

Review on Concepts of Software Testing

Sandeep Kaur^{#1}, Navdeep Kaur^{*2}, Harjeet Kaur^{#3}

[#]Department of Mathematics, Baba Farid Group of Institutions

¹sandeep8216@gmail.com

²harmeetb84@gmail.com

³harjeetsidhu1996@gmail.com

Abstract:

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we cannot completely test a program with moderate complexity. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality.

Keywords: *Software Testing Techniques, Software Testing Methods and Software testing levels*

I. INTRODUCTION

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.^[1] Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.[5] Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and to verify that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- meets the requirements that guided its design and development,
- responds correctly to all kinds of inputs,
- performs its functions within an acceptable time,
- is sufficiently usable,
- can be installed and run in its intended environments, and
- Achieves the general result its stakeholder's desire.[6]

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources. As a result, software testing typically (but not exclusively) attempts to execute a program or application with the intent of finding software bugs (errors or other defects). The job of testing is an iterative process as when one bug is fixed, it can illuminate other, deeper bugs, or can even create new ones.

Software testing can provide objective, independent information about the quality of software and risk of its failure to users and/or sponsors.^[1]

Why is software testing necessary

Software Testing is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong –humans make mistakes all the time. Since we assume that our work may have mistakes, hence we all need to check our own work. However some mistakes come from bad assumptions and blind spots, so we might make the same mistakes when we check our own work as we made when we did it. So we may not notice the flaws in what we have done. Ideally, we should get someone else to check our work because another person is more likely to spot the flaws .There are several reasons which clearly tells us as why Software Testing is important and what are the major things that we should consider while testing of any product or application.

Software testing is very important because of the following reasons

1. Software testing is really required to point out the defects and errors that were made during the development phases.

2. It's essential since it makes sure of the Customer's reliability and their satisfaction in the application.
3. It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence.
4. Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
5. Testing is required for an effective performance of software application or product.
6. It's important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
7. It's required to stay in the business.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester.[6] These tests can be functional or non-functional, though usually functional.

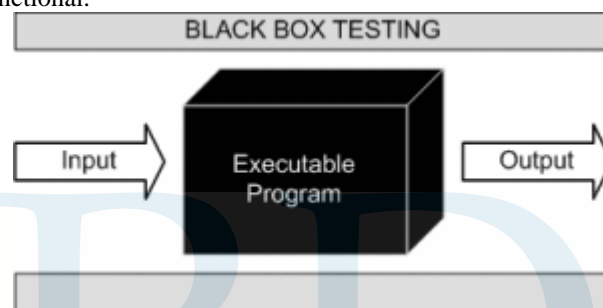


Fig. 1 Black Box Testing

This method is named so because the software program, in the eyes of the tester, is like a black box; inside which one cannot see. This method attempts to find errors in the following categories:

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external database access
- Behavior or performance errors
- Initialization and termination errors

Definition by ISTQB

Black box testing: Testing, either functional or non-functional, without reference to the internal structure of the component or system. Black box test design technique: Procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

Example

A tester, without knowledge of the internal structures of a website, tests the web pages by using a browser; providing inputs (clicks, keystrokes) and verifying the outputs against the expected outcome.

Levels

Black Box Testing method is applicable to the following levels of software testing:

- Integration Testing
- System Testing
- Acceptance Testing

The higher the level, and hence the bigger and more complex the box, the more black box testing method comes into use.

Black box testing techniques

Following are some techniques that can be used for designing black box tests.

- *Equivalence partitioning*: It is a software test design technique that involves dividing input values into valid and invalid partitions and selecting representative values from each partition as test data.

- *Boundary Value Analysis*: It is a software test design technique that involves determination of boundaries for input values and selecting values that are at the boundaries and just inside/ outside of the boundaries as test data.
- *Cause Effect Graphing*: It is a software test design technique that involves identifying the cases (input conditions) and effects (output conditions), producing a Cause-Effect Graph, and generating test cases accordingly.

Black box testing advantages

- Tests are done from a user's point of view and will help in exposing discrepancies in the specifications.
- Tester need not know programming languages or how the software has been implemented.
- Tests can be conducted by a body independent from the developers, allowing for an objective perspective and the avoidance of developer-bias.
- Test cases can be designed as soon as the specifications are complete.

Black box testing disadvantages

- Only a small number of possible inputs can be tested and many program paths will be left untested.
- Without clear specifications, which is the situation in many projects, test cases will be difficult to design.
- Tests can be redundant if the software designer/ developer has already run a test case.
- Ever wondered why a soothsayer closes the eyes when foretelling events? So is almost the case in Black Box Testing.

White Box Testing

White Box Testing (also known as Clear Box Testing, Open Box Testing, Glass Box Testing, Transparent Box Testing, Code-Based Testing or Structural Testing) is a software testing method in which the internal structure/ design/ implementation of the item being tested is known to the tester. The tester chooses inputs to exercise paths through the code and determines the appropriate outputs.[1] Programming know-how and the implementation knowledge is essential. White box testing is testing beyond the user interface and into the nitty-gritty of a system. This method is named so because the software program, in the eyes of the tester, is like a white/ transparent box; inside which one clearly sees.

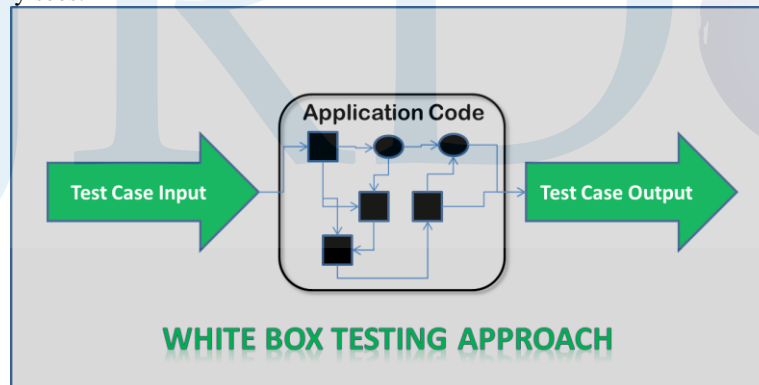


Fig. 2 White Box Testing

Definition by ISTQB

- white-box testing: Testing based on an analysis of the internal structure of the component or system.
- white-box test design technique: Procedure to derive and/or select test cases based on an analysis of the internal structure of a component or system.

Example

A tester, usually a developer as well, studies the implementation code of a certain field on a webpage, determines all legal (valid and invalid) AND illegal inputs and verifies the outputs against the expected outcomes, which is also determined by studying the implementation code.[7]

White Box Testing is like the work of a mechanic who examines the engine to see why the car is not moving.

Levels:

White Box Testing method is applicable to the following levels of software testing:

- Unit Testing: For testing paths within a unit.
- Integration Testing: For testing paths between units.
- System Testing: For testing paths between subsystems.

However, it is mainly applied to Unit Testing.

White box testing advantages:

- Testing can be commenced at an earlier stage. One need not wait for the GUI to be available.
- Testing is more thorough, with the possibility of covering most paths.

White box testing disadvantages:

- Since tests can be very complex, highly skilled resources are required, with thorough knowledge of programming and implementation.
- Test script maintenance can be a burden if the implementation changes too frequently.

Since this method of testing is closely tied with the application being testing, tools to cater to every kind of implementation[3]

II. CONCLUSION

Software testing is mainly used to find out the bugs from the software and make it error free. This paper is basically defined the software testing techniques, methods and levels. But in future we can enhance the techniques and explain the testing tools. A necessary concluding remark concerns the many fruitful relations between software testing and other research areas. By focussing on the specific problems of software testing, we have in fact overlooked many interesting opportunities arising at the border between testing and other disciplines. As there is always a scope so Further to this paper a research and study can be done on the software testing to propose a generic testing framework and techniques to support functional, performance and security testing for object oriented development framework and other platforms using some algorithm(s) with/ without use of tools in minimum amount of time.

III. REFERENCES

- [1]. Gaurav Saini and Kestina Rai (2013), "Software Testing Techniques for Test Case Generation", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, issue 9, September 2013.
- [2]. Ms. T.M.S. Ummu Salima, Ms. A.Askarunisha and Dr.N.Ramaraj, "Enhancing The Efficiency of Regression Testing Through Intelligent Agents", IEEE, International Conference on Computational Intelligence and Multimedia Applications 2007
- [3]. Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan "Software Testing Techniques and Strategies", International Journal of Engineering Research and Applications (IJERA), Vol. 2, Issue 3, May-Jun 2012, pp.980-986
- [4]. Jovanović, Irena, "Software Testing Methods and Techniques"
- [5]. Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010.
- [6] Khan, Mohd. Khan, Farmeena. 2012. A Comparative Study of White Box, Black Box and Grey Box Testing Techniques. 2012. International Journal of Advanced Computer Science and Applications (IJACSA). Vol. 3.No.6.(2012)
- [7] Akhilesh, Babu, Kolluri, K, Tameezuddin, Kalpana, Guddika dula. 2012. Effective Bug Tracking Systems. Theories and Implementation", IOSR Journal of Computer Engineering ISSN:2278-0661 Volume 4, Issue 6 (Sept/Oct 2012), pp 31-36.
- [8] Bhasin, Harsh, Khanna, Esha, Sudha. 2014. Black Box Testing based on Requirement Analysis and Design Specifications International Journal of Computer Applications. (0975 – 8887) .18,18(February 2014).