International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Selecting Efficient and Effective Software Testing Techniques for Finding Errors

[1]Mrs Sudha Katkuri,
[1] Research Scholar,
Rayalaseema University,
Kurnool. Reg. Id: PP. COMP.SCI. 0262

[2]Prof P. Premchand
[2] Professor & Dean,
Faculty of Informatics,
Dept. of CSE, UCE, Osmania University.

***Abstract* - Software testing is the process taken up for evaluating an attribute or capability of a program and ensures that it meets the required result. It is often impossible to find all the errors in the program. The major disadvantage in software testing domain is how to find a suitable set of test cases to find errors in a software system. Thus, the selection of right testing technique at the right time will make the software testing efficient and effective both in terms of time and cost. In this paper the different software testing techniques are described by the virtue of its usage and the main goal of this paper is to analyse and compare the testing technique to find out the best one to trace errors from the software.**

Keywords: Testing techniques, Performance Testing, Reliability Testing, Security Testing

I. INTRODUCTION

Software testing is a sequence of processes designed to check the code does what it was designed to do. As per the *ANSI/IEEE* 1059 standard [1, 2], Testing is defined as — A process of analyzing a software item to find the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.
The other related definition is this: [3] Testing is the mechanism of executing a program with the aim of finding errors. The concept of testing is as old as coding and is changing along with time, where lot of new testing techniques and tools evolved. Gelperin and Hetzel [4] proposed the concept of the testing process model based on associated publishing event.

Software Testing is a process that is performed to evaluate software quality and also to improve it (A project of the IEEE Computer Society Professional Practices Committee, 2004). Thus, the aim of testing is systematic and stepwise detection of different types of errors within a specified amount of time and also with in the allocated budget. Software testing is also a major component of software quality assurance (SQA), and a number of software organizations are put their budget up to 40% on testing.

The following are the four main objectives of testing :
Detection: different defects, errors, and deficiencies are detected. System limitations and various capabilities, quality of all components, the work products, and the overall system are calculated

Prevention: In this information to overcome or reduce the number of errors, to clarify system specifications and performance is provided. Different ways to avoid risks and to tackle problems in the future are identified.

Demonstration: It shows how the system can be used with various acceptable risks. It also illustrates functions with particular conditions and shows how products are ready for integration or use.

Improving quality: By performing effective testing on software, defects or errors can be minimized and hence quality of software is improved.

For life-critical and defence software like flight control and defence R&D, testing can be much expensive and sometimes out of budget, as risk analysis is also involved. Risk analysis - the probability by which a software would experience unexpected events, such as delays, schedule, outright cancellation and crossing budget and much more. So, a large number of test cases and test plans are made in testing which means that the behaviour of a program is inspected on a finite set of test cases i.e. test inputs, execution preconditions, and also expected outcomes for a particular objective, such as to follow a particular program path or to verify compliance with a specific requirement, for which valued inputs are created.

In experimental cases, the set of test cases is considered to be infinite, thus theoretically there are a large number of test cases even for the smallest and simplest program (Stacey, D. A., Software Testing Techniques). In such case, testing may take a lot of time even months and more to execute. So, how to choose a proper set of test cases? Practically, different techniques are used, and some of them are also related with the risk analysis, while others are related with test engineering expertise.

The basic purpose of software testing is verification, validation and error detection in order to find various errors and problems – and the aim of finding those problems is to get them fixed. Software testing is more than just error detection.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Software testing is done under controlled conditions for:

Verification: To check if system behaves as specified. It is to cross check and testing of items, which includes software, for conformance and consistency of software by evaluating the outputs against pre-defined requirements. The question here is, are we building the product right?

Validation: In this we cross check that what has been given in specifications by the user and what the user actually wanted. The question here is: Are we building the right system?

Error Detection: to detect errors. A number of tests should be done to make things go wrong to determine if what things should happen when they should not.

Sr. No.	Model name	Period	Function
1.	The Debugging Process Model	Before 1956	Testing and debugging was used interchangeably.
2.	The Demonstration Process Model	1957-78	Testing to make sure that the software satisfies its specification.
3.	The Destruction Process Model	1979-82	Testing to detect implementation faults.
4.	The Evaluation Process Model	1983-87	Testing to find error/bugs in requirements, design and implementation.
5.	The Prevention Oriented Period	After 1988	Testing to overcome faults in requirements, design and implementation.

Table 1: Various Testing Process Model.

Software testing is a destructive process of trying to find the errors. The primary objective of testing can be quality assurance, estimation of reliability, verification or validation.
Testing is a series of activities to check the correctness and completeness of the software.

The other objective of software testing is to get the quality of software system by systematically exercising the software in carefully controlled circumstances.

Software testing can be classified into the following based on the purpose of testing :

1. Correctness Testing
2. Performance Testing
3. Reliability Testing
4. Security Testing

II. SOFTWARE TESTING TECHNIQUES

Software testing is a process of discovering errors in a program and makes it a feasible task. It is useful process of executing program with the intent of finding bugs. The figure below shows some of the most important techniques of software testing which are classified by purpose. [4]

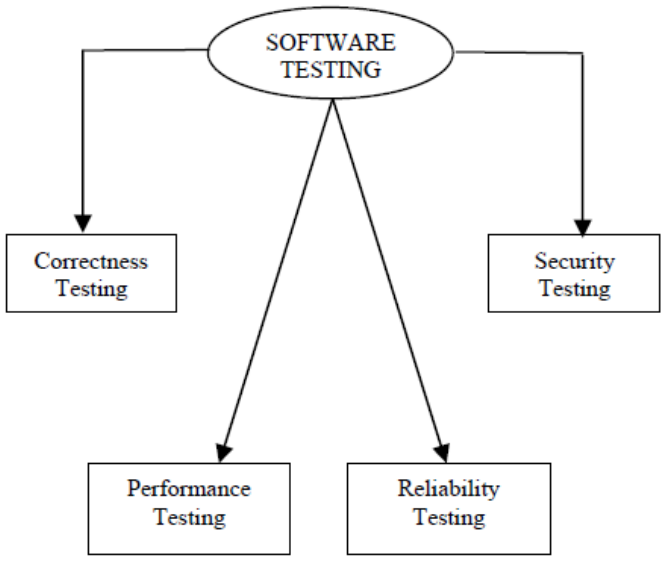

Fig. 1 Represent different software testing techniques which are classified by purpose

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

2.1 Correctness Testing

The most essential purpose of testing is correctness which is also the minimum requirement of software. Correctness testing means the correct behaviour of system from the wrong one for which it needs some type of Oracle. Either a white box point of view or black box point of view can be taken in testing software as a tester may or may not know the inside detail of the software module under test. For e.g. Data flow, Control flow etc. The purpose of black box, white box or grey box testing are not limited to correctness testing only. [4]

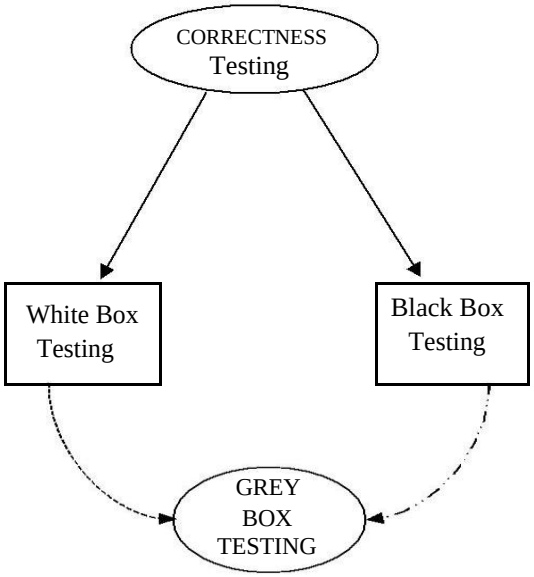

Fig. 2 The various forms of correctness testing

2.1.1 White Box Testing

White box testing is the process of giving the input to the system and checking how the system processes that input to generate the required output. It is necessary for a tester to have the full knowledge of the source code. White box testing can be applied for integration, unit and system levels of the software testing process. In white box testing one can be sure that all parts through the test objects are properly executed. [2][10]

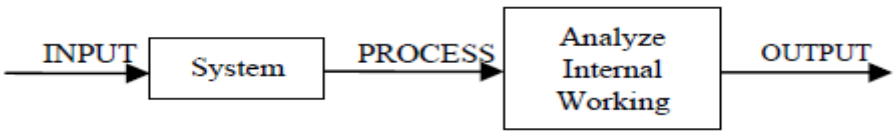

Fig. 3 Represent working process of White Box Testing

White box testing has several other names such as[5] : Logic Driven Testing, Design Based Testing, Open Box Testing , Transparent Box Testing , Clear Box Testing , Glass Box
Testing, Structural Testing

Some important types of white box testing techniques are:

1. Control Flow Testing
2. Branch Testing
3. Path Testing
4. Data flow Testing
5. Loop Testing

2.1.2 Black Box Testing

A black box is any device whose internal details and workings are not understood by or accessible to its user. It is testing of software by just giving inputs and checking the results, without any knowledge of the coding or internal structure in the program. The main aim is to test how well the system conforms to the specified requirements for the system. Black box means very little or no knowledge to the internal logical structure of the system. Thus, it only examines the fundamental aspect of the system. It just checks that all inputs are properly accepted and corresponding inputs are correctly produced.

The black box testing methods where the user involvement is not required are functional testing, stress testing, load testing, exploratory testing, ad-hoc testing, usability testing, smoke testing, recovery testing and volume testing, and the black box testing techniques where user involvement is required are user alpha testing, acceptance testing, and beta testing.

Other types of Black box testing methods includes equivalence partitioning, boundary value analysis, comparison testing graph based testing method, orthogonal array testing, specialized testing, fuzz testing, and traceability metrics. [2]

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

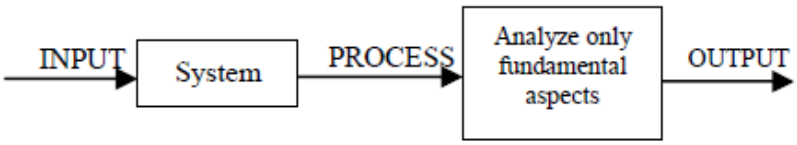

Fig. 4 Black Box Testing working model

2.1.3 Grey Box Testing

Grey box testing techniques is a mixture of both white box and black box. It is used to test a piece of module of software against its specifications often with some knowledge of its internal working as well. [2]

Grey box testing also includes reverse engineering to check, for instance, boundary values or error messages. Grey box testing is a process which involves testing software while already having some knowledge of its underline code or logic. The knowledge of internals of the software in grey box testing is more than black box testing, but less than clear box testing. [11]

2.2 Performance Testing

'Performance Testing' involve all the phases as the mainstream life cycle testing as an independent discipline which involve strategy such as plan, design, execution, analysis and reporting. This testing is done to evaluate the compliance of a system or component with given performance requirement. [2] Evaluation of a performance of any software includes resource usage, throughput and stimulus response time.

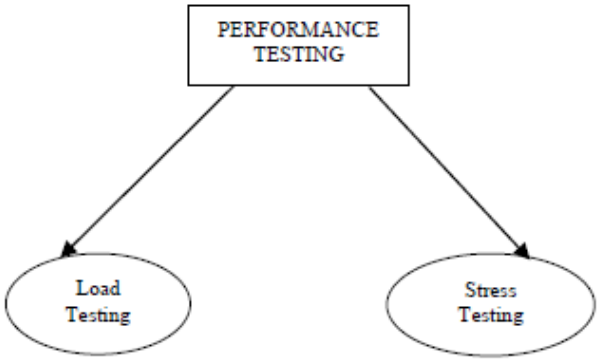

Fig. Performance testing types

Some of the main goals of performance testing are: [5]

- It is to measure response time of end to end transactions.
- The delay of network between client and server measurement.
- Monitoring of system resources which are under various loads.

Some of the common mistakes which happen during performance testing are: [5]

- Ignoring of errors in input.
- Analysis is too complex.
- Erroneous analysis.
- Phase 7 – Preparation of Report

There are two kinds of performance testing:

2.2.1 Load Testing

The main objective of the load testing is to determine whether the given system is able to
handle the expected number of users or not. This can be done

- Level of details is inappropriate.
- Ignore significant factors.
- Incorrect Performance matrix.
- Important parameter is overlooked.
- Approach is not systematic.

There are seven different phases in performance testing process: [5]

- Phase 1 – Requirement Study
- Phase 2 – Test plan
- Phase 3 – Test Design
- Phase 4 – Scripting
- Phase 5 – Test Execution
- Phase 6 – Test Analysis

by making the virtual user to behave as real user so that it will be easy to perform load testing. It is done only to check whether the system is performing well or not. The main objective of load testing is to check whether the system can perform well for specified user or not. Load testing increases the up time for critical web applications by helping us to spot the bottle necks in the system which is under large user stress.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

Load testing is done to check an application against heavy loads of inputs such as testing of website to find out at what point the website or applications fails or at what point its performance degrades. [2][5]

Two ways for implementing load testing are

1. Manual Testing: It is not a good choice for practical purpose as it is very iterative in nature and it involves [5]
   - Measure response time
   - Compare results
2. Automated Testing: The automated load testing tools provide more efficient and cost effective solutions when compared with Manual testing. In Automated testing, Tools test can easily be rerun desired number of times and decreases the chances of human error during testing.[5]

2.3 Reliability Testing

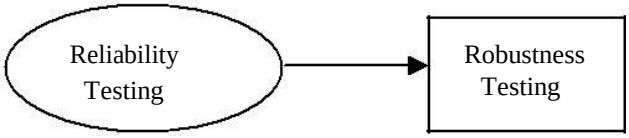

Fig. 6 Reliability testing

'Reliability Testing' is very important, as it discover all the failures of a system and removes them before the system is deployed. Reliability testing will be there in many aspects of software in which testing process is included; this testing process is an effective sampling method to measure software reliability. Estimation model is prepared in reliability testing which is used to analyze the data to estimate the present and predict future reliability of software. [4][2]

Depending on that estimation, the developers can decide whether to release the software or not and the end user will decide whether to adopt that software or not.

Based on reliability information, the risk of using software can also be assessed. The variance of reliability testing are Robustness and stress testing. By Robustness means, under stressful environment how a software component works. In Robustness testing, the machine crashes, abnormal terminations etc are looked into. Robustness testing is very portable and scalable. [4]

2.4 Security Testing

Security Testing: 'Security testing' enables only the authorized personnel can access the program and only the authorized personnel can access the functions available to their security level. Security testing of any system or (system under development) is all about finding the major loopholes and weaknesses of a system which can lead to unauthorized access by an authorized user. [1][2]

2.2.2 Stress Testing

Stress testing can be defined as performing random operational sequence, at larger volumes, at faster speeds and for longer periods of time, as a method to accelerate the rate of finding defects and verify the robustness of the product, or we can say stress testing is a testing, which is done to evaluate a system or component at or beyond the limits of its specified requirements to determine the load under which it fails and how. Stress testing also done to check the behaviour of the system as the number of users increases. The application or software is tested against heavy loads such as large number of inputs, large number of queries, etc. in stress testing. [2] [5]

Security testing will guide the tester to find and fix the problems. It ensures that the system will run for a long time without any major problems. It also ensures that the software used by any organization are very much secured from any unauthorized attack. The Security testing is beneficial for the organization in all aspects. [1][2]

Five major concepts which are covered by security testing are

Confidentiality: By security testing, we will ensure the confidentiality of the system i.e. The information is not disclosed to the unknown party other than intended recipient.
Integrity: By security testing, we will maintain the integrity of the system by allowing the receiver to determine that the information which he is getting is correct.
Authentication: Security testing maintains the authentications of the system and WPA, WPA2, WEP are several forms of authentication.
Availability: Information is always kept available for the authorized personnel whenever they needed and assures that information services will be ready for use whenever expected.
Authorization: Security testing ensures that only the authorized user can access the information or particular service. Access control is an example of authorization.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

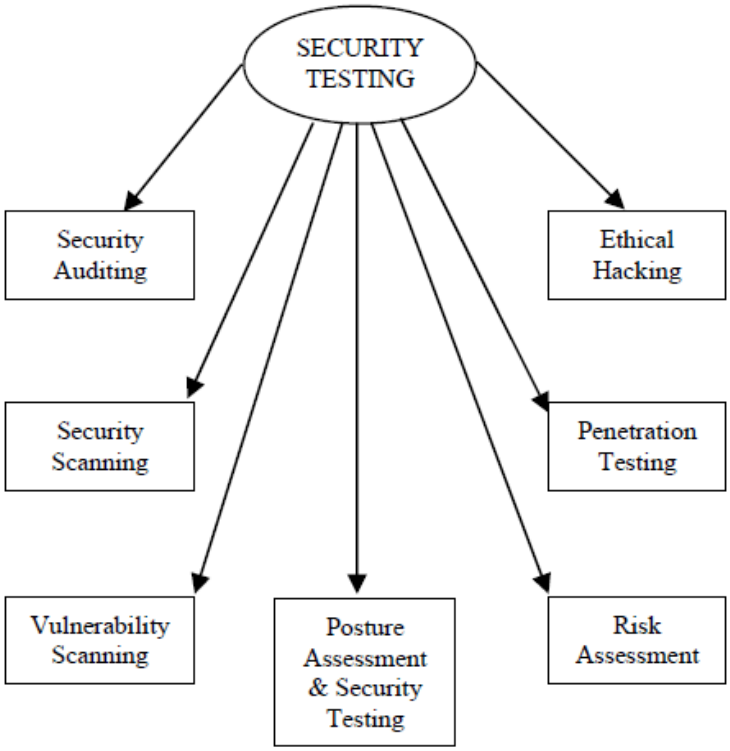

Fig. 7 Represent various type of security testing

Different types of security testing in any organization are as follows: [3]

1. Security Auditing and Scanning: Security Auditing means direct inspection of the operating system and of the system on which it is developed. In Security Scanning the auditor scans the operating system and then tries to find out the weaknesses in the operating and network.
2. Vulnerability Scanning: This which involves the scanning of the program for all known vulnerability.
3. Risk Assessment: Risk Assessment is a process in which the auditors analyze the risk that may occur with any system and all the probability of loss which occurs because of that risk. It is analyzed through interviews, discussions, etc.
4. Posture Assessment and Security Testing: Posture Assessment and Security Testing helps the organization security levels by combining the features of security scanning, risk assessment and ethical hacking.
5. Penetration Testing: Penetration Testing is an effective way to find out the potential loopholes in system and it is done by a tester which forcibly enters into the application under test. A tester will enter into the system with the help of combination of loopholes that the application has kept open unknowingly.
6. Ethical Hacking: Ethical Hacking involves large no. of penetration test on a system under test. To stop the forced entry of any external elements into a system which is under security testing.

III. CONCLUSION

Software testing is an important aspect to build the reliable and robust software product. But it is really not possible to find out all the errors in the program. So, the basic query is, which strategy we would adopt to test. In this paper, it has been elaborately described some of the most prominent and commonly used strategies of software testing which are categorized by purpose of testing and they are classified into [5]

1. Correctness testing, which is used to test the right behavior of the system and it is further divided into black box, white box and grey box testing techniques (combines the features of black box and white box testing).
2. Performance testing, which is an independent discipline and involves all the phases as the main stream testing life cycle i.e. strategy, plan, design, execution, analysis and reporting. Performance testing is further divided into load testing and stress testing.
3. Reliability testing, which discovers all the failure of the system and removes them before the system deployed.
4. Security testing makes sure that only the authorized personnel can access the system and is further divided into Security Auditing and Scanning, Vulnerability Scanning, Risk Assessment, Posture Assessment and Security Testing, Penetration Testing and Ethical Hacking.

A lot of scope in this testing area is still open for Research to find an appropriate testing technique for a given software.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 6 Issue 04, April-2017

REFERENCES:

[1] Software testing-Brief introduction to security testing by Nilesh Parekh published on 14-07-2006 available at http://www.buzzle.com/editorial/7-14-2006-102344.asp
[2] Software testing glossary available at http://www.aptest.com/glossary.html#performance testing
[3] Open source security testing methodology manual of PETE HERZOG and the institute for security and open methodology-ISECOM.
[4] Software testing by Jiantao Pan available at http://www.ece.cmu.edu/~roopman/des-899/sw_testing/
[5] Software Testing by Cognizant Technology Solution.
[6] Introduction to software testing available at http://www.onestoptetsing.com/introduction/
[7] Software testing techniques available at http://pesona.mmu.edu.my/~wruslan/SE3/Readings/ GB1/pdf/ch14-GB1
[8] Paper by Lu Luo available at http://www.cs.cmu.edu/~luluo/Courses/17939Report .pdf
[9] Security testing-wikipedia the free encyclopedia available at http://en.wikipedia.org/wiki/security-tetsing.
[10] White box testing from wikipedia, the free encyclopedia.
[11] Software testing for wikipedia available at http://en.wikipedia.org/wiki/grey_box_testing#grey_ box_testing