ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

# Different Approaches to White Box Testing to Find Bug

**Nidhi Gupta**
M.Tech, CSE

**Abstract**
*White box testing is a mechanism that involves multiple processes and designs that verifies whether the code works as expected. It provides the internal structure of a system or a component (thus known as Structural Testing) and provides full visibility of code and logic of software product (also known as Clear Box and Glass Box Testing). It works mainly to improve Security, flow of input and output, and works to improve design and usability. This paper will help software testers to understand white box testing for security, different approaches to white box testing techniques, relevant tools and techniques and expected results.*

**Keywords**
*Basis Path Testing, Branch Testing, Control Flow Testing, Data Flow Testing, Loop Testing, White Box Testing, test case etc.*

## I. Introduction

Software testing is used to verify and validate the quality of software [27] by executing a program to find bugs. Software testing can be categorised as Black Box Testing and White Box Testing. It is done to ensure that the software being developed is error free, cost effective and according to customer requirement. To perform testing we have white box testing and black box testing methods. Our focus is on white box testing which validates designs, decision, and to uncover vulnerabilities to provide secure software. White box Testing is often used for Verification (i.e. are we building the product right). It is based on the analysis of internal working and structure of the software. First step is to analyse design documentation, source code and other artefacts associated with the development of the software. Second, tester must be aware of different tools and techniques for white box testing [2]. 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

### A. Applicable for 3 testing's

White Box Testing is used for the following 3 types of Testing:

1. Unit testing, tests individual software or hardware groups of related units (IEEE, 1990). A unit of software component that cannot be further sub divided (IEEE, 1990). Test cases are written to ensure that the unit is coded correctly before integrating it with other code. It is proven that approximately 65% of the bugs can be estimated in Unit Testing only.
2. Integrating Testing, tests software component, hardware component or both when combined to evaluate the interaction between them. Test cases are written to examine interfaces between various units. It can be done with Black box test cases also.
3. Regression Testing, selective retesting of a system or component to verify that the modifications have not caused any unintended errors. It can be done with Black box test cases, white box test cases or the combination of two.Using white box testing techniques a developer can design test cases.

### 1. Test Cases

A test case involves an input, an action and an expected result. Purpose of using it is to uncover errors. Serious defects that can crash the system are found before the application is released to the end user. White box testing tests cases are primarily concerned with the broadest coverage of the source code. Hence, the tester must be fully aware of the logic of the application and must develop test cases that could execute maximum of code.

Written specifications and user documents can provide excellent information for making test cases. Test cases are written based on the function and flow of application. Grouping these test forms a test procedure in white box testing. Test cases involve fields such as Test Case Id, Assumptions, Test Data, Expected Results, Actual Results, Pass/Fail, and Comment and so on. Good test cases in white box testing do not try to find all the bugs, but rather focus on what is good for the application which maintains quality assurance of an organisation with respect to time and money.

A software engineer can design test cases [3] that can:

1. Exercise internal data structure to ensure their validity.
2. Exercise logical decisions on both their true and false side.
3. Exercise independent path within a module.
4. Execute loops at their boundaries and within their operational bounds.

## II. Literature Work

Mohd. Ehmer khan concluded that white box testing is a verification and validation technique which software engineer can use to examine their code. Tester need to understand the white box techniques that are available to make educated decisions about their use for the specific system we are currently, and in future, will be testing. Properly planned with explicit input output combinations white box testing is a controlled V & V technique.

Vinod Dandoti summarised that there are alternatives to white box testing. Performing only the white box testing is not cost effective even though it increases the test effectiveness. It is good practice to perform White box testing for high-risk areas, and black box testing for the whole system. More tests can be developed and executed by complementing both.

## III. Testing Techniques

White Box Testing involves code, code structure, internal design and how they are coded. Techniques that are used are:

1. Data Flow Testing
2. Control Flow/Coverage Testing

3. Basis PathTesting
4. Loop Testing

### A. Data Flow Testing

Data Flow Testing specifies how data revolves around the program and how the value is allocated to a variable. It not only explores program control flows but also shows how a variable is defined and used. It aims to execute the sub paths from points where each variable is defined to points where it is referenced. These sub paths are termed as Definition-use pairs or du-pairs.

Most failures execute the incorrect definition such as:
1. Incorrect assignment or input statement
2. Incorrect path is taken, which leads to incorrect definition (predicate is faulty)
3. Definition is missing (i.e. null definition)

Data flow testing involves the following steps to correct them:
1. Number the line
2. List the variables
3. List occurrences and assign a category to each variable
4. Identify du variable and their use
5. Define test cases depending upon the required coverage.

### B. Control Flow/ Coverage Testing

It uses the program control flow as a model control flow testing which is effective relatively in small programs or segments of larger programs. But it has certain limitations:
- It cannot catch all initialization mistakes
- Specification mistakes are not caught by this testing
- It becomes difficult for some other person to find missing path and features, other than the developer.

#### 1. Statement coverage

It measures the percentage of statements that have been executed by test cases. Statement coverage less than 100% means not all lines of code have been executed. Goal of white box testing is to execute each and every path in the program but it becomes unfeasible if the path contains a loop.

#### 2. Branch coverage

Branch coverage means stronger logic coverage which is performed usually to satisfy statement coverage. Here multiple test cases are written which results either true or false at least once so that each branch is traversed. For example If Statement and Go To Statements.

#### However, there are 3 exceptions:

1. In pathological situation, program has no decision.
2. When program has multiple entry points
3. Statements within ON-units, as traversing every branch will not execute all ON-units.

#### 3. Condition Coverage

A criterion which is stronger than decision coverage is condition coverage. Every condition in a decision of the program has taken all possible outcomes at least once and every decision in the program has taken all possible outcomes at least once. Test cases are written such that each condition in a decision takes on all possible outcomes at least once. Condition coverage only checks what is inside an If condition. If the value checked is evaluated outside the If condition, then compiler may terminate the evaluation of expression.

#### 4. Function Coverage

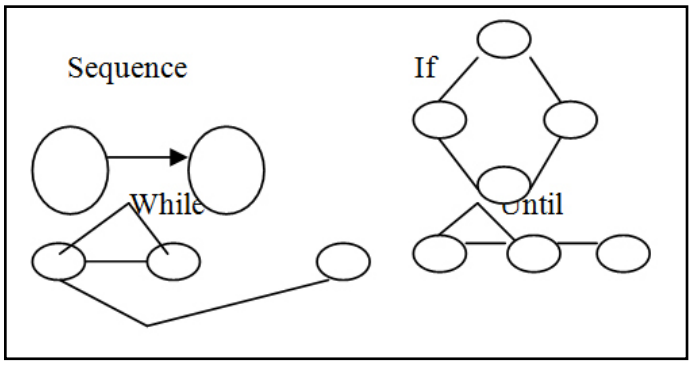
Programs are coded by calling a set of functions. Each function is a smallest unit that does a specific functionality. It specifies much functionality of the design has been exercised by the verification environment. Tests are written to exercise each of the different functions in the code, thus it is proven that if each test is completed properly, then entire set of functionality is verified. However, this coverage has 2 limitations.
- There is not a defined list of 100% functionality of the design is and therefore, there may be a missing functionality in the list.
- There is no real way to check that the coverage model is correct, manual check is the only way.

### C. Basis Path Testing

Basis Path proposed by Tom Mc Cabe ensures that all independent path of the code have been tested. An independent path is any path through the code that introduces at least one new set of processing statements or a new condition. Basis path testing provides a minimum, lower bound on the number of test cases that needs to be written. Test cases that exercise basic set will execute every statement at least once. Basic path testing makes sure that each independent path through the code is taken in a predetermined order.

#### 1. Flow Graph Notation

Notation used for representing control flow is Control flow graph. A control flow graph is a directed graph that consists of node and control flow. Node is labelled by a circle where as control flow represented by an arc with an arrow or line.
On a flow graph:
1. The symbol arrows called as edges that represents the flow of control
2. Circles are called nodes, which represents one or more actions
3. Area bounded by edges and nodes are called regions
4. A predicate node is a node containing a condition.




Procedural designs can be translated into flow graph:

#### 2. Cyclomatic Complexity

It is software metric which provides logical complexity of a program by measuring the number of paths through a method. Every method, at a minimum has a cyclomatic complexity of 1, since there is at least 1 path through a method which means getter/ setter method has CCN=1.
public String getName()

```
{
return this.name;
}
```

Cyclomatic complexity number (CCN) can be calculated as CCN= E-N+P, Where E represents the number of edges on the graph, N the number of nodes on the graph, and P the number of connected components

Drawbacks

1. It does not measure data complexity of a program
2. It must be used with care as it gives misleading figures of simple comparisons and decision structures.

### 3. Deriving Test cases

Procedure for deriving test cases:

1. Draw a flow graph using design or code as foundation
2. Determine cyclomatic complexity of the resultant flow graph.
3. Determine set of linearly independent paths
4. Prepare test cases that will execute each path

Each test case is executed and compared to the expected result. Development of test cases makes it sure to the tester that all the statement in the program have been executed at least once.

### 4. Loop Testing

Statement and branch Coverage are not sufficient. Thus loop testing is performed which focuses on the validity of loop construct. Loops are simple to test unless dependencies exist.

#### Single Loop Strategy

- Zero iterations
- One iteration
- Two iteration
- Typical number of iterations
- n-1, n, and n+1 iterations

#### Nested Loop Strategy

- Single loop strategy
- Select minimum values for outer loop(s)
- Treat inner loop as a single loop
- Work outwards and choose typical values for inner loops

#### Concatenated Loops

- Treat as single, if independent
- Treat as nested, if dependent

## IV. Advantages Of White Box Over Black Box Testing

White box testing has multiple advantages in order to make secure software:

1. It removes extra lines of code that can bring in hidden defects
2. Optimizes the code
3. As the internal code is known to the tester, it becomes easy to find out which type of data can help in testing application efficiently.
4. Bugs are identified in hidden code.
5. Testing can be done at an earlier stage which covers most possible paths to test.

## V. Conclusions and Future Work

In the present paper, we have discussed white box testing which is a controlled V & V technique. We have reviewed various White Box testing techniques such as Data Flow Testing, Control Flow/ Coverage Testing, Basis Path Testing, and Loop Testing with test cases. Each testing techniques will allow testers to design test cases that will help developers to properly plan specific decisions related to the systems. 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

Next, I have compared advantages of White box testing over Black box testing technique.

## References


[1] *Ammann, Paul; Offutt, Jeff, "Introduction to Software Testing", Cambridge University Press, 2008.*

[2] *Beizer B, "Software Testing Technique", International Thomson Computer Press, 1990.*

[3] *Black, Rex, "Advanced Software Testing"- Vol. 2: Guide to the ISTQB Advanced Certification as an Advanced Test Manager. Santa Barbara: Rocky Nook Publisher., December 2008.*

[4] *Binder, Robert V," Testing Object-Oriented Systems: Models, Patterns, and Tools. Addison Wesley 1999.*

[5] *Böhm, Jacopini. "Flow diagrams, turing machines and languages with only two formation rules" Comm. ACM, 9(5):366-371, May 1966.*

[6] *Cem Kaner," Exploratory Testing", Florida Institute of Technology, November 2006.*

[7] *Dorf, Richard C.: Computers, "Software Engineering, and Digital Devices, Chapter 12, pg. 15. CRC Press, 2006.*

[8] *Ehmer Khan, Mohd (May 2010). "Different Forms of Software Testing Techniques for Finding Errors". IJCSI International Journal of Computer Science Issues 7(3): 12. Retrieved 12 February 2013.*

[9] *Falk, Jack. Testing,"Computer Software, 2nd Edition. Wiley Computer Publishing.*

[10] *Glenford J. Myers," The Art of Software Testing, 2nd edition, 2004.*

[11] *Golang.org. "testing - The Go Programming Language". Retrieved 3 December 2013.*

[12] *Girish Janardhanudu nd Ken van Wyk, "White Box Testing", The Build Security In, September 26, 2005*

[13] *IEEE Standards Board, "IEEE Standard for Software Unit Testing: An American National Standard, ANSI/IEEE Std 1008-1987"*

[14] *Kaner, Cem; Falk, Jack and Nguyen, Hung Quoc,"Testing Computer Software", 2nd Ed. New York,1999.*

[15] *Kolawa, Adam, Huizinga, Dorota," Automated Defect Prevention: Best Practices in Software Management", Wiley-IEEE Computer Society Press, 2007.*

[16] *Khan Farmeena,"A Comparative Study of White Box, Black Box and Grey Box Testing Techniques", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012.*

[17] *Liu Chang ," Teaching "Data Flow Testing" in an Software Engineering Course Teaching "Data Flow Testing" in*

*an Software Engineering Course", School of Electrical Engineering and Computer Science, Ohio University Athens, Ohio 45701, USA.*

*[18] Leitner, A., Ciupa, I., Oriol, M., Meyer, B., Fiva, A., "Contract Driven Development = Test Driven Development – Writing Test Cases", Proceedings of ESEC/FSE'07: European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, September 2007.*

*[19] Mohd. Ehmer Khan,"Different Approaches to White Box Testing Technique for Finding Errors" International Journal of Software Engineering and Its Applications Vol. 5 No. 3, July, 2011.*

*[20] McCabe, Watson "Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric",1996.*

*[21] McCabe "A Complexity Measure". IEEE Transactions on Software Engineering: 308–320, December 1976.*

*[22] Paul Ammann, Jeff Offutt," Introduction to Software Testing", Cambridge University Press, 2013 .*

*[23] Pan, Jiantao "Software Testing (18-849b Dependable Embedded Systems)". Topics in Dependable Embedded Systems. Electrical and Computer Engineering Department, Carnegie Mellon University, 1998.*

*[24] Ramdeo, Anand (5 May 2011). "Gray Box Testing - Software". Testing Geek. Retrieved 19 January 2012*

*[25] ROBERT GOLD," CONTROL FLOWGRAPHS AND CODE COVERAGE", Ingolstadt University of Applied Sciences, Int. J. Appl. Math. Comput. Sci., 2010, Vol. 20, No. 4, 739–749.*

*[26] Robert V. Binder,"Testing Object-Oriented Systems", Models, Patterns, and Tools.*

*[27] Srinivas Nidhra and Jagruthi Dondeti," BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW", International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012 DOI.*