White-box Black-box testing

Kai Zhang

ITMD536/A20498615

2021.10.31

Nazneen Hashmi

Abstract

Software testing is a most regularly used technique to verify, validate, maintain a quality product executing well. There are many approaches to software testing, the paper focuses on structural testing and data driven testing. Black box testing is a kind of data driven testing which regarding all software specifications and requirements. Black Box Testing just concentrates upon inputs and output of the software system and not bothering at all regarding internal knowledge of the software. White Box Testing is usually performed by the developer or someone who has knowledge of the coding language used in the given application. It is a process of testing the core design, build and coding database construction pattern, and implementation of a software application as a product. In this paper, we have described and compared the two most prevalent and commonly used software testing techniques for detecting errors.

White-box Black-box testing

**The introduction**

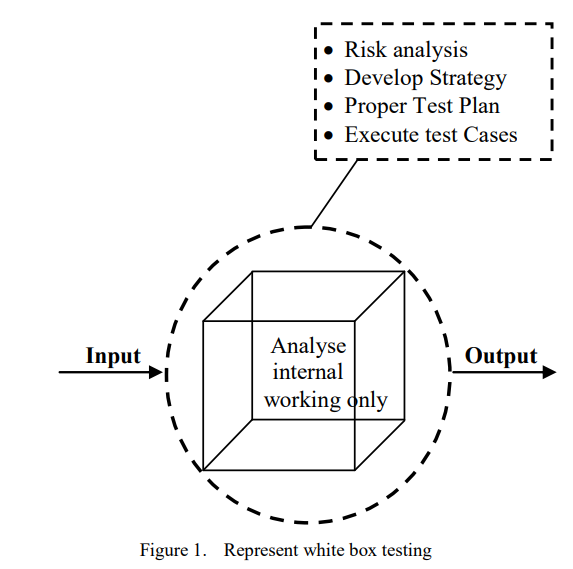
Software testing is a most often used technique for verifying and validating the quality of software. With the development of science and technology, software products have been applied in various fields of society, from the daily life of mobile phone calls, ATM operation to robots, aerospace, we can feel the convenience and speed brought by software. It can be said that without software, our life will be unimaginable which is a significant activity of the software development life cycle. With the increase and enhancement of software functions, software systems are becoming larger and more complex. As we all know, there is no perfect software in the world, and the reasons can be roughly divided into too much input, too many output results, too many ways of software implementation, and no objective standards in software specifications [1]. Every day, software bugs can cause problems: mobile phones don't talk properly, Mars probes are lost in space, banks repeatedly debit or withdraw unusual amounts of cash, operating systems restart unexpectedly, etc. Therefore, it is particularly important to test the software adequately. The so-called software testing refers to the use of engineering methods and testing tools to test the function, performance, reliability and other aspects of the software product, and find out the problems, defects or deviations of the product to the maximum extent. Additionally, it works in establishing the confidence of a developer that a program does what it is intended to do.

The main objective of software testing is to affirm the quality of software system by systematically testing the software in carefully controlled circumstances, another objective is to identify the completeness and correctness of the software, and finally it uncovers undiscovered errors[2][3].

**1 Black box test and white box test**

**1.1 White Box Test:**

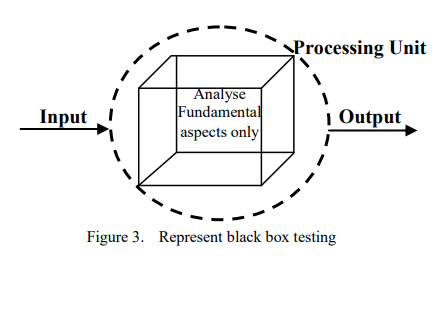
It means actually running the program under test, testing it from the program's source code without using the user interface. This type of testing involves identifying weaknesses or errors in the internal code in terms of algorithms, overflows, paths, and conditions from the code syntax and correcting them.



White box testing is also known as "transparent box testing" or "structural testing". In testing, the program is regarded as a "white box" (or "transparent box"), the internal structure of the program is transparent, and the software product is tested. But there is a necessary condition, that is, the tester before testing must be familiar with and master the internal structure of the product, and through a variety of programming language design test cases, realization of software testing, test for the most part, are references to the data definition and data testing, secondly, include the program logic path coverage testing.

**1.2 Black Box Test**

Black pot testing is what we call "functional testing." , it mainly test is whether the product function can be realized, don't care products internal structure, only to make a test on the features of the product evaluation, testing, will often program as a can't open the black box, the tester can't see the internal structure of the product from the outside circumstances, to the software interface and software function test due to can't see internal structure of the product, Software Requirements Specification has become the main basis for software testers to define testing objectives.



Black basin testing, also known as functional testing, data-driven testing, or specification-based testing, is rigorously tested by using the entire software or certain software features, without examining the program's source code or having a clear understanding of how the program's source code was designed. Testers learn how the software works by entering their data and seeing the output. Should be more suitable for testing web games.

**1.3 Differences between the two**

**1.3.1 By definition**

White-box testing involves finding flaws or errors in internal code from code syntax in algorithms, overflows, paths, conditions, and so on, and correcting them. While black box testing focuses on testing software functions, it does not involve the internal structure and content characteristics of the program. Black box testing is not a replacement for white box testing; it is complementary to white box testing and is likely to find other types of errors that white box testing is less likely to find.

**1.3.2 For testing purposes**

The purpose of black box testing is to detect whether there are incorrect or missing features; On data or parameters, whether input can be received correctly; Whether there are data structure errors or external information access errors; Whether the performance can meet the requirements; Whether there are initialization or termination errors. The purpose of white-box testing is to determine whether the actual state is the same as the expected state by examining the state of the program at different points, regardless of its functionality.

Black-box testing aims at testing the software’s functionality and behavior according to its applicable requirements. Typically, black-box tests verify boundary-level conditions and validate that the external interfaces work as specified. In contrast, testers perform white-box testing when doing unit and unit integration testing because they are interested in what goes on inside the box. Testers may dig deeply by injecting faults and assess code coverage (i.e., tests that look at what percentage of the code is actually executed by the test cases) as part of their white-box analysis. Gray-box testing is performed when testers deem it prudent to look inside the box at that which is needed to design the test cases for black-box testing.[9]

**1.3.3 Detection methods**

White box testing is an exhaustive path test, while black box testing is an exhaustive input test. These two approaches are based on completely different viewpoints and reflect two extremes of the issue. They have different focuses and advantages, but cannot be substituted for each other. In modern testing philosophy, the two methods are not separate but used interleaved.

**1.4 Advantages and disadvantages of white box test**

**1.4.1 Advantages:**

(1) Careful consideration of software implementation;

(2) Each branch and path in the code can be detected;

(3) Reveal hidden errors in the code;

(4) Test the code thoroughly with the possibility of covering most paths.

(5) Testing provide you with help in the code optimization.

(6) White testing can be executed early without depending on GUI of application.

**1.4.2 Shortcomings**

(1) Expensive;

(2) Failed to detect missed paths and data sensitivity errors in the code;

(3) It is not possible for testing every path of software program;

(4) Do not verify the correctness of specifications;

(5) If there is much change in implementation, it is quite inflexible to maintain test scripts.

**1.5 Advantages of black box test:**

**1.5.1 Advantages**

(1) Higher efficiency than white box test for a large system

(2) Testers can be non technical person. Programming and implementation knowledge is not required for this testing technique.

(3) Testers and programmers are independent of each other

(4) It is easy to understand and accept the test from the user's point of view requirements

(5) help to expose the inconsistency or ambiguity of specifications

(6) Test cases can be performed immediately after the specification is completed

**1.5.2 Shortcomings**

(1) Only a small number of inputs are tested, making it nearly impossible to test every possible input.

(2) It is very difficult to design test cases without clear and concise specifications.

(3) If testers are not informed of the use cases that developers have executed, there will be unnecessary duplication of test data.

(4) There are many program paths that are not tested.

(5) Can not directly test specific program segments, and these program segments may be very complex, may hide more problems.

(6) Most research-related tests are directed at white-box tests.

(7) Redundant tests can be formed if software developer has already run the test cases

**2. Software testing methods**

**2.1 Black box test**

Black-box testing means that testers can focus not on how the software is implemented and run, but on the inputs and outputs. Equivalence class partitioning [6] is a typical and important black-box testing method, which divides all possible input data (valid and invalid) of a program into several equivalence classes. Then select representative data from each part as test cases for reasonable classification. Test cases are composed of effective equivalence classes and invalid equivalence classes, so as to ensure the integrity and representativeness of test cases. Boundary value analysis [6] is a black box test method that tests the boundary values of input or output. In general, the boundary value analysis method is a supplement to the equivalence class partition method. In this case, the test cases are usually from the boundary of the equivalence class.

Take the indicator light of an electronic product as an example, assume that the input parameter of an interface function is (integer data), which controls the red, yellow and green lights of the electronic product, and is defined as:

(1) When -10≤I < 0, the red light will be on;

(2) When 0≤I < 10, the green light is on;

(3) when 10≤I < 20, the yellow light is on;

(4) When I is other values, none of the lamps will be on.

In order to verify the control results of the input parameters on the display lamp, the tester will design different input parameters to verify. Taking equivalence class division as an example, the tester may use -5, 6 and 18 values for verification (of course, other data that meet the conditions can also be used for verification), and the expected results are obviously red light on, green light on and yellow light on respectively. Considering the test performance, test cost, and delivery schedule, it is impossible for the tester to verify all integers , because normally (excluding hardware environment exceptions),I is -5 and [-10, 5) (5,0) norm, and the results of any integer data within the range are the same. The result of 6 is consistent with any integer data within the range of [0, 6)(6, 10); the result of 18 is consistent with any integer data within the range of [10, 18) (18, 20). This is the type of data that makes the light on. The other type of data, such as -20 and 30, does not turn on after input. This type of validation data falls into the scope of invalid equivalence class.

As a supplement to equivalence class partitioning, the tester will verify logarithms -11, -10, 0, 10, and 20, and the result should be:

When I=-11, none of the lights turn on;

When I=-10, the red light is on;

When I=0, the green light is on;

When I=10, the yellow light is on;

When I is equal to 20, none of them turn on.

The boundary value analysis method is used in this way.

**2.2 White box test**

White box testing, testers need to understand the specific design and implementation principle of software. White box tests are classified into static white box tests and dynamic white box tests.

**2.2.1 Static White Box Test**

Static white-box testing is a process of systematically scrutinizing software design and implementation without executing programs to find software defects [1]. Testers need to have some programming background and understand the implementation of the product. Testing focuses on checking requirements, design, and source code:

(1) Whether the implementation is consistent with the requirement design;

(2) Whether there are normative problems in the code;

(3) Whether the program has a reasonable solution for abnormal or error branches;

(4) code readability and easy to understand.

**2.2.2 Dynamic White Box Test**

Dynamic white-box testing refers to using the information obtained from viewing code functions and implementation methods to confirm what needs to be tested and how to carry out the test [1]. For example, for the following program:

#define OK 1

#deline ERROR -1

#define A 1000

#define B 3500

int Fun(int flag, long int address, long int offset)

{

if(0== flag) {

Fun Wirte(address, offset, A);

return OK；

}

else if(l== flag){

Fun Wirte(address, offset, B);

return OK；

}

else

printf("The entered flag is incorrect \n");

return ERROR；

}

What this function does

If flag is 0, write 1000 to address+offset via Fun Wirte, return 1;

If flag is 1, write 3500 to address+offset via Fun Wirte, return 1;

If the input parameter fIag is any other value, a prompt message is printed and -1 is returned.

For testers, the idea of test case design (before executing the program, clear the data at address+offset) :

(1) Input parameter FIag-0. After executing the program, query whether the value of address+offet is 1000.

(2) Input parameter FIag-1 and run the program to check whether the value of address+offset is 3500.

(3) Input parameter FIag-2 and check whether the value of address+offset is 0 after executing the program.

Obviously, this requires testers to understand the design idea and implementation principle of the function Fun, and have certain programming foundation. In fact, test cases with flags equal to 0, 1 and 2 can also meet the requirements of statement coverage, condition coverage and judgment coverage in logical coverage [7].

**3. A comparative study of the two**

As is known to all, black box testing and white box testing is a software testing methods of two kinds of relative method, should be used for different applications, the development software of different test methods to test it, and with what kind of test method mainly depends on the function of the software product design, logic design, etc., is now on how to choose from these two kinds of test methods to make inquiry

**3.1 Different starting points for testing**

With the continuous maturity and improvement of software technology, the scale and complexity of software testing are constantly increasing, and software testing has gradually formed a complete system and gradually normalized.

Starting from the definition of the two, the black read test starting point is made clear, the function can be achieved, due to not privy to the logical structure of the product, so this method can't help testers to conduct a comprehensive test, and white box testing, white-box testing starting point is: according to the internal structure of the product can test the product function implementation, due to the test With pre-knowledge of the product, testers can complete the testing process more fully.

It is obvious that these two kinds of test method test starting point is different, which makes the two methods have different concerns during test, then emphasis will be different, use different testing methods, design the thinking of the test cases are basically fixed, the fixed thinking can sometimes help testers faster develop test cases, but could help The tester found some errors that were easy to overlook.

**3.2 Different test coverage**

Those with testing experience must know that the test coverage of black box test and white box test is different. Even for the same program, test cases designed with different testing methods are also different. Why are there differences? Specific analysis is made below.

First of all, the SOFTWARE Specification (S Pecific ATIAN) is the basis for the black box test. Therefore, the functional and performance requirements recorded in the software specification are the main basis for the design of test cases. The number of requirements in the software specification determines the number of black box test cases[8].

Unlike black mind testing, white box testing focuses more on the code itself, so the number of test cases is closely related to the complexity of the code, the number of code, and so on. These factors not only increase the number of test cases, but also increase the execution time of the test task, which may make testers unable to complete the test task on time.

From a theoretical point of view, white box testing can be complete coverage testing. However, it may be difficult to achieve full coverage due to other unknown and uncontrollable factors. However, it is still an important means for project development to verify test coverage.

It can be concluded that the coverage of white box test is wide and complete, that is, the coverage is wide and the test is comprehensive; And black box testing coverage is prospective and Ming, namely, accurate test cases, design goal is clear, although the white box testing more comprehensive, but black basin test has its advantages, because the number of test case design of black box testing is less, so can effectively reduce test time, and reduce the test cost, 3.3 two test methods for testers to acquisition of the tester The requirements are shown in Table 3.1.

Table 3.1 Different requirements for testers of the two test methods

|  |  |
| --- | --- |
| Category | Requirements |
| Black box testing | 1. Testers do not need to know the program source code and test through the front-end interface.  2. In most cases, the use of the software is evaluated from the user's point of view. (Simulated user)  3. You don't need to understand the internal structure of the program.  4. Master the skills of manual test and automatic test. |
| White box testing | 1. The tester has access to the program source code, can understand the code, can program with programming language, and is familiar with various scripting languages.  2. The main testers are the developers themselves, who test according to requirements while writing the code.  3. Before testing, understand the internal structure of the program.  4. Master the skills of manual test and automatic test. In particular, coding tests.  5. Familiar with automated test tools. |

Seen two test methods for the requirement of the testers might say, black box testing code, do not need to research the black box testing than white box testing is simple, that is not the case, sometimes black box testing is more challenging than white box testing, due to the internal structure of the unknown, will increase the difficulty of the test, and the number of testers through the IT industry to explore found that there are Many people who change their careers will choose to do "black box testing" because it has a lower starting point and is easier to get started. However, professional testers are still needed to complete the test better.[9]

**3.4 Different application stages**

Software development is a stage, the development of the software is developed step by step, from scratch, this is the need to process, according to the development process of the time, it can be divided into before, during and post three stages, the three stages of different requirements for software developers, the requirement of software testers is different also, and the two kinds of relative testing method is in a different order Sections play their role, as shown in Figure 3.1.

Figure 3.1 Application of two trial methods in software development stage

In the early stages of software development, the main demand analysis, general design and detailed design, the early stage of the amount of code is not a lot, testers will generally use white box test test, through which can better finish test task, early written software requirements specifications according to customer's requirements and design documents for testing, therefore, previous test task is very heavy Want, it is not only related to the customer's interests directly, but also has the influence on the back of the work, ready to front, can greatly help the tester after completion of task, also can effectively avoid some unnecessary software defects, the most important point is that the early stage of the test found that the more the question, the less the cost of repair will also, because of repair It's not that complicated, and the cost will be lower in the later period.[12]

In the medium term, testing is usually done in a combination of the two methods, which helps testers find errors and problems more quickly.

Will be carried out in the late stages of development, system testing and validation testing, the black box testing is play an important role in the late check pay more attention to the integrity of the software product functionality and usability, using black box testing can be more efficient and more accurate testing software functions and the use of the late, will test, from the viewpoint of "user" 1 as the user As a result, black box testing plays an irreplaceable role in system testing and validation testing. Although the method used in each stage is different, the method used in each stage is not invariable, which is selected according to the requirements of the software itself. The two methods complement each other and often make the test more perfect.[11]

**4. Conclusion**

White pot test and black box test are two different testing methods in software testing. Black, white box test has essential difference, but also interrelated, complementary, they from different angles for analysis and design, each has the focus, in the modern test concept often cross use of these two methods, in order to achieve better test effect, make the application system more complete general, product quality is higher.

Software testing throughout the entire software development process, the issues of the testing process are not necessarily caused by programmers write code, there are also may be due to the early stage of requirements analysis does not reach the designated position, logical structure design is not reasonable, so, in the process of development to constantly test, and make the work of good writing test report, so as to help the development As the team progresses smoothly, software testers are always present and an integral part of the project's "back office".

According to the starting point from the two testing methods of testing, coverage, applicable stage, and the study of the requirements of the testers can find the different between the two, although black box testing is different from white basin test test method, but you can see white box testing is supplement and perfection of black box testing, choose what kind of test method depends on the software requirements specifications, in a test When testing a software product, one or the other methods may be selected, but in most cases, the two methods need to be combined for testing, so that the tester can find as many bugs as possible, will better help developers to complete their coding work, how to choose, according to the needs of the test task to help choose, can also refer to the opinions of some developers, and to ensure the quality of software in the first place.

**References**

1. Science - Computer Science; Studies from Weizmann Institute of Science Yield New Information about Computer Science (White-box Vs. Black-box Complexity of Search Problems: Ramsey and Graph Property Testing)[J]. Computer Weekly News,2019.
2. Software Testing Methodologies by Azhar available at http://azharpaperpresentation.blogspot.com/2010/04/software-testingmethodologies.html
3. Mohd. Ehmer Khan, “Different Approaches to White Box testing Technique for Finding Errors,” IJSEIA, Vol. 5, No. 3, pp 1-13, July 2011

[4]Arpita Dutta,Anwesha Patel,Rajib Mall. Investigation into the effectiveness of white-box T-way testing[J]. IET Software,2019,13(6).

[5]Aitor Arrieta,Shuai Wang,Urtzi Markiegi,Ainhoa Arruabarrena,Leire Etxeberria,Goiuria Sagardui. Pareto efficient multi-objective black-box test case selection for simulation-based testing[J]. Information and Software Technology,2019,114.

[6]田春竹,邢航.浅析白盒测试在软件测试中的应用[J].中国信息化,2019(08):48-50.

[7]Ilan Komargodski,Moni Naor,Eylon Yogev. White-Box vs. Black-Box Complexity of Search Problems[J]. Journal of the ACM (JACM),2019,66(5).

[8]淡海英.软件测试中的白盒测试分析[J].时代农机,2018,45(11):244.

[9]范海燕.动态软件测试中的白盒测试和黑盒测试探讨[J].南方农机,2018,49(18):76.

[10]Zhiyi Zhang,Ju Qian. Test Suite Augmentation via Integrating Black- and White-Box Testing Techniques[J]. International Journal of Performability Engineering,2018,14(6).

[11]Zhiyi Zhang Ju Qian. Test Suite Augmentation via Integrating Black- and White-Box Testing Techniques[J]. International Journal of Performability Engineering,2018,14(6).

[12]冯潭秋.基于图论算法在白盒测试技术应用中的研究[J].数字技术与应用,2016(02):134-135.

[13] 软件测试实用技术与常用模板M.机械工业出版社,李龙, 2010.

[14] 软件测试(M.机械工业出版社.(美)RonPatton著, 2002.