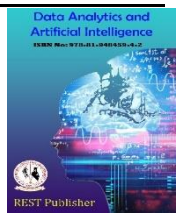
DOI: https://doi.org/10.46632/daai/2/6/8

Data Analytics and Artificial Intelligence

Vol: 2(6), 2022

REST Publisher; ISBN: 978-81-948459-4-2

Website: http://restpublisher.com/book-series/daai/



A Research Paper on White Box Testing

C Kalpana

S.S.T College of Arts & Commerce, Maharashtra, India

Corresponding Author's Email: rkalpz@gmail.com

Abstract: Software testing is necessary to ensure software quality. Software testing is the process of finding bugs while running a program in order to obtain bug-free software. It is intended to evaluate program performance. Software testing is a set of activities performed with the goal of finding bugs in software. It also checks and verifies that the program works correctly and has no errors. It is used to determine the accuracy and effectiveness of the software and products manufactured. Two important testing techniques are white box testing and black box testing. In my article, he described one of his most important techniques: white-box testing. White-box testing provides testers with a clear view of all code and internal implementation details.

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

1. INTRODUCTION

Software testing is not an easy task, it is a time-consuming task, and it is very difficult to tell when testing is complete. The main purpose of testing is to detect software bugs so that errors can be found and corrected. Testing cannot determine that the product will function properly under all conditions. It's just that it doesn't work properly under certain conditions. The scope of software testing often includes examining code, executing that code in different environments and conditions, and examining various aspects of the code. In today's software development culture, the test organization can be separated from the development team. Test team members have different roles. Information gathered from software testing can be used to modify the software development process. The first step in white-box testing is identifying the source code, structures, and other tools involved in developing software. The second tester should be familiar with various tools and techniques of white box testing. Its main purpose is to improve security, input/output flow, and improve design and usability. What Is White Box Testing? This test visualizes the internal details and structure of the system. Therefore, it is very efficient in detecting and resolving problems, as bugs can often be found before problems occur. Therefore, this method can be defined as testing software with knowledge of internals and coding. White box testing is also known as clear box testing, white box analysis, or clear box analysis. This is a bug-finding strategy in which testers have complete knowledge of how program components interact. In this testing method, test cases are calculated based on analysis of the internal structure of the system based on code coverage, branch coverage, path coverage, condition coverage, etc. It includes some features such as

1. Testers have extensive knowledge of the inner workings of the software.
2. Data domains and inner bounds can be easily tested.
3. Great for testing algorithms
4. Mainly done by testers and developers.
5. High granularity. Various Names of White Box Testing

Logic Driven Test

1. Clear Box Test
2. Open Box Test
3. Glass Box Test
4. Transparent Box Test
5. Code Base Test
6. Structural Test

Testing Principles There are seven key testing principles to consider when testing your product. ▪ Testing reveals bugs

1. Perfect testing is not possible
2. Testing needs early
3. Bug accumulation
4. Pesticide paradox

5. Testing is contextual
6. Bugs it is wrong not to have

2. WHITE BOX TESTING TECHNIQUES

A number of techniques can be used in white-box testing because specific knowledge and attention to the structure of the software under test mitigates intractable problems. The intent to exhaust some aspect of the software is still strong in white box testing, and some exhaustion can be achieved. Execute each line of code at least once, iterate through all branching statements, or cover all possible combinations of true and false conditional predicates. Various white box testing techniques are shown below.

1. Instruction Coverage
2. Branch Coverage
3. Path Coverage
4. Branch Coverage

Branch coverage is also called decision coverage or alled coverage. Branch coverage is a testing technique aimed at verifying that all possible branches from all decision points are executed at least once, thereby ensuring that all reachable code is executed. We guarantee. That is, each branch is taken in both true and false directions. It helps to validate all branches in your code and make sure there are no branches leading to abnormal Capture all possible control paths, behavior of your application. including all loop paths captured Formula: zero, once, and multiple elements ith path coverage techniques, and ranch test = (number of prepare test cases based on the decisions tested / total logical complexity measure of the number of procedural designs To do. This type of decisions) x 100% test ensures that every statement in your program is executed at least Statement Coverage once. Path testing is a structural Statement coverage is also testing technique that attempts to called line coverage or segment find all possible paths through a coverage. program's source code. The idea is Statement coverage only covers that you can test as many individual true conditions. paths as possible to maximize Statement coverage is also coverage for each test case. This called line coverage or segment gives you the best chance of finding coverage. all errors in your code. Being able to • Statement coverage only covers test your code means that you have true conditions a base on which you can rigorously define your test cases. This allows mathematical analysis of both test cases and their results, resulting in more accurate measurements. For example, if a method has N decisions, it can have 2N paths. Various strategies have been developed to identify a useful subset of paths for testing when path coverage is impractical. Statement coverage is a white-box Loop Coverage test design technique in which every Basic Path Coverage executable statement in the source code is executed at least once. It is • Data Flow Coverage used to calculate and measure the number of instructions in your source code that can be executed Basic Path Testing according to your requirements. Testing base paths means making Statement coverage can be sure that every independent path is calculated as follows: tested by a code module. An independent path is any path through the code that introduces one or more new sets of processing instructions. Basic path testing is a Path Coverage structured testing technique used to design test cases. It also checks Test cases are run such that every possible path at least once. pass is executed at least once. Building and running tests for all possible paths gives 100% statement and branch coverage. This type of testing technique is used to find most errors and bugs. Data Flow Testing Use control flow graphs to explore errors and errors in your data. Data flow testing is used to detect misuse of data in your program. When testing data flows, all tests used data flow graphs to represent data dependencies between a set 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:

1. Simple Loop
2. Nested Loop
3. Concatenated Loop
4. Unstructured Loop

3. WHITE BOX TESTING STRATEGIES

Unit Testing: This is the level of software testing where individual units/components of software are tested. Its purpose is to verify that each unit of software is functioning as intended. A unit is the smallest testable piece of software. It usually has one or several inputs and usually one output. This is usually done by a tester and the main purpose of unit testing is to show that all parts have been tested individually and that all parts are functionally correct.

Integration Testing: This is the phase of software testing where individual software modules are combined and tested as a group. Occurs after unit tests and before validation tests. The main purpose of this test is to reveal malfunctions and errors in the integrated components. Integration testing follows two approaches o top-down approach o bottom-up approach This type of testing is performed by an integration tester or testing team.

Regression Testing: It is a type of software testing that ensures that previously developed and tested software behaves the same after being modified or interfaced with other software. Changes may include software improvements, patches, configuration changes, etc. This ensures that old code continues to work after new code changes are made. These tests are run to ensure that new code changes do not
adversely affect existing functionality. Regression testing required

1. When requirements change and the code is changed to meet the requirements.
2. When new features are added to the software.
3. Troubleshooting.
4. Fixed performance issues.

Type of Regression Test
o Final Regression Test o Regression Test

4. ADVANTAGES OF WHITE BOX TESTING

1. You can start testing even before the GUI is complete.
2. Since internal functionality is taken into account, all possible conditions are considered and test cases are generated. Therefore, all functions are tested.
3. Identify the correctness of specific steps within the application.
4. Thoroughly check whether the program can run successfully in other parts of the application.
5. Identify errors in hidden code, thus speeding up the debugging process.
6. Optimize your program and increase its efficiency by removing extra lines of code that your program does not need.

5. DISADVANTAGES OF WHITE BOX TESTING

1. It is not possible for the tester to look into every bit of the code and identify the hidden errors. This may result in failure of the application.
2. Sometimes a change in the code may be required and thus all the scenarios may need to be tested again.
3. White box testing is an exhaustive method.
4. It takes time to tester to develop the test cases.
5. Test cases are a waste if changes in the implementation code are done frequently.
6. If the application is large then complete testing through white box techniques is not feasible.

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、ワイリー、2003
[3]. “A Path-Oriented Automatic Random Testing Based on Propagation of Double Constraints,” Ruilian Zhao, Yuandong Huang International Journal of Software Engineering & Applications (IJSEA), Vol. 3, No. 2, March 2012
[4]. Black boxes and white - Box Testing Methodology - 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]. 6 Paul Ammann, Jeff Offutt, "Introduction to Software Testing", Cambridge University Press, 2013
[7]. M. Herman, P. McMinn, J. Souza, and S. Yoo. Searchbased software engineering: techniques, taxonomies, tutorials Empirical Software Engineering and Validation, Page 1 {59. 2012.

[8]. C. Henard、M. Papadakis、G. Perrouin, J. Klein, P. Heymans, and Y. Le Traon. Avoid combinatorial explosion: Use similarity to generate and prioritize t-wise test configurations for software product lines IEEE transformer. soft. Eng., 40(7):650{670, July 2014.

[9]. www.wikipedia.com

[10]. "Fundamentals of Software Engineering" by Rajiv Maul