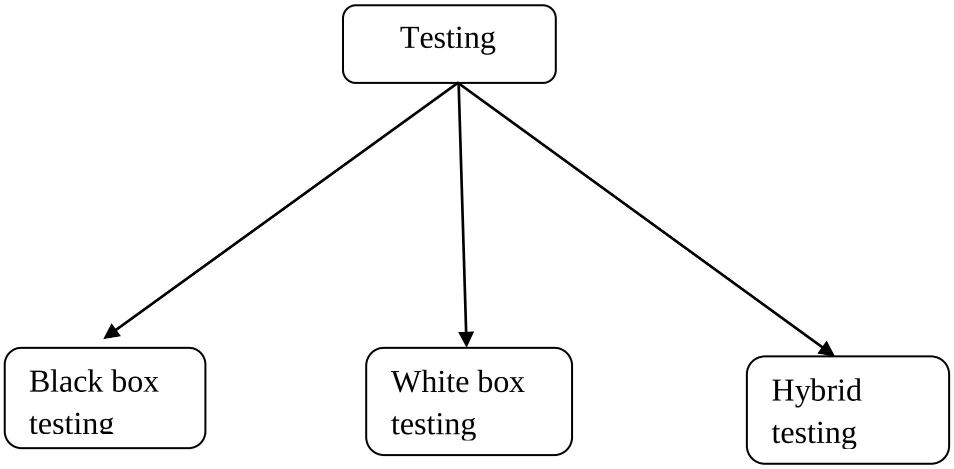

Fig.1 Different Testing Techniques

II. GOAL OF TESTING

The objective of testing is to find problems and fix them to improve quality

There are four main objectives of software testing:-

1) Demonstration: It demonstrates functions under special conditions and shows that products are ready for integration or use.

2) Detection: It discovers defects, errors and deficiencies. It determines system capabilities and limitations, quality of components, work products and the system.

3) Prevention: It provides information to prevent or reduce the number of errors clarify system specifications and performance. Identify ways to avoid risk and problems in the future.

4) Improving Quality: By doing effective testing, we can minimize errors and hence improve the quality of software [4].

III.DIFFERENT TESTING TECHNIQUES

A. *Black Box Testing*

Black Box Testing is based on the requirements specifications and there is no need to examining the code in black box testing. This is purely done based on customers view point only tester knows the set of inputs and predictable outputs. [9][5]

1) *Equivalence Partitioning*: Equivalence partitioning in black box testing method that divides the input domain of a program into classes of data from which test cases are derived. Equivalence partitioning strives to define a test case that uncovers classes of errors, thereby reducing the total number of test cases that must be developed. Test case design for equivalence partitioning is based on an evaluation of equivalence classes for an input condition. An equivalence class represents a set of valid or invalid states for input conditions. An input condition is a specific numeric value, a range of values, a set of related values, or a Boolean condition.

2) *Boundary Value Analysis*: A greater number of errors occurs at the boundaries of the input domain rather than in the "center". It is for this reason that boundary value analysis has been developed as a testing technique. BVA leads to a selection of test cases that exercise bounding values. BVA leads to the selection of test cases at the "edges" of the class. Boundary condition means, an input value may be on the boundary, just below the boundary or just above the boundary.

3) *Cause-Effect Graph*: A cause represents a distinct input condition that brings about an internal change in system. An effect represents an output condition, a system transformation or a state resulting from a combination of causes.

4) *All Pair Testing*: In this technique, test cases are designed to execute all possible discrete combinations of each pair of input parameters. Its main objective is to have a set of test cases that covers all the pairs.

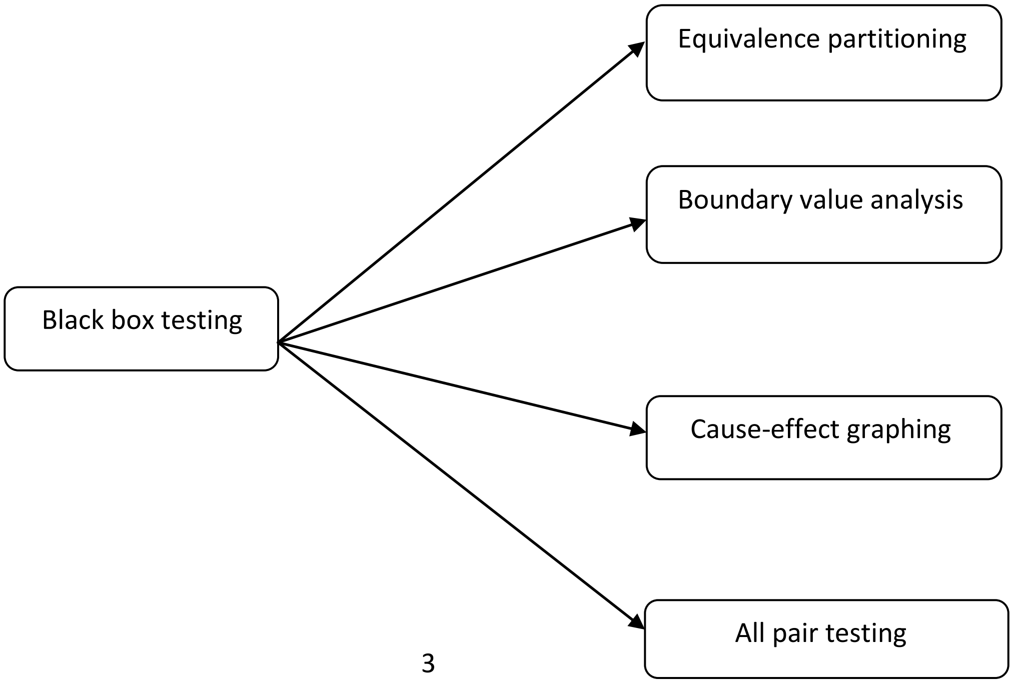

Fig.2 Represent different forms of black box testing [2]

Advantages:

1. Testers need not to have knowledge on specific programming language.
2. Testing is done from user's point of view.
3. It helps to expose any ambiguities or inconsistencies in the requirement specifications. [6]
4. Programmer and tester both are independent of each other.

Disadvantages:

1. Test cases are hard to design without clear terms.
2. Probability of having duplication of tests that are already done by programmer.
3. Various parts of back end are not tested at all.

B. White Box Testing

White box testing mainly focuses on internal logic and structure of the code. White-box is done when the programmer has techniques full knowledge on the program structure. With this technique it is possible to test every branch and decision in the program.[11][4]

1) *Control Flow Testing:* It is a structural testing strategy that uses the program control flow as a model control flow and favors more but simpler paths over fewer but complicated path.

2) *Basis Path Testing:* Basis path testing allows the test case designer to produce a logical complexity measure of procedural design and then uses this measure as an approach for outlining a basic set of execution paths.

3) *Data Flow testing:* In this type of testing the control flow graph is annotated with the information about how the program variables are define and used.

4) *Loop Testing:* It exclusively focuses on the validity of loop construct.

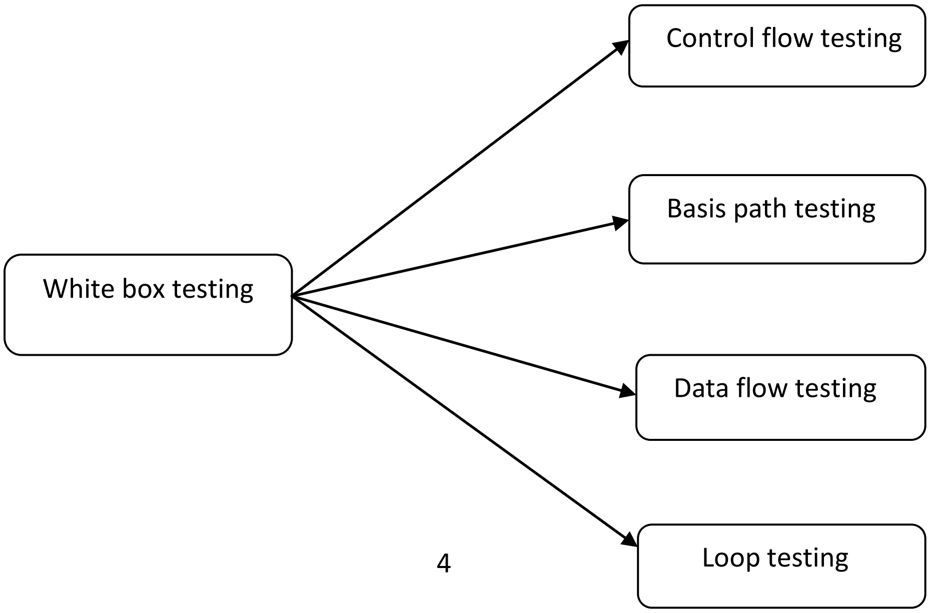

Fig.3 Represent different forms of white box testing [6]

Advantages:

1. It reveals error in hidden code by removing extra lines of code.
2. Maximum coverage is attained during test scenario writing. [7]
3. Developer carefully gives reasons about implementation.

Disadvantages:

1. A skilled tester is needed to carry out this testing because knowledge of internal structure is required.
2. Many paths will remain untested as it is very difficult to look into every nook and corner to find out hidden errors.

C. Hybrid Testing

Hybrid attempts, and generally succeeds, to combine the benefits of both black-box and white-box testing. Hybrid testing takes the straight-forward approach of black-box testing, but also employs some limited knowledge of the inner workings of the application.

White box + Black box = Hybrid, it is a technique to test the application with limited knowledge of the internal working of an application and also has the knowledge of fundamental aspects of the system. [7] Therefore, a tester can verify both the output of the user interface and also the process that leads to that user interface output. Hybrid testing can be applied to most testing phases; however it is mostly used in integration testing.

Advantages:

1. It provides combined benefit of black box and white box testing techniques.
2. In grey box testing, tester can design excellent test scenarios.
3. Unbiased testing
4. Create an intelligent test authoring.

Disadvantages:

1. Test coverage is limited as the access to source code is not available.
2. Many program paths remain untested.
3. The test cases can be redundant. [7]

TABLE 1

COMPARISON BETWEEN THREE FORMS OF TESTING TECHNIQUES

Sr.no.	Black box testing	Hybrid testing	White box testing
1.	The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledge of the internal workings of the application.

2.	Also known as closed-box testing, data-driven testing, or functional testing.	Also known as translucent testing, as the tester has limited knowledge of the insides of the application.	Also known as clear-box testing, structural testing, or code-based testing.
3.	Performed by end-users and also by testers and developers.	Performed by end-users and also by testers and developers.	Normally done by testers and developers.
4.	Testing is based on external expectations - Internal behavior of the application is unknown.	Testing is done on the basis of high-level database diagrams and data flow diagrams.	Internal workings are fully known and the tester can design test data accordingly.
5.	It is exhaustive and the least time-consuming.	Partly time-consuming and exhaustive.	The most exhaustive and time-consuming type of testing.
6.	Not suited for algorithm testing.	Not suited for algorithm testing.	Suited for algorithm testing.
7.	This can only be done by trial and error method.	Data domains and internal boundaries can be tested, if known.	Data domains and internal boundaries can be better tested.

IV. CONCLUSIONS

Software testing is the activity that executes software with an intention of finding errors in it. Software testing can provide an independent view of the software to allow the business to appreciate and understand the risk of software implementation. To carry out software testing in a more effective manner, this paper provides a review of three main techniques of software testing.

REFERENCES

[1] Software Testing. Gregory M. Kapfhammer. The Computer Science and Engineering Handbook, CRC Press. May, 2004.

[2] Mohd. Ehmer Khan, “Different Approaches to Black Box Testing Technique for Finding Errors,” IJSEA, Vol. 2, No. 4, pp 31-40, October 2011.

[3] Jovanovi?, I. "Software Testing Methods and Techniques. " The IPSI BgD Transactions on Internet Research: 30.

[4] F. Saglietti, N. Oster, and F. Pinte, “White and grey-box verification and validation approaches for safetyand security-critical software systems,” Information Security Technical Report, vol. 13, no. 1, pp. 10–16, 2008.

[5] T. Murnane and K. Reed, “On the effectiveness of mutation analysis as a black box testing technique,” in Software Engineering Conference, 2001. Proceedings. 2001 Australian, 2001, pp. 12 –20.

[6] Nidhra, Srinivas, and Jagruthi Dondeti. "Black box and white box testing techniques- A Literature." *International Journal of Embedded Systems & Applications* 2.2 (2012).

[7] Khan, Mohd Ehmer, and Farmeena Khan. "A Comparative Study of White Box, Black Box and Grey-box Testing Techniques." *International Journal of Advanced Computer Sciences and Applications* 3, no. 6 (2012): 12-15

[8] P. Mitra, S. Chatterjee, and N. Ali, “Graphical analysis of MC/DC using automated software testing,”in Electronics Computer Technology (ICECT), 2011 3rd International Conference on, 2011, vol. 3, pp. 145 –149.