



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 Issue: IV Month of publication: April 2016

DOI:

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Survey of Software Testing Techniques

Rahul Jampani¹, Nitish Talasu², Dr Manjula R³

^{1,2,3}School of Computer Science and Engineering, VIT University, Vellore, India.

Abstract: *Software created will never be perfect without being tested and approved. This makes us to find a way for testing hidden bugs at different levels by a good technique. Testing is a process where we can spot hidden bugs, code errors or any unused code in a given software. Currently there are many techniques being used for testing errors in software, among them most practiced techniques for testing a software are black box testing, white box testing and grey box testing. In this paper we are comparing about various software testing techniques to find out their effectiveness in generating test cases and enhancing the quality of the software system.*

Keywords: *Black box testing, White box testing, Grey box testing, Red box testing.*

I. INTRODUCTION

Software testing is an important phase in software development cycle. It also guarantees the quality of software. During the testing process test suites, which contain collection of test cases are used to test the programs. Test cases consist of input and the desired output which is compared with the output given by the software, so that we can determine how good given software performs. Testing consumes 40-50% of development cycle and also consumes more effort for software's which require more reliability [13]. It does not guarantee high quality software systems. Software testing can only detect flaws and is not fully exhaustive. It is not possible to completely test a software due to various reasons such as domain of software is too large, many input paths, complex design, specification issues etc. As a system grows due to upgradations, significant amount of testing has to be done to verify system after upgradation. Testing remains effectively to assure quality of software. Testing is an important emerging research area within domain of computer science.

The main objective of testing is to find bugs and to fix them which improves quality of software. The four main objectives of testing [16]:

A. Demonstration

It demonstrates functionalities under special cases. It ensures that products are ready for use.

B. Detection

It detects errors, defects, deficiencies. It also determines capabilities of system, limitations and ascertain components quality.

C. Prevention

It gives information to prevent number of errors that effects system performance and specifications. It also identifies ways to avoid risks in future.

D. Improvising Quality

We can minimize errors by practising effective testing resulting in higher quality of software.

II. RELATED WORK

Software testing used to detect errors in the application code. We can also define software testing as a way of accessing functionality and flawlessness of a software through analysis. Main goal of testing is to guarantee quality, reliability, estimation, validation and verification. Software testing is the prime component of software quality assurance and represents a review of design, coding. Software testing's main objective is to assure the completeness and correctness of the software, and get to know the undiscovered errors [7].

Software testing techniques are mainly classified as black-box testing and white-box testing [3] [5]. Black box testing is also called as functional testing, a functional testing technique will be created based on specifications made by client [3]. With this testing technique, the tester will not have any access to source code. Black box testing mainly concentrates on outputs generated in response to selected inputs and execution environment [3]. The software tester knows only that information can be given as input to

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

the black box, and the black box will give some response accordingly to the tester. This will be done purely on requirement specification knowledge.

White box testing is also called as structural testing or glass box testing. In this testing technique test cases are developed based on the information extracted from the application code [3]. The white box tester will have knowledge on code and writes test cases with appropriate parameters [3]. This mainly concerned about control flow and data flow of a program [1] [5] [18]. To test a software more accurately, it is must to cover both specification and code functionality [3] [6] [8].

Software testing is a vast area that consists various technical and non-technical areas such as Requirement specifications, maintenance, process, design and implementation, and management issues in software engineering.

III. OVERVIEW OF VARIOUS TESTING TECHNIQUES

A. Black Box Testing

Black box testing is also called as behavioural testing, in which the interior structure, logic of software that is being tested is unknown to analyser. This testing is based on requirement specification and it's not necessary to analyse code. It is basically performed under the end user point of view. It helps to identify incomplete and unpredictable specifications [7], so that they can be rectified later. Black box testing is done from beginning of software project development cycle [13]. Testers need to gather end user requirements and based on that test scenarios have to be prepared.

Various types of black box testing that are discussed below:

- 1) *Equivalence Class Partitioning*: In this procedure information area of system is separated into equivalent classes from where experiments can be done. This reduces the quality of experiments. This procedure can be applied at any testing level and tests one condition from each class.
- 2) *Boundary Value Analysis*: This procedure mainly concentrates on boundaries or extreme boundaries values that are created in software that being tested. It also incorporates inside and outside limits.
- 3) *Cause Effect Graph*: This procedure creates a graph which mentions relationships between given inputs and their corresponding effects.
- 4) *Orthogonal Array Testing*: This procedure used when number of inputs to software is comparatively small but too complex for carrying complete testing of every possible input to software.
- 5) *All Pair Testing*: In this procedure all possible combinations of input parameters are designed and executed. Its main aim is to cover all possible inputs.
- 6) *State Transition Testing*: This is mainly used for testing navigation of GUI. It's also used in designing test cases to check validations of state transitions.
- 7) *Fuzzing*: This procedure is used to detect run code errors using malformed/semi malformed data injection in an automated/semi-automated session.

Advantages

- a) Testers need not have programming knowledge.
- b) It's done at end users perspective.
- c) It helps in finding incomplete/unfulfilled aspects of given requirement specifications for developing software.
- d) Developer team and testing team are independent to each other.
- e) Test cases can be reproducible.

Disadvantages

- a) Testing cannot be done completely as tester have limited knowledge on software being tested.
- b) Tester cannot find precise reason for errors in software.
- c) Designing test cases will be difficult as tester having least knowledge on internal logic.
- d) Backend test cases cannot be tried.

B. White Box Testing

White box testing is also called as structural testing, in which the interior structure, logic of software that is being tested is transparent to analyser. In this test cases are designed from developer's perspective. As internal structure/logic is known to tester

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

each and every part of code can be tested. It is used to find logical errors of software's source code [4]. Random typographical errors can be found and it also brings out incorrect assumptions in program. White box testing is used at all levels of software development [14]. Tester needs to look the source code and be able to find which part of codes are not working properly.

Various types of white box testing are as follows

- 1) *Static white box testing*: The main aim of static testing is to check whether code is according to functional requirements, design, coding standards, all functionalities covered and error handling.
 - a) *Desk Checking*: This testing is primarily done on code. Developer himself involves in these kind of testing.
 - b) *Code Walkthrough*: In this testing bunch of technical experts review the code.
 - c) *Formal Inspection*: In this testing code is formally inspected and all the flaws will be detected. This is very efficient and economical method for finding errors in design and code.
- 2) *Structural White Box Testing*: Testing take into account the code, code structure, internal design and how they are written.
- 3) *Control Flow/Coverage Testing*: This technique uses program control stream as model control stream. It can be applied to almost every software. It is mostly used for small programs or segments of larger programs[13].

Types of Coverage testing:

- a) Statement coverage.
 - b) Branch coverage.
 - c) Decision/Condition coverage.
 - d) Function coverage.
- 4) *Basis Path Testing*: This technique used to test the control flow based codes. Independent paths are formed using bot control flow and control flow chart that converts the code into a model [13].

Types of Basis path testing:

- a) Flow graph notation.
 - b) Cyclomatic complexity.
 - c) Deriving test cases.
 - d) Graph matrices.
- 5) *Loop Testing*: This technique is used for validation of loops in code. Test cases are designed according to the definitions and variable usage [13].

Types of Loop testing:

- a) Simple Loops.
 - b) Nested Loops.
 - c) Concatenated Loops.
 - d) Unstructured Loops.
- 6) *Data Flow Testing*: This technique represents the program and its execution diagrammatically. It checks how variables are declared and their uses. It is concerned about the evolution of a particular piece of data in an application.
- a) *Advantages*
 - i. It removes extraneous part of code that are hidden.
 - ii. It is thorough testing as every part of structure/logic is explored.
 - iii. It helps to optimize the code.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

- iv. White box testing can be started despite developing GUI.
 - b) *Disadvantages*
 - i. High knowledge of programming is needed for tester.
 - ii. It's very expensive.
 - iii. In large applications it's highly impossible to analyse code line by line.

C. Grey Box Testing

Grey box testing is also called as translucent testing. It is a blend of black box testing and white box testing [14]. Tester will have partial knowledge of internal structure /logic. It increases the area of testing by focussing on all complex system's layers by blending all existing black box and white box testing [2]. A tester can check both the output and also the process that leads to output. It is applied to most testing stages, mainly in integration testing.

Various types of grey box testing:

- 1) *Orthogonal Array Testing*: This testing technique uses all possible combinations.
- 2) *Matrix Testing*: The status report of the project is stated using this testing. This matrix gets the data from performed actions.
- 3) *Regression Testing*: This technique is used if any changes are made in software and runs test cases.
- 4) *Pattern Testing*: This testing verifies the design and architecture are implemented correctly.

Advantages

- 1) It gives benefits of both black box and white box as it's a blend of both.
- 2) It is an unbiased testing.
- 3) Using this testing, test cases can be designed excellently.
- 4) It depends on interface and working of functionalities rather than source codes.
- 5) It is non-intrusive.

Disadvantages

- a. Limited access to code.
- b. Many of the paths of code are untested.
- c. Test cases can be repeated.

D. Red Box Testing

Red box testing technique is also called as user acceptance testing. In this testing any technique can be applied to know the acceptance of software by the end user. It is generally performed by end user. It is performed after the completion of software.

Various types of red box testing:

- 1) *Alpha Testing*: It is performed by end user in a controlled environment [17]. It is performed among end users to confirm the product specifications that have been proposed by them.
- 2) *Beta Testing*: It is performed by end user in a uncontrolled environment i.e real environment [18]. The Software is released to a few people in this way testing ensures that software has no faults. Sometimes it is released to public to increase the feedback for the future usage. This is considered as second phase of software testing in which some of the users whom this software meant for tries the product.

Advantages

- a) It checks the acceptance of end user.
- b) Vendors have good communication with both the clients and developers, as requirement definition is improved through the acceptance testing and authorised by client.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

III. COMPARISON OF TESTING TECHNIQUES

Property	Black box testing technique	White box testing technique	Grey box testing technique	Red box testing technique
Knowledge on Internal Structure/logic	No need	Need to have full knowledge	Need to have partial knowledge	Need to have partial knowledge
Granularity	Low	High	Medium	Low
Users	Tester, Developer and user	Tester and Developer	Tester, Developer and user	End user
Exhaustiveness	Least	Most	Moderate	Moderate
Testing Algorithms	Don't Suits	Suits	Don't Suits	-
Domains in which technique is used	Functional and Business	All Domains	A bit deeper in Functional and Business	All Domains
Internal Behaviour	Ignored	Fully known	Partially known or Fully unknown	Ignored
Domain where technique suits	Quick web service prototyping and rapid test scenario	Applications where internal structure is known	Web applications	All
Synonyms	Opaque box, Closed box, Input Output, Behavioural testing	Glass box, Clear box, Transparent Box, Structural testing	Translucent box testing	Acceptance testing

TABLE I: Comparison of Testing Techniques

Table-I presents the comparative analysis of three major testing techniques. All testing techniques cannot be implemented for

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

testing in software development lifecycle. Black box testing is a technique where tester carries his testing activity without any knowledge of internal structure and logic. It is most commonly used technique at end user level to ensure all requirements stated are done. White box testing is done at developer's or expert's level who has knowledge over structure of code, functionalities of each module. It is helpful to go and rectify the error once the error is known. Grey box testing can be done at developer's level or end user's level where tester will have partial knowledge over backend of product. It can state the error and can also rectify that. Red box testing is done purely at end user level after system testing. This checks products quality by verifying requirement specifications.

IV. CONCLUSION

Software testing plays an important role in software development life cycle. It optimizes time and cost by early detection of errors and rectifying them. Testing ensures that product has no errors and delivered to customer, satisfying his requirements. Testing is an art where tester needs creativity, experience with proper knowledge on techniques. The testing purpose can be verification, validation and quality assurance. Our paper compared three main testing techniques that helps in testing a software in more effective manner.

REFERENCES

- [1] D. Shao, S. Khurshid, and D. E. Perry, "A Case for White-box Testing Using Declarative Specifications Poster Abstract," in Testing: Academic and Industrial Conference Practice and Research Techniques - MUTATION, 2007. TAICPART-MUTATION 2007, 2007, p. 137.
- [2] F. Saglietti, N. Oster and F. Pinte, "White and grey-box verification and validation approaches for safety-and security- critical software systems," Information Security Technical Report, Vol. 13, no. 1, pp 10-16, 2008.
- [3] H. Liu and H. B. Kuan Tan, "Covering code behavior on input validation in functional testing," Information and Software Technology, vol. 51, no. 2, pp. 546-553, Feb. 2009
- [4] Irena Jovanovic, "Software Testing Methods and Techniques", IPSI BGD Internet Research Society, Vol. 5, No. 1, pp 30-41, January 2009.
- [5] J. H. Hayes and A. J. Offutt, "Increased software reliability through input validation analysis and testing," in Software Reliability Engineering, 1999. Proceedings. 10th International Symposium on, 1999, pp. 199 -209.
- [6] M. Shaw, "What makes good research in software engineering?," International Journal on Software Tools for Technology Transfer (STTT), vol. 4, no. 1, pp. 1-7, 2002.
- [7] Mohd. Ehmer Khan, "Difference Approaches to Black box testing technique for finding Errors", IJSEA, Vol. 2, No. 4, pp 31-40, October 2011.
- [8] P. Jorgensen, Software testing: a craftman's approach, CRC Press, 2002. p. 359.
- [9] S.Liu and Y.Chen , "A Relation-based method combining functional and structural testing for test case generation," Journal of Systems and Software, Vol.81, No. 2,pp, 234-248, Feb.2008.
- [10] S.M.K Quadri and shiek umar Farooq, "Software Testing- Goals and Techniques," International Journal of Computer Applications (0975-8887) Vol. 6, No.9, September 2011.
- [11] S.M.K Quadri and shiek umar Farooq, "Testing Techniques Selection: A Systematic Approach", Proceedings of the 5th National Conference: INIACom-2011.pp-279-281, March 10-11, 2011.
- [12] Shivkumar Hamukhrai Trivedi, "Software Testing Techniques", International journal of Computer Science and Software Engineering. Vol. 2. Issue 10, October 2012, ISSN:2277 128X
- [13] Srinivas Nidhra and Jagruthi Dondeti, "Black box and White box techniques-A Literature review.", International Journal of Embedded Systems and Applications, Vol 2, No. 2, June 2012.
- [14] Surendra Singh Rathod, "A Scenario of Different types of Testing Techniques in Software Engineering", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 6, June 2014, ISSN:2277 128X.
- [15] T. Murnane and K. Read "On the effectiveness of mutation analysis as a black box testing technique," in software Engineering Conference. 2001. Proceedings. 2001 Australian. 2001, pp. 12-20.
- [16] Taraq Hussain and Dr. Satyaveer Singh, "A Comparative Study of Software Testing Techniques Viz. White box testing Black box testing and Grey box testing", International Journal of Allied Practice, Research and Review, Vol. 2, Issue 4, 2015, pp 26-33, ISSN:2350-1294.
- [17] User Acceptance Testing available at "<http://guide.agilealliance.org/guide/acceptance.html>"
- [18] User Acceptance Testing available at "<https://www.technopedia.com/definition/3887/user-acceptance-testing-uat>"
- [19] Vineta Arnican, "Complexity of Equivalence Class and Boundary Value Testing Methods", Scientific Paper, University of Latvia, 2009. Vol.751 Computer Science and Information Technologies 80-101P.



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)