

A STUDY OF WHITE BOX AND BLACK BOX SOFTWARE TESTING

¹Deepak Dhawan, ²Heena

^{1,2}Assistant professor

Government College

Sri Muktsar Sahib

Abstract- Software testing is the process of executing and evaluating the software with the aim of finding out the errors. It plays a significant role in the software development life cycle. Different techniques and methods are used for software testing like white box testing, black box testing and grey box testing. Considering the meaning and principles of software testing, the paper focuses on white box testing and black box testing. The paper further examines the various forms of white box testing and black box testing. Comparison between both the white box testing and black box testing is also studied and focused upon.

Index Terms – White box testing, Black box testing.

I. INTRODUCTION

Software testing is the process whose aim is to find errors, to evaluate the capabilities and attributes of the software unit and to check whether it satisfies the requirements of the users or not. In this process, system components and system requirements are evaluated and exercised manually or by automated tools to check whether specified requirements are satisfied and difference between expected and actual results are analyzed [1]. It can also be said that testing is executing or exercising a program with the aim of detecting the errors before delivery to the end user. In Software development life cycle, software testing phase is very important. It involves time and cost, but without testing it is impossible to deliver software product successfully, as during development process, mistakes can be done by the software developers and these mistakes can only be corrected by testing [2].

II. REVIEW OF LITERATURE

Coward [1988] had given a review on software testing by explaining the aims of software testing and described the static and dynamic analysis and structural and functional strategies of testing.

Khan [2011] presented various approaches of black box testing and white box testing techniques for detecting the presence of errors in the software. In his work he has also explained the various forms of white box testing techniques such as data flow testing, basis path testing, loop testing and branch testing and various forms of black box testing techniques such as equivalence partitioning, boundary value analysis, cause and effect graph and all pair testing.

Hussain et. al [2015] represented the outline of software testing and emphasized the role of testing in the reliability of software. They performed the comparative analysis of white box, black box and grey box testing technique and concluded that white box techniques gives better results in terms of software reliability.

Jampani et. al [2016] presented the comparative analysis of various software testing techniques in order to find their effectiveness for the generation of test cases and for enhancement of software quality. In their work they compared three testing techniques namely black box, white box and grey box.

Nouman et.al [2016] represented the survey and tutorial of techniques that are widely used for testing programs during the software development life cycle. They have presented the summary of existing methods of testing and performed the analysis of various commercially available automatic testing tools for white box and black box testing techniques.

Gaur et. al [2016] described the meaning, principles, methodologies and importance of software testing and gave the importance of emerging technique of genetic algorithm for software testing. They also gave comparative study of various software testing techniques.

Er. Suruchi [2016] presented the analysis and comparison of various software testing techniques to find out the best one and tried to relate testing techniques for measuring the attributes of software quality.

Grover [2016] explained the levels and methods of software testing and performed the comparative analysis of important testing levels and methods.

III. PRINCIPLES OF SOFTWARE TESTING

The various principles of software testing are [3, 4]

1. Exhaustive testing is impossible-It is not possible to exercise all combinations of paths during testing. So priorities and risks are used to focus on more suspected components instead of less suspected components.

2. An independent third party should conduct testing- In order to have effective testing, it must not be conducted by the software engineer who developed the software product, rather it must be done by an independent third party.

3. Traceable- All the tests must be traceable to the requirements of the customers, as most of the critical defects occurs if programs fails to satisfy the customer's requirements.

4. Planning must be done long before the beginning of testing- After the completion of the requirement model planning of tests can be started and as soon as design phase is solidified test case design can be extended.

5. Testing should start from small and progress towards large-Initially tests must lay emphasis on individual units then focus should shift to integrated components then finally to the whole system.

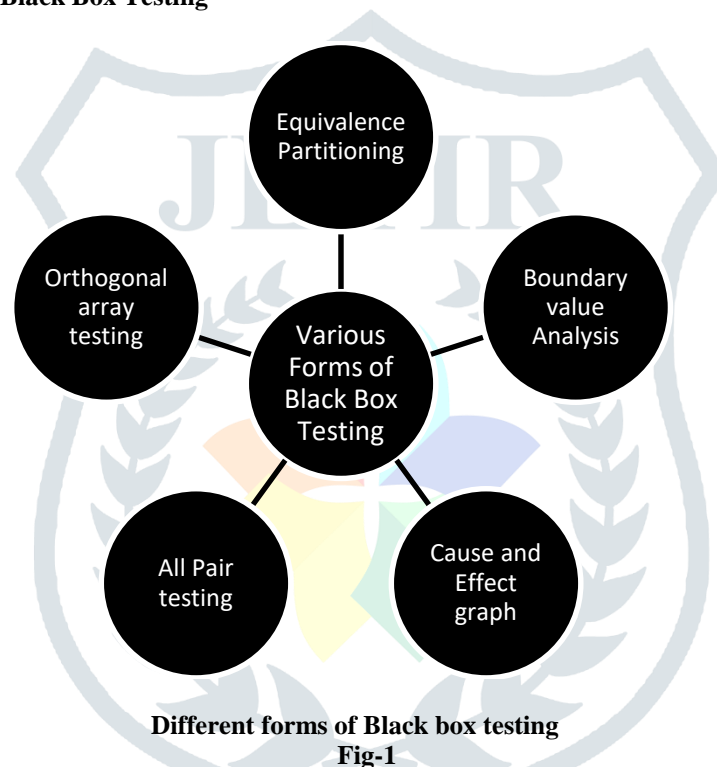
IV. TECHNIQUES OF SOFTWARE TESTING

The various software testing techniques are

1. Black Box Testing

It is a testing technique in which main focus is on the functional requirements of the software. The tester is only aware about the input and required output and the knowledge of internal working of the system is not required by him. The main basis for deciding the test cases are requirements and specifications of the module or program. The system is treated as a black box so explicit usage of knowledge of internal code or structure is not used. So it can be said that black box testing lay emphasis on functional requirements of the software. The test cases are designed to check for the errors which are related to missing or incorrect functions, data structures, interfaces, initialization and termination errors [3, 5].

1.1 Various techniques of Black Box Testing



The different approaches of Black box testing are [3, 6]

1.1.1 Equivalence partitioning- In this technique, the input domain of the software unit is divided into classes or partitions and test cases are derived. Thus it reduces the total number of test cases into finite set. For the input condition an equivalent class defines set of valid and invalid states.

1.1.2 Boundary value analysis-This technique tests the errors at the boundaries of the input domain. The various boundary values can be minimum, maximum, just inside and outside boundaries It is a test case designing technique that complements equivalence class partitioning. Instead of just focusing on input conditions, this approach derives the test cases from the output domain also.

1.1.3 Cause and effect Graph- In this method a graph is created and relationship between cause and effect is established. The input condition that leads to internal change in the system represents cause and the output condition or transformation represents the effect.

1.1.4 All pair Testing- In this technique test cases are designed for exercising all the possible combinations of every pair of input parameters.

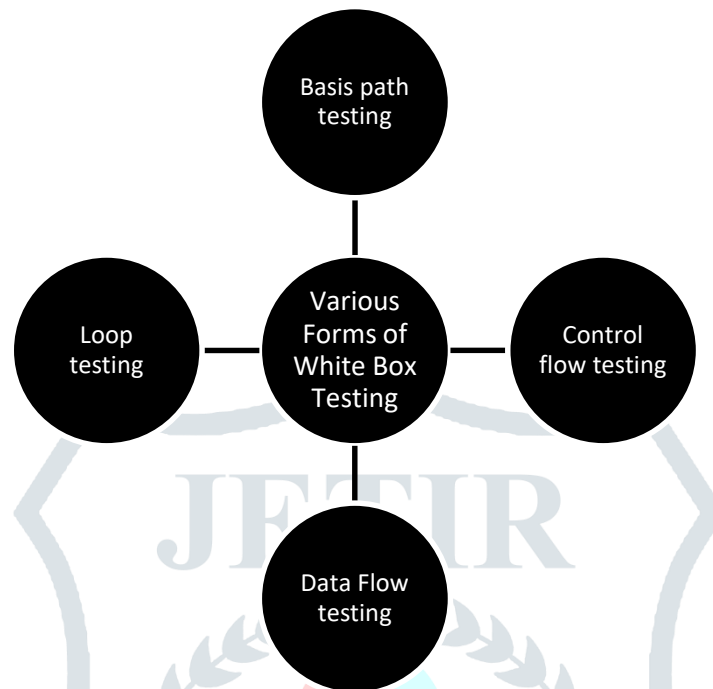
1.1.5 Orthogonal Array Testing- It is a technique that is applicable for those problems whose input domain is comparatively small but not too large to accommodate exhaustive testing.

2. White box Testing

It is a testing technique that focuses on the structure and internal working of the software. Soin this form of testing the tester is well aware about the internal structure and design of the component being tested. It is a test case design method in which

control structure of procedural design is used to derive the test cases. Using this technique the test cases designed ensure that within a module, all the independent paths have been exercised atleast once [3].

2.1 Various techniques of White Box Testing



Different forms of White box testing
Fig-2

The different approaches of white box testing are [3, 7]

2.1.1 Basis Path Testing

This method is based on a logical complexity measure of procedural design. This measure derived by test case designer acts as a guide to define basis set of execution paths.

i. Flow graph Notation- Flow graph notation is a way to represent flow of control in the program and helps to derive the basis set. The nodes in the flow graph represents procedural statements and edges represents control flow

ii. Cyclomatic Complexity- It represents a metric that helps to quantitatively measure the logical complexity of the program.

This metric tells about the number of independent paths in the basis set and defines the upper bound for the number of tests that must be performed. The three ways to compute cyclomatic complexity which is represented as $V(G)$ are

- i. $V(G)=P+1$ where P is number of predicate nodes in the flow graph.
- ii. $V(G)=E-N+2$ where E and N represents number of edges and number of nodes respectively of graph G.
- iii. $V(G)=$ Number of regions in the graph.

2.1.2. Control Flow Testing

This technique uses structure testing strategy that involves control flow of the program. The metric used to measure adequacy of test cases is called coverage. The different methods of coverage are statement coverage, condition coverage and branch coverage.

2.1.3 Data Flow Testing

This technique of testing focuses on data movement within the program. In this approach, the test paths of the program are selected on the basis of the locations where variables are defined and used. It helps to uncover errors such as un-initialized or undeclared variables used somewhere in the program.

2.1.4 Loop Testing

This test checks the validity of the loops which can be of different types such as simple loops, nested loops, unstructured loops and concatenated loops.

V. COMPARISON

White box testing and black box testing techniques are complementary to each other as one is not the alternative of the other

because the errors uncovered by both the approaches are different. Table 1 shows the differences between these two testing approaches [8-11].

Table 1. Comparison of Black Box and White Box testing technique

Sr No	Black Box Testing	White Box Testing
1.	It is testing technique in which tester does not have knowledge of the internal design and structure of the item to be tested	It is testing technique in which tester have knowledge of the internal design and structure of the item to be tested
2.	It is less time consuming task.	It is more time consuming task.
3.	It is not applicable for testing of algorithms	It is applicable for testing of algorithms
4.	In order to carry out black box testing programming knowledge is not required.	In order to carry out white box testing programming knowledge is required.
5.	It applies to higher levels of testing such as System testing and acceptance testing.	It applies to lower levels of testing such as Unit testing.
6.	Requirement specifications forms the basis for the test case design.	Detailed design documents forms the basis for the test case design.
7.	The main objective of this testing is to know the functionality that is provided by the system.	The main objective of this testing is to know that how system is performing.
8.	It is a kind of external or functional testing.	It is a kind of internal or structural testing.

VI. CONCLUSION

Software testing can be viewed as a process of examining and analyzing the software unit in order to find errors and to check the differences between expected and actual behavior. Various testing approaches and techniques have been proposed in the literature. Both the white box and black box software techniques have their own merits and demerits and are complementary approaches to each other. It cannot be said that one is alternative for another as both forms of testing techniques uncover different forms of errors.

REFERENCES

- [1] Hooda, I. and Rajinder, C.S. (2015), "Software Testing Process, Testing Types and Techniques", *International Journal of Computer Applications*, Vol. 111 No 13, Feb, pp. 10-14.
- [2] Clarence, J.M.; Ganesan, N; Ghosh,A. and ,Ghosh, N. (2012), "Testing OO Software Systems: A Survey of Steps and Challenges" *International Journal of Computer Applications*, Vol. 42, No. 8, March, pp. 16-20.
- [3] Roger Pressman,"Software Engineering, A Practitioner's Approach", Fifth Edition, ISBN-0-07-365578-3.
- [4] Quadri S.M.K. and Farooq, S.U. (2010), "Software Testing-Goals Principles and Limitations",*International Journal of Computer Applications*, Vol. 6, No. 9, September, pp. 7-10.
- [5] Pankaj Jalote (2001), "An Integrated Approach to Software Engineering", Second Edition, ISBN 81-7319-271-5.
- [6] Khan, M. E. (2011), "Different Approaches to Black Box Testing Technique for Finding Errors", *International Journal of Software Engineering and Applications*, Vol. 2, No. 4, October, pp.31-40.
- [7] Khan M. E. (2011), "Different Approaches to White Box Testing Technique for Finding Errors", *International Journal of Software Engineering and Its Applications*,Vol. 5, No. 3 July, pp. 1-10.
- [8] Hussain T. and Dr. Singh S. (2015), " A Comparative Study of Software Testing Techniques viz White Box and Black Box and Grey Box Testing" *International Journal of Allied Practice, Research and Review*, Vol. 2, No. 5, August, pp.1-8.

- [9] Gaur J.; Goyal A.; Choudary T. and Sabitha S. (2016), "A Walk Through of Software Testing Techniques", *International Conference on System Modeling and Advancement in Research Trends*, November, pp-103-08.
- [10] Er. Suruchi (2016), "Comparative Analysis of Software Testing Techniques: A review", *International journal of Advanced Science and Research*, May, ISSN:2455-4227, pp.46-50.
- [11] Jampani R.; Talasu N. and Dr. Manjula, R. (2016), "Survey of Software Testing Techniques", *International Journal of Research in Applied Science and Engineering Technology*, Vol. 4, No. 4, April, pp. 924-29.
- [12] Nouman M.; Pervez, U. and Hasan O.(2016), "Software Testing A Survey and Tutorial on White Box and Black Box Testing of C/C++Programs",*2016 IEEE Region 10 Symposium Bali*, Indonesia, pp 225-30.
- [13] Khan J.; Khan R.U.; Khan F. and Raza S. (2013), "A Survey of Software Testing: Technique Comparison and Analysis", *International journal of Computer Science and Telecommunications*, Vol. 4, No. 7, July, pp. 29-34.
- [14] Coward P. D. (1988), "A review of software testing", *Information and Software Technology - The Software Life Cycle*, Vol. 30, No. 3, April, pp. 189-98.
- [15] Grover, D. K. (2016), "Comparison Study of Software Testing Methods and Levels-A Review", *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 4, No. 8, August, pp.15011-16.

