International Journal of Trend in Research and Development, Volume 2(4), ISSN 2394-9333
www.ijtrd.com

A Review on Software Testing

Swetha.C, Gomathi.S, Ranjani.R,
Department of IT,
Sri Krishna Arts and Science College,
Coimbatore, India.

Abstract: This document is based on the software testing. Importance of testing, process of testing in steps, types of testing, methods of testing, levels of testing. This document describes about the comparison of three different boxes of testing.

Keywords: testing, black, white, grey box testing, functional and non-functional testing)

I. INTRODUCTION

Software testing is a process of executing a program or application with the intent of finding the software bugs [1]. It can also be stated as the process of validating and verifying that a software program or application or product: Meets the business and technical requirements that guided its design and development Works as expected. Can be implemented with the same characteristic.

Software testing is a process used to identify the correctness, completeness, and quality of developed computer software. It includes a set of activities conducted with the intent of finding errors in software so that it could be corrected before the product is released to the end users.

In simple words, software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free.

According to ANSI/IEEE 1059 standard, Testing can be defined as "A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item".

II. What is Importance of Testing?

- Testing should be introduce in the early stage of the SDLC, The cost of fixing the bug is larger if testing is not done in early stage & bugs found in later stages[1].
- In the today's competitive market only the quality product stays longtime firmly, so to make sure the produce the good quality product the testing of application is key factor in SDLC.
- As it not possible makes it software application is defect free but testing will be necessary.
- Most important thing of testing is the development environment is different than the Testing environment and the testing done on testing environment is similar to the Production environment.

A) The Fundamental Test Process in to the Following Basic Steps:

1) Planning and Control
2) Analysis and Design
3) Implementation and Execution
4) Evaluating exit criteria and Reporting
5) Test Closure activities

B) Difference between Testing and Debugging

a) Testing: It involves the identification of bug/error/defect in the software without correcting it [2]. Normally professionals with a Quality Assurance background are involved in the identification of bugs. Testing is performed in the testing phase.

b) Debugging: It involves identifying, isolating and fixing the problems/bug. Developers who code the software conduct debugging upon encountering an error in the code. Debugging is the part of White box or Unit Testing. Debugging can be performed in the development phase while conducting Unit Testing or in phases while fixing the reported bugs.

III. Testing Methods

- Black Box Testing
- White Box Testing
- Grey Box Testing

A) Black Box Testing

Black box testing – Internal system design is not considered in this type of testing [3]. Tests are based on requirements and functionality.

Table 1.1: Advantages and Disadvantages of Black Box Testing.

Advantages	Disadvantages
Well suited and efficient for large code segments.	Limited coverage, since only a selected number of test scenarios is actually performed.
Code access is not required.	Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
Clearly separates user's perspective from the developer's perspective through visibly defined roles.	Blind coverage, since the tester cannot target specific code segments or error-prone areas.
Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems.	The test cases are difficult to design.

B) White Box Testing

White box testing – This testing is based on knowledge of the internal logic of an application's code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Table 1.2: Advantages and Disadvantages Of White-Box Testing

Advantages	Disadvantages
As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.	Due to the fact that a skilled tester is needed to perform white-box testing, the costs are increased.
It helps in optimizing the code.	Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems, as many paths will go untested.
Extra lines of code can be removed which can bring in hidden defects.	It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools.
Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.	

C) Grey Box Testing

Grey box testing is the testing of software application using effective combination of both White box testing & Black box testing method. This is nice & powerful idea to test the application.[3] Grey Box Testing approach is the testing approach used when some knowledge of internal structure but not in detailed. The name is comes because the application for tester is like a grey box like a transparent box and tester see inside it but not fully transparent & can see partially in it.

Table 1.3: Advantages and Disadvantages Of Grey-Box Testing

Advantages	Disadvantages
Offers combined benefits of black-box and white-box testing wherever possible.	Since the access to source code is not available, the ability to go over the code and test coverage is limited.
Grey box testers don't rely on the source code; instead they rely on interface definition and functional specifications.	The tests can be redundant if the software designer has already run a test case.
Based on the limited information available, a grey-box tester can design excellent test scenarios especially around communication protocols and data type handling.	Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

The test is done from the point of view of the user and not the designer.	

D) Comparison of Testing

Table 2: Compares The Three Different Boxes (Black, White, Grey)

Black-Box Testing	Grey-Box Testing	White-Box Testing
The internal workings of an application need not be known.	The tester has limited knowledge of the internal workings of the application.	Tester has full knowledge of the internal workings of the application.
Also known as closed-box testing, data-driven testing, or functional testing.	Also known as translucent testing, as the tester has limited knowledge of the insides of the application.	Also known as clear-box testing, structural testing, or code-based testing.
Performed by end-users and also by testers and developers.	Performed by end-users and also by testers and developers.	Normally done by testers and developers.
Testing is based on external expectations - Internal behavior of the application is unknown.	Testing is done on the basis of high-level database diagrams and data flow diagrams.	Internal workings are fully known and the tester can design test data accordingly.
It is exhaustive and the least time-consuming.	Partly time-consuming and exhaustive.	The most exhaustive and time-consuming type of testing.
Not suited for algorithm testing.	Not suited for algorithm testing.	Suited for algorithm testing.
This can only be done by trial-and-error method.	Data domains and internal boundaries can be tested, if known.	Data domains and internal boundaries can be better tested.

IV. LEVELS OF TESTING

There are different levels during the process of testing. Levels of testing include different methodologies that can be used while conducting software testing. The main levels of software testing are:

- Functional Testing
- Non-functional Testing

A) Functional Testing

This is a type of black-box testing that is based on the specifications of the software that is to be tested[3]. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. Functional testing of software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

B) Non Functional Testing

This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing software from the requirements which are nonfunctional in nature but important such as performance, security, user interface, etc.

C) Distinguish between Functional and Non Functional Testing

Table 3: Difference of Functional and Non-Functional Testing

Functional Testing	Non- Functional Testing
In functional Testing tester tests how well the system performs.	In Non-Functional Testing tester tests how well the system responds.
Functional Testing is based on client requirements.	Non- Functional Testing is based on client expectations.
Functional Testing means Testing the application against business requirements.	Non- Functional Testing means Testing the application against clients and performance requirements.
It is a part of System Testing.	It is also a part of System Testing
Functional Testing Validating the behavior of application.	Non- Functional Testing Validating the performance of application.

This Testing covers Unit Testing, Integration Testing, Smoke Testing, Sanity Testing, Regression Testing and so on.	This Testing covers Load/Performance Testing, Stress/Volume Testing, Security Testing, Installation Testing and so on.
It is always concentrating on customer requirements.	It is always concentrating on customer expectations.
Functional Testing means how is your system is doing.	Non- Functional Testing means how well your system is doing example usability, performance and stress testing.

CONCLUSION

 Testing is a critical part of the development of any system. Testing can be carried out at a number of levels and is planned as an integral part of the development process[2]. There is a wide range of approaches to test case generation and evolution of the adequacy of a test suite. Test needs to be managed effectively if it is to be efficient.

REFERENCES

[1]www.tutorialspoint.com
[2]www.testingexcellence.com
[3]www.softwaretestingclass.com