



# Comparative Study on Software Testing Techniques

**Niranjanamurthy M<sup>1</sup>, Nitesh S N<sup>2</sup>, Nagesh S N<sup>3</sup>, Balaji Sriraman<sup>4</sup>**

<sup>1</sup>Assistant Professor, Department of MCA, MSRIT, Bangalore-54, INDIA

<sup>2</sup>Student, Department of MCA, MSRIT, Bangalore-54, INDIA

<sup>3</sup>Student, Department of MCA, MSRIT, Bangalore-54, INDIA

<sup>4</sup>Student, Department of MCA, MSRIT, Bangalore-54, INDIA

<sup>1</sup>niruhds@gmail.com; <sup>2</sup>nitesh.nadiger@gmail.com; <sup>3</sup>nagesh.nadiger1@gmail.com; <sup>4</sup>balazee1710@gmail.com

---

*Abstract- Software testing is the process of evaluation of a software item to detect differences between given input and the output which is expected. Also to assess the feature of a software item, testing assess the quality of the product. Software testing is a process which is to be done during the development process. In other words software testing is a validation and verification process. Executing a system in order to establish gaps, errors or missing requirements in contrary to the actual desire or requirements is testing. White Box (or glass box) testing is the process of giving input to the system and checking how the system processes input to generate the output. Black Box testing is the process of giving input to the system and checking the output without bothering how the output is generated. White Box and Glass Box Testing combined to produce the Gray Box Testing. In this, the tester has little knowledge about the internal working of the software. So the testing of the output as well as the process carried out to generate the output. Software testing is a method of assessing the functionality of a software program. And Software testing has been classified into two main categories, dynamic testing and static testing. In this survey paper we discussed what is Testing, Terminology used in Testing, Levels in Testing and Comparison of Manual and Automation Testing, Comparison of Selenium and QTP tool and Comparison of Black, White and Gray box Testing.*

*Keywords— Software Testing, White Box, Black Box and Gray Box Testing Terminology used in Testing, Levels of Testing, Comparison of software testing Techniques*

---

## I. INTRODUCTION

Software Testing is the process of identifying the security, correctness, completeness and quality of the developed computer software. Testing is a process of technical searching/investigation, performed on behalf of stakeholder, which is to reveal quality related information about the product with respect to the circumstances in which it is meant to operate. This includes and is not limited to the process of executing a program or application with the purpose of finding the errors. Quality is a value to some person and not absolute. Testing can never completely begin the correctness of arbitrary computer software. Testing provides comparison between the state and behaviour of the product against a specification.

**Black Box Testing** – Internal system design was not considered in this testing. Tests are based on requirements and functionality.

**White Box Testing** – Testing is based on knowledge of the inner logic of an application's code also known as Glass box Testing. Internal working of a code and software should be known. Tests are built on coverage of branches, paths, conditions and code statements.

**Grey Box Testing** - Grey Box testing is a technique of testing the application with the finite amount of knowledge on the internal working of an application.

In software testing, the term "more you know, the better it carries" a lot of weight when testing an application.

In today's fast growing world of the Internet customer wish to get products to be implemented and updated faster than other contenders. Customer requires more for software releases with new features to be implemented as quick as possible, but they don't like software's which is defected. As next version of the product will be released in coming days, & gets couple of days of testing before it is transported. Hence due to the short time frame or constant releases, there will be more bugs assembled into the products and which will be fixed in the next release, correct? Yes In a definite manner, this model has its own problems. Releasing such applications/software with so much of bugs, it may affect the user experience which reflects in the quality impression of the company brand. They will be remembered for the delivered bad quality product. So this importance of testing plays vital role in SDLC.

Here are the advantages of software testing in the Software Development Life Cycle: Testing should be introduced in the initial stage of the SDLC. The cost of fixing the bug is quite large if testing is not done in initial stage & bugs found in later stages.

In today's competitive market, only the quality products stays for long duration, thus to produce the good quality product, product is made sure that the testing of application is the key factor in SDLC.

As it is not possible to make the software application fault free but testing will be necessary.

Most important part of testing is the development. The development environment differs from the Testing environment and the testing performed on testing environment will be similar to the Production environment. [10]

## II. AIM OF THE STUDY

- To know what is Software Testing.
- To Understand the Levels of Software Testing
- To know Terminology used in Testing.
- Comparison of Manual and Automation Testing
- Comparison of Selenium and QTP tool
- Comparison of Black box, White box and Gray box Testing

## III. RELATED WORK

Software testing is the deciding part of software development in delivering a quality software product that is fault free and the process of automating software testing is important to its success. Testing is necessary since software reliability is described using testing and roughly fifty percent of the budget in software development will be spent on testing of software projects. [1]

A software development process, also known as a software development life cycle (SDLC), is a structure imposed on the development of software products. Software/application testing refers to process of checking the software with intention to find out error in it. Software

testing is a technique aimed at evaluating an attribute or capability of a program or product and determining that it meets its quality. Software testing is also used in testing the software for other software quality factors such as integrity, usability, security, capability, efficiency, portability, reliability, maintainability, compatibility etc. [2]

Software testing identify the defects, flaws or errors in the application code that must be predetermined. We can also define software testing as a process of accessing the functionality and correctness of a software through analysis. The main goal of testing can be assuring quality, reliability, estimation, validation and verification. Software testing is the fundamental component of software quality assurance and represents a review of design, coding and its specification. The main objective of software testing is to affirm the quality of software system by systematically testing the software in carefully controlled circumstances, and another objective is to identify the completeness and correctness of the software, and finally it uncovers undiscovered errors. [3]

In development of software process, testing plays a huge role, on which the reliability and quality of the product delivered strictly depends on. Testing is not constrained to the detection of “bugs” in the software, but also increases confidence in the proper functioning and assists with the evaluation of functional and non-functional properties. Testing related activities enclose the entire development process and may take a large part of the effort required for producing software. In this paper we give in depth overview of the software testing. From its definition to the organization, test levels to the test techniques, test execution to the analysis of effectiveness in the test cases. Significance of breadth is more compared to depth. Due to the enormous quantity of the topic, in the attempt will be all-embracing. For each of the subject covered only a brief description and useful references will be provided for further reading.[4]

Test Coverage is a major indicator of software quality and it is a essential part of software maintenance. It helps in assessing the success of testing by providing data on many different coverage items. Although many research efforts have been put on getting the coverage information by either code based testing or requirement based testing. Nothing much has been paid to measure and analyse the coverage by covering maximum number of coverage items. This paper provides knowledge on the current test coverage researches conducted by other researchers in software testing. With the current proceedings the gaps and uncovered measurement of Test Coverage can be explored to the more distant. [5]

#### **Terminology:**

**Mistake** – An incorrect result produced by a human action.

**Fault [or Defect]** – An inaccurate step, process, or data definition in a program.

**Failure** – The lack of ability of a system or component to perform its required function within the specified performance requirement.

**Bug**- “A computer bug is an flaw, error, failure, mistake or fault in a computer program which prevents from working correctly or produces an incorrect or different result. Bugs appears from mistakes and error, made by humans, in either in a program’s source code or its design.”

**Error** – The difference between a computed, observed, or Measured value or condition and the true, specified, or theoretically correct value or condition.

**Specification** – A document that specifies in a Verifiable manner, precise, complete, requirements, design, behaviour, or other properties of a system or component, and often the process of determining whether these provisions have been Satisfied. We observe errors, which can often be associated with failures. But the ultimate source of the fault is often very hard to find.[6]

#### IV. SOFTWARE TESTING LEVELS

Testing levels are basically to identify missing areas and prevent overlap and repetition between the development life cycle stages. In software development life cycle models there are defined phases like requirement gathering and coding, design, analysis or implementation, deployment and testing. Each stage will be run through testing. Hence there are various stages of testing.

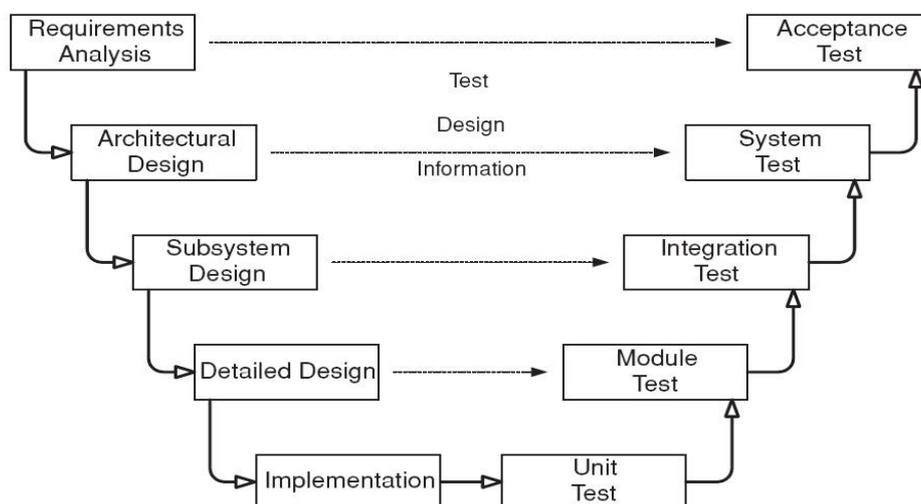


Fig.: levels of testing

The various levels of testing are:

1. Unit testing: It is basically done by the developers to make sure that their code is working fine and meet the user specifications. They test their code snippets which they have written like functions, classes, interfaces and procedures.

|                   |  |
|-------------------|--|
| <b>Objectives</b> | To test the function of a program or unit of code such as a program or module, To test internal logic, To verify internal design, To test path & conditions coverage, To test exception conditions & error handling, To test the function of a program or unit of code such as a program or module, To test internal logic, To verify internal design, To test path & conditions coverage, To test exception conditions & error handling |
| <b>When</b>       | After modules are coded  |
| <b>Who</b>        | Developer  |
| <b>Input</b>      | Internal Application Design, Master Test Plan, Unit Test Plan  |
| <b>Output</b>     | Unit Test Report   |
| <b>Methods</b>    | White Box testing techniques, Test Coverage techniques   |
| <b>Tools</b>      | Debug, Re-structure, Code Analyzers, Path/statement coverage tools   |

2. Component testing: It is also called as module testing. The main difference between unit testing and component testing is in unit testing the developers test their piece of code but in component testing the entire component is tested. For example, in a student record application, there are two modules one which will save the records of the students and other module is to upload the results of the students. Both the modules are developed individually and when they are tested one by one then we call this as a component or module testing.

3. Integration testing: Integration testing is done when two modules are combined, in order to test the behavior and functionality of both the modules after combining together. Following are the types of integration testing:

- o Big bang integration testing
- o Top down
- o Bottom up
- o Functional incremental

|                   |  |
|-------------------|--|
| <b>Objectives</b> | To technically verify proper interfacing between modules, and within sub-systems |
| <b>When</b>       | After modules are unit tested  |
| <b>Who</b>        | Developers   |
| <b>Input</b>      | Internal & External Application Design ,Master Test Plan ,Integration Test Plan  |
| <b>Output</b>     | Integration Test report  |
| <b>Methods</b>    | White and Black Box techniques ,Problem / Configuration Management               |
| <b>Tools</b>      | Debug ,Re-structure ,Code Analyzers  |

4. Component integration testing: In the example above when both the modules or components are integrated then the testing done is called as Component integration testing. This testing is basically done to make sure that the code should not break after integrating the two modules.

5. System integration testing: System integration testing (SIT) is a testing where testers basically test that in the same environment all the related systems should maintain data integrity and can operate in coordination with other systems.

6. System testing: In system testing the testers basically test the compatibility of the application with the system.

|                   |   |
|-------------------|---|
| <b>Objectives</b> | To verify that the system components perform control functions,To perform inter-system test,To demonstrate that the system performs both functionally and operationally as specified,To perform appropriate types of tests relating to Transaction Flow, Installation, Reliability, Regression etc. |
| <b>When</b>       | After Integration Testing   |
| <b>Who</b>        | Development Team and Users  |
| <b>Input</b>      | Detailed Requirements & External Application Design,Master Test Plan,System Test Plan   |
| <b>Output</b>     | System Test Report  |
| <b>Methods</b>    | Problem / Configuration Management  |
| <b>Tools</b>      | Recommended set of tools  |

7. Acceptance testing: Acceptance testing is primarily performed to ensure that the requirements of the specification are met.

|                   |   |
|-------------------|---|
| <b>Objectives</b> | To verify that the system meets the user requirements                             |
| <b>When</b>       | After System Testing  |
| <b>Who</b>        | User / End User   |
| <b>Input</b>      | Business Needs & Detailed Requirements,Master Test Plan,User Acceptance Test Plan |
| <b>Output</b>     | User Acceptance Test report   |
| <b>Methods</b>    | Black Box techniques,Problem / Configuration Management                           |
| <b>Tools</b>      | Compare, Keystroke capture & Playback, Regression testing                         |

8. Alpha testing: Alpha testing is performed at the end of development process in the developer site.

9. Beta testing: Beta testing is done just before the product launch in the customer site.[7]

## V. COMPARISON OF SOFTWARE TESTING TECHNIQUES

### A. Comparison of Black Box Testing, Grey Box Testing and White Box Testing,

| S. No. | Black Box Testing  | Grey Box Testing   | White Box Testing                                   |
|--------|--|--|---|
| 1.     | Analyses fundamental aspects only i.e. no proved edge of internal working              | Partial knowledge of internal working  | Full knowledge of internal working                  |
| 2.     | Granularity is low   | Granularity is medium  | Granularity is high                                 |
| 3.     | Performed by end users and also by tester and developers (user acceptance testing)     | Performed by end users and also by tester and developers (user acceptance testing)   | It is performed by developers and testers           |
| 4.     | Testing is based on external exceptions – internal behaviour of the program is ignored | Test design is based on high level database diagrams, data flow diagrams, internal states, knowledge of algorithm and architecture | Internal are fully known                            |
| 5.     | It is least exhaustive and time consuming  | It is somewhere in between   | Potentially most exhaustive and time consuming      |
| 6.     | It can test only by trial and error method   | Data domains and internal boundaries can be tested and over flow, if known   | Test better: data domains and internal boundaries   |
| 7.     | Not suited for algorithm testing   | Not suited for algorithm testing   | It is suited for algorithm testing (suited for all) |

[3]

### B. Comparison between Automation Testing and Manual testing

| Automation Testing   | Manual Testing   |
|--|--|
| Automation testing perform the same operation each time  | Manual testing is not reliable. Using this method test execution is not accurate all the time.   |
| Automation testing is very much helpful regressions in testing where code changes frequently.                  | To execute the test cases first time using manual testing will be very much useful. But it is not sure that it will catch the regression defects under frequently changing requirements. |
| Automation testing will be useful to execute the set of test cases frequently.                                 | Manual testing will be useful when the test case only needs to run once or twice.  |
| After making Automation test suites, fewer testers required to execute the test cases.                         | To execute the test cases every time tester requires the same amount of time.  |
| Automation testing can also be done on different machine with different OS platform combination, concurrently. | Using manual testing, testing on different machine with different OS platform combination is not possible, concurrently. To execute such task different testers are required.            |
| Using Automation testing, testers can test complicated application too.  | It does not involve in programming task to fetch hidden information.   |
| Automation runs test cases significantly faster than human resources.  | Manual testing is slower than automation. Running tests manually can be very time consuming.   |
| Some time it is not helpful in UI testing  | It is very much helpful in UI testing  |

|   |   |
|---|---|
| Automation testing is very useful for automating the Build Verification Testing (BVT) & it is not mundane and tiresome. | To execute the Build Verification Testing (BVT) is very mundane and tiresome in manual testing. |
| Initial cost of automation testing is more than manual testing but useful always.                                       | Manual testing requires less cost than automation.  |

[8]

### C. Comparison between Selenium and QTP Tool

| FEATURES                                       | SELENIUM  | QTP   |
|--|---|---|
| Cost   | Open source & Portable  | Licensed and very expensive; Ten user license costs approx. 60L         |
| Application support                            | Web Applications only   | Client server applications only (like built in TCL/TK and PowerBuilder) |
| Support for web browsers                       | Supports IE, Firefox, Safari and Opera  | Supports IE & Firefox only  |
| Object Oriented Language support & Scalability | Supports Java, .Net, Perl, PHP, Python, and Ruby  | Supports VB script only   |
| Support for operating system/platforms         | Supports Windows PC/MAC/UNIX Platforms  | Supports Windows Platform only  |
| Support for Test management tool integration   | Not available. Need to track separately   | TD/QC will be easily integrated   |
| Test Development Environment                   | We can use wide range of IDEs like Eclipse, Netbeans, Visual Studio etc   | Need Separate environment   |
| UI object management & Storage                 | Managed using UI-Element user extension and properties A set of dynamically loaded libraries that is stored in the Java archive file. | Built-in object repository and easy handling                            |
| Support for Dialog Boxes                       | Supports partially  | Supports all kinds of dialog boxes                                      |
| Support for File upload (system)               | Not available   | Supports all kinds of File upload                                       |

[9]

## VI. CONCLUSIONS

Software Testing is the process of identifying the correctness, security, completeness and quality of the developed computer software. Testing is a process of non-theoretical enquiry, performed on behalf of stakeholder, which is to reveal quality related information about the product with respect to the circumstances in which it is meant to operate. This includes and is not limited to the process of executing a program or application with the purpose of finding the errors. Quality is a value to some person and not absolute. Testing can never completely

begin the correctness of arbitrary computer software. Testing provides comparison between the state and behavior of the product against a specification.

Testing of software can be done both in Automation and Manual testing method. It totally depends on the project requirement, budget related with the project and the testing method which will be benefited to the project. The article gives fundamental information about Manual Testing and Automation Testing.

Testing the software involves the validating and verifying of a software program. The faults are to be identified in order to fix them.

Thus the main objective of software testing is to continue and distribute a quality product to the client. Every software is awaited to meet certain requirements. So a software is to be checked whether it fulfills the needs when it is developed. Banking software is completely dissimilar from a software required in a shop. The necessities of both the software are not same. Hence the potential is to be checked. The main aim of software testing is to know the faults of the software before the user realize them. A good tester is one who makes the software unsuccessful. He will always be in a mental attitude to destroy the software.

#### ACKNOWLEDGEMENT

I thank Dr. T. V. Suresh Kumar, Prof. and Head, Dept. of MCA, MSRIT, Bangalore-54. for his continuous support and encouragement for completing this research paper and also thanks to MSRIT management.

I thank Dr. Jagannatha, Associate Professor. of Dept. of MCA, MSRIT, Bangalore-54, for his valuable guidance and support for completing this paper.

#### REFERENCES

- [1] Abha Jain, Manish Jain, Sunil Dhankar- "A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing". International Journal of Engineering, Management & Sciences (IJEMS)ISSN-2348 –3733, Volume-1, Issue-1, January 2014
- [2] Harpreet Kaur, Dr.Gagan Gupta."Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete". IJERA ISSN : 2248-9622, Vol. 3, Issue 5, pp.1739-1743 Sep-Oct 2013
- [3] Mohd. Ehmer Khan, Farmeena Khan-"A Comparative Study of White Box, Black Box and Grey Box Testing Techniques". (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No.6, 2012
- [4] H.S. Samra-"A Study on Software Testing"- International Journal of Advanced Research in Computer Science and Software Engineering". IJARCSSE ISSN: 2277 128X Volume 3, Issue 1, January 2013
- [5] Muhammad Shahid, Suhaimi Ibrahim and Mohd Naz'ri Mahrin-"A Study on Test Coverage in Software Testing"-International Conference on Telecommunication Technology and Applications-IACSIT Press, Singapore. Proc .of CSIT vol.5 2011
- [6] Shivkumar Hasmukhrai Trivedi-"Software Testing Techniques"- International Journal of Advanced Research in Computer Science and Software Engineering ISSN: 2277 128X, Volume 2, Issue 10, October 2012
- [7] <http://istqbexamcertification.com/what-are-software-testing-levels/>
- [8] <http://www.softwaretestingclass.com/automation-testing-vs-manual-testing/>
- [9] <http://automationinqtp.blogspot.in/2013/01/qtp-vs-selenium.html>
- [10] <http://www.softwaretestingclass.com/importance-of-testing/>
- [11] <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [12] <http://www.measuringusability.com/blog/ux-methods.php>
- [13] [http://en.wikipedia.org/wiki/Comparison\\_of\\_usability\\_evaluation\\_methods](http://en.wikipedia.org/wiki/Comparison_of_usability_evaluation_methods)