



WWJMRD 2018; 4(2): 384-387

www.wwjmr.com

International Journal

Peer Reviewed Journal

Refereed Journal

Indexed Journal

UGC Approved Journal

Impact Factor MJIF: 4.25

E-ISSN: 2454-6615

Paramjeet Kaur

MCA, UGC-NET

Assistant Professor of

Computer Science

Guru Nanak College

Ferozepur Cantt, Punjab,

India

A Research Paper on White Box Testing

Paramjeet Kaur

Abstract

To ensure high quality software, it is required to test Software. Software Testing is a process of finding errors while executing a program so that we get a zero defect software. It is aimed at evaluating the capability of a program. Software testing is the set of activities conducted with the intent of finding errors in software. It also verifies and validate whether the program is working correctly with no bugs. It is used to determine the correctness, effectiveness of software and product to be manufactured. Two main important testing techniques are White box testing and black box testing. In my paper I have described one of the most important technique that is white box testing in which all codes and internal implementation details clearly revealed to the testers.

Keywords: Introduction, white box testing, Coverage, Software testing strategies, advantages, disadvantages

Introduction

Software Testing is not an easy task it is time consuming activity and even when our testing complete it is also very difficult to determine. A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed In white box testing first step is to determine source code, structures and all other tools which is associated with the development of the software. Second testers must have to aware different tools and techniques of white box Testing. It works mainly to improve Security, flow of input and output, and works to improve design and usability.

What Is White Box Testing?

In this testing, internal details and structure of system is made visible. Thus, it is highly efficient in detecting and resolving problems, because bugs can often be found before they cause trouble. We can thus define this method as testing software with the knowledge of its internal structure and coding. White box testing is also called clear box testing, white box analysis or clear box analysis. It is a strategy for finding errors in which the tester has complete knowledge of how the program components interact. In this method of the testing the test cases are calculated based on analysis of internal structure of the system based on code coverage, branch coverage, paths coverage, condition coverage etc. It involves several features such as

1. Tester has full knowledge of the internal working of the software.
2. Data domains and internal boundaries can be easily tested.
3. Best suited for Algorithm testing
4. Mostly done by testers and developers.
5. Granularity is high.

Correspondence:

Paramjeet Kaur

MCA, UGC-NET

Assistant Professor of

Computer Science

Guru Nanak College

Ferozepur Cantt, Punjab,

India

Various Names of white Box Testing

- Clear box Testing
- Open box Testing
- Glass box Testing
- Transparent Box testing
- Code based Testing
- Structural Testing
- Logic driven Testing

- Exhaustive testing is impossible
- Early testing must be done
- Defect clustering
- Pesticide paradox
- Testing is context-dependent
- Absence of error is a fallacy

White Box Testing Techniques

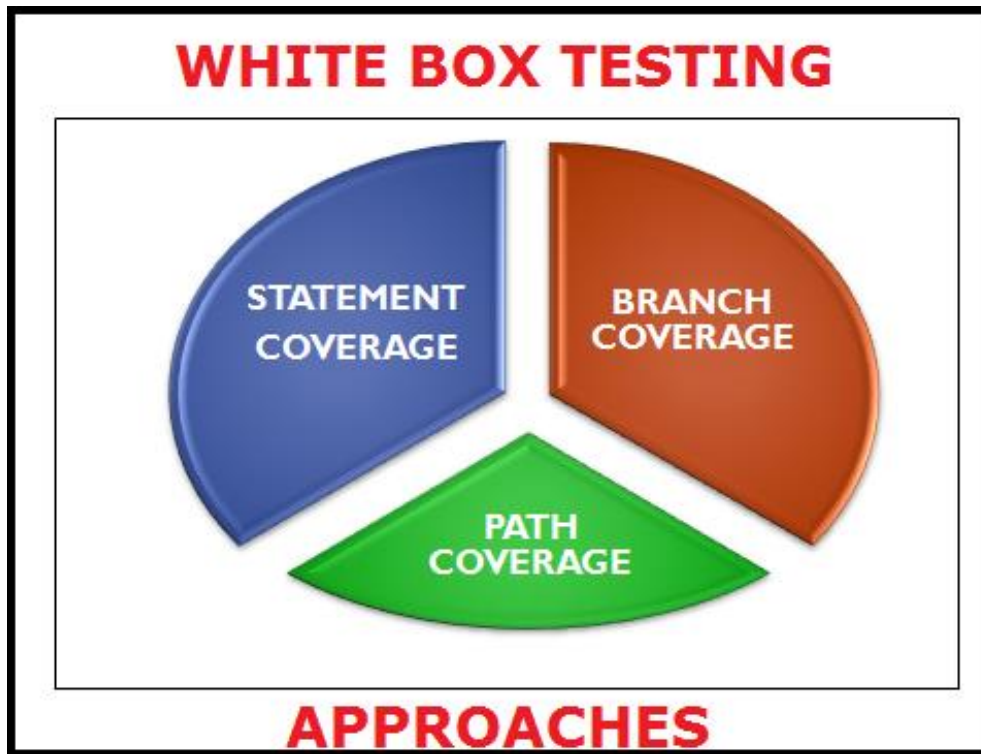
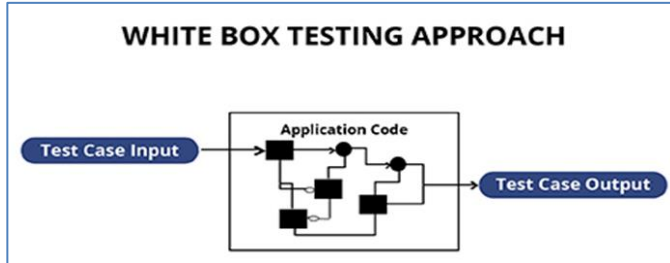
There are many techniques available in white box testing because the problem of intractability is eased by specific knowledge and attention on the structure of the software under test. The intension of exhausting some aspect of the software is still strong in white box testing and some degree of exhaustion can be achieved, such as exhausting each line of code at least once, traverse every branch statements or cover all the possible combination of true and false condition predicates. The various white-box testing techniques are listed below.

- Statement coverage
- Branch coverage
- Path coverage

Testing principles

There are seven major testing principles which must be considered while testing a product.

- Testing shows the presence of defects



Branch Coverage

Branch coverage is also known as Decision coverage or all-edges coverage. Branch coverage is a testing method, which aims to ensure that each one of the possible branch from each decision point is executed at least once and thereby ensuring that all reachable code is executed. That is, every branch taken each way, true and false. It helps in validating all the branches in the code making sure that no branch leads to abnormal behavior of the application.

Formula:

$$\text{Branch Testing} = (\text{Number of decisions outcomes tested} / \text{Total Number of decision Outcomes}) \times 100 \%$$

Statement Coverage

- The statement coverage is also known as line coverage or segment coverage.
- The statement coverage covers only the true conditions.
- The statement coverage is also known as line coverage or segment coverage.
- The statement coverage covers only the true conditions.

Statement coverage is a white box test design technique which involves execution of all the executable statements in the source code at least once. It is used to calculate and measure the number of statements in the source code which

can be executed given the requirements. The statement coverage can be calculated as shown below

$$\text{Statement Coverage} = \frac{\text{Number of executed statements}}{\text{Total number of statements}} \times 100$$

Path coverage

In this the test case is executed in such a way that every path is executed at least once. All possible control paths taken, including all loop paths taken zero, once, and multiple items in path coverage technique, the test cases are prepared based on the logical complexity measure of a procedural design. In this type of testing every statement in the program is guaranteed to be executed at least one time. Path testing is a structural testing method that involves using the source code of a program to attempt to find every possible executable path. The idea is that we are then able to test each individual path in as many as possible in order to maximize the coverage of each test case. This gives the best possible chance of discovering all faults within a piece of code. The ability to use the code for testing means that there exists a basis on which test case can be rigorously defined. This allows for both the test cases and their results to be analyzed mathematically, resulting in more precise measurement. If there are say N number of decisions in a method, then it could have 2^N number of paths. Various strategies have been developed for identifying useful subsets of paths for testing when Path Coverage is impractical:

- Loop Coverage
- Basis Path Coverage
- Dataflow Coverage

Basis path testing

Basis path testing means for ensuring that all independent paths through a code module have been tested. An independent path is any path through the code that introduces at least one new set of processing statement. Basis path testing is a structured testing technique used for designing test cases. It also examines all possible paths at least once. Creating and executing tests for all possible paths results in 100% statement coverage and branch coverage. This type of testing technique used to find highest numbers of errors and bugs.

Data Flow Testing

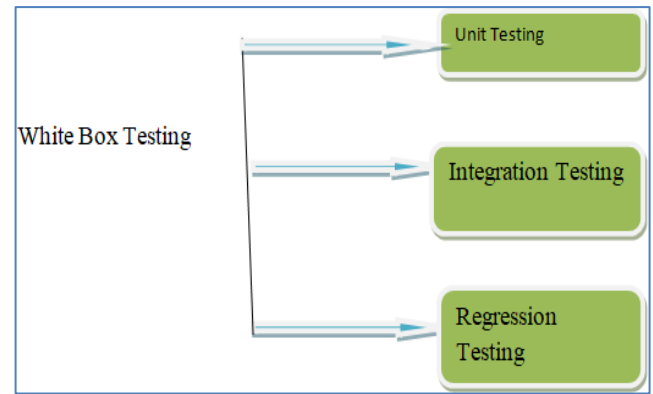
It uses the control flow graph to explore errors and bugs in the data. Data flow testing used to detect improper use of data in program. In data flow testing for any testing we used data flow graph which represents data dependencies between a numbers of operations.

Loop Testing

Loop testing is a white box testing approach that concentrates on the validity of loop constructs. There are four loops can be defined:

- Simple Loop
- Nested Loop
- Concatenated Loop
- Unstructured Loop

White Box Testing Strategies



Unit Testing

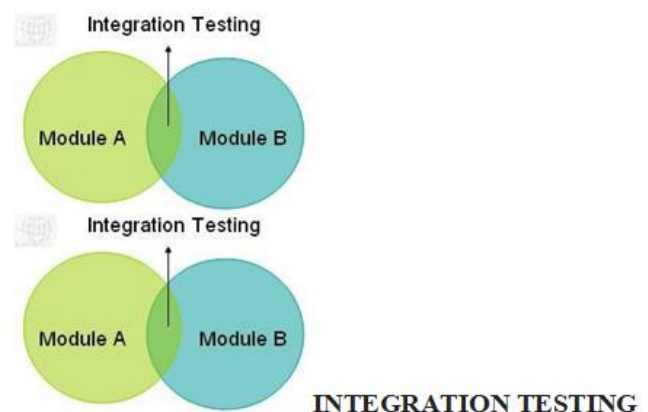
It is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of software. It usually has one or a few inputs and usually a single output. It is usually performed by the testers and main aim of unit testing is to show that test each parts separately and show that each parts are correct in its functionality.

Integration Testing

It is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. The main aim of this testing is to show faults and errors in the integrated components. Integration testing follows two approaches

- Top-down approach
- Bottom-up approach

This type of testing is done by integration tester or test team.



Regression Testing

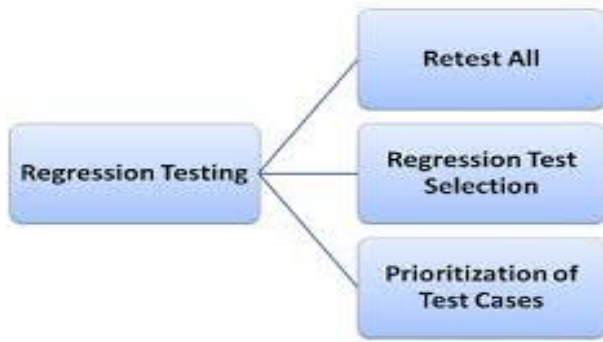
It is a type of software testing that ensures that previously developed and tested software still performs the same way after it is changed or interfaced with other software. Changes may include software enhancements, patches, configuration changes, etc. It ensures that old code still works once the new code changes are done. This testing is

done to make sure that new code changes should not have side effects on the existing functionalities. Regression testing is required

- When there is a change in requirements and code is modified according to the requirement.
- When new feature is added to the software.
- Defect fixing.
- Performance issues fixing

➤ **Types of Regression testing**

- Final Regression Testing
- Regression testing



Advantages of White Box Testing Are

Increased Effectiveness:-Cross checking design decisions and assumptions against source code may outline a robust design, but the implementation may align with the design intent.

Full Code Pathway Capable: - all the possible code pathways can be tested including error handling, resource dependencies and additional internal code logic.

Early defect identification: - Analyzing source code and developing tests based on the implementation details enables testers to find programming errors quickly.

Reveal Hidden Code Flaws:-It helps in removing the extra lines of code, which can bring in hidden defects.

Introspection: - Introspection, or the ability to look inside the application, means that testers can identify objects programmatically.

Thoroughness: - In situation where it is essential to know that every path has been thoroughly tested, that every possible internal interaction has been examined, white box testing is the only viable method. As such, white box testing offers testers the ability to be more thorough in terms of how much of an application they can test.

Disadvantages of White Box Testing

Difficult to scale:-requires intimate knowledge of target system, testing tools and coding languages.

Difficult to maintain: - requires specialized tools such as source code analyzers, debuggers and fault injectors.

Highly intrusive: - requires code modification has been done using interactive debuggers, or by actually changing the source code. This may be adequate for small programs; however, it does not scale well to larger applications.

Expensive: - White box testing very expensive type of testing.

Time consuming: - This type of testing also very time consuming testing.

Conclusion

This paper defines a complete set of white box testing This paper will help developers: 1. Study Internal data structure and how value is allocated to a variable 2. To make logical decisions based 3. Tests every independent path of the code 4.Execute loops at the boundaries. We must emphasis on testing while developing a software as it can result harmful to man and resources.

References

1. [Eckel] B. Eckel, Thinking in Java, Third Edition, ISBN 0-13-100287-2, Prentice Hall, 2003
2. [Frankel] D. Frankel, Model Driven Architecture, ISBN 0-471-31920-1, Wiley, 2003
3. "A path-oriented automatic random testing based on double constraint propagation" Ruilian Zhao, Yuandong Huang International Journal of Software Engineering & Applications (IJSEA), Vol.3, No.2, March 2012
4. black box and white box testing techniques –a literature review, Srinivas Nidhra and Jagruthi Dondeti, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012
5. White Box Coverage and Control Flow Graphs Venezia Elodie, 2011
6. Paul Ammann, Jeff Offutt, "Introduction to Software Testing", Cambridge University Press, 2013
7. M. Harman, P. McMinn, J. Souza, and S. Yoo. Search based software engineering: Techniques, taxonomy, tutorial. In Empirical Software Engineering and Verification, pages 1{59. 2012.
8. C. Henard, M. Papadakis, G. Perrouin, J. Klein, P. Heymans, and Y. Le Traon. Bypassing the combinatorial explosion: Using similarity to generate and prioritize t-wise test configurations for software product lines. IEEE Trans. Softw. Eng., 40(7):650{670, July 2014.
9. www.wikipedia.com
10. "Fundamental of Software Engineering " Rajib Mall