

# Importance of Software Testing in the Process of Software Development

Azeem Uddin<sup>1</sup> Abhineet Anand<sup>2</sup>

<sup>1,2</sup>Galgotias University, Greater Noida, Uttar Pradesh, India

**Abstract**— Software Testing is the process to check whether the software is defect-free or not. It is the process of verification and validation of software service or application by checking whether it is meeting the user requirements and what all is implemented as per the characteristics. Software testing plays a vital role in the process of developing a high quality software. Testing is necessary because we all make mistakes. Some of those mistakes are unimportant but some of them are expensive and dangerous. Therefore, there is a need to check everything that we produce. Hence, this paper presents an introduction to Software Testing and its importance with strong exemplary illustrations. It not only provides a vivid account of the need, objectives and principles of testing but also summarises traits of an excellent Software tester. At last, this paper includes a comparative study of present scenario of software testing with regard to testing in early days. Moreover, this paper focuses on “Why to test” not How to test, so that the need of carrying out Software testing can be crystal clear in a more effective manner.

**Key words:** Software Testing, Need of Software Testing, Software Testing Objectives, Software Testing Principles, Software Testing Importance, A Good Software Tester

## I. INTRODUCTION

Testing is not a PIT, it is a LADDER...!

In simple words, Testing is finding out how well something works. Relating it to the software, Software Testing is the process to check the correctness, completeness and accuracy of developed computer software. In technical terms, we can say, Testing is the combination of Verification and Validation.

This can be well-explained with the help of the simplest example. Suppose, you have to buy a pen from the stationary shop. You will ask the shopkeeper about the same, he will give you a new pen. Now, there can be two possibilities, that particular pen can be working and at the same time, it can be defective as well. So, in order to check it, you will ask the shopkeeper for a rough page, you will write something and you will test it whether it is defective or not. This is nothing but a straightforward example of Testing. Developing a software is not a cakewalk. Everyone wants to get things that work well. They don't want to receive a defective piece and that's quite obvious as well. Applying this concept to customer service, customers also expect an error-free software from the development team that is made according to their requirements. Hence, from here, Testing plays a very crucial role in the development of a high quality software. Therefore, by summing up all these ideas, Software Testing may be defined as the process of exercising or evaluating a system by manual or automatic means to verify that it satisfies specified requirements or to identify differences between actual and expected results. A successful test is one that does detect a fault. Testability is one of the most important Software Quality Attributes that influence the Software product.

So, this paper is all about understanding the need of Software Testing so that an error-free, reliable, flexible and efficient Software can be delivered to customers.

## II. NEED OF SOFTWARE TESTING

In order to understand the need of Software Testing in a better way, we should go back to 1960s and 1970s when software crisis occurred and there were many difficulties in the development of software systems. Some examples are stated below-

### A. The Northeast Blackout In 2003

It has been major power system failures in the history of North America that involves 100 power plants, suddenly, the power plant was failed. 50 million customers faced this problem and there was a financial loss of \$6 billion.

### B. Arian 5-Space Rocket

In 1996, Arian 5- space rocket, developed at cost of \$7000 million over a period of 10 years was destroyed within less than one minute after its launch as there were software bugs in Rocket Guidance system.

### C. US Banks Credit

In 1996, US banks credited \$924 lakhs in the accounts of nearly 800 customers. This problem was due to programming bugs in the Banking System.

If we try to find out the causes of above failures, then, there can be many factors responsible for the same. They can be

- Hardware oriented World
- Inexperienced developer
- User changing requirements
- Rapid increase in software cost

But, the main aspect that is totally responsible for the failures in big projects is LACK OF PROPER TESTING because at that time also, developers were trying their level best to develop a robust software according to the available resources, skills and technology, but they were lagging in Testing process that resulted in severe loss of time and money due to the absence of appropriate testing methodologies, tools and knowledge.

Therefore, to know how to test the system is an issue, more importantly, one should know why to test the system that requires knowledge of goals, objectives and principles of testing.

## III. OBJECTIVES OF SOFTWARE TESTING

The major role of Software Testing is that there should be no discrepancy in software development process. The goals and objectives of testing are numerous, which when achieved help developers build a defectless and satisfactory software. Some of the Objectives of Software Testing are summarised below-

#### A. To Find and Prevent Defects

The foremost task of a tester is to find defects in the software and report them to developer so that they can be rectified. A tester must form best set of test cases so that maximum defects can be arised. After all, Testing shows presence of defects.

#### B. Satisfies the SRS & BRS

Another objective of testing is to check whether the developed software satisfies the Software Requirement Specification and Business Requirement Specification or not because until or unless the software is satisfying user requirements, it is of no use to the customer inspite of using best programming skills, designing and tools.

#### C. Writing High Quality Test Cases

A test case is a set of conditions under which a tester will determine whether an application under test satisfies requirements or works correctly. The process of developing test cases can also help problems in the requirement or design of an application. The more accurate are the test cases, the better will be testing process.

#### D. Software Reliability Estimation

Testing also helps to estimate the reliability of software. Software reliability is the probability of failure-free software operation for a specified period of time in a specified environment. Reliability estimation helps to find the number of failures occurring in a specified amount of time to find the mean life of software and to discover main cause of failure etc.

#### E. Minimum Cost and Effort

Testing is too expensive- it's a myth. There is always a saying that we should pay less for testing and more for maintenance.

But in actual, if there will be no proper testing, it may result in improper design of software that will be expensive to handle and there will be great loss of time and money.

#### F. Gain Customer Confidence

Software testing helps to gain confidence of customers by providing them a quality product.

### IV. TOP QUALITIES OF HIGHLY EFFECTIVE TESTERS

#### A. Keen Observer

A good software tester must be a keen observer so that he can keep track of all the details about the project and can organise the testing in a better way. This attribute helps the tester to remain updated and informed about the test progress.

#### B. Good Communication Skills

One of the most important skills that a tester must have is- Good communication skills. A tester must have good verbal and written communication skills so that he can excel in writing test plans, test cases, handling defects etc.

#### C. Analytical Skills

Analytical skill is the skill of performing an analysis and analysis is the key process of testing. Therefore, proper analytical and technical skills are required to become a good

software tester so that correct requirement analysis and test plans can be conducted.

#### D. Good Time Manager

Time is the biggest constraint in the process of software development. A software with extraordinary functionalities, best coding, and proper design can be of no use to customer if it is not delivered on desired time. Testing is the part of software development lifecycle (SDLC), hence, early testing must be done at each and every phase of SDLC to prevent delay in software delivery process. For this, testers must be able to manage time in such a way that desired testing results can be achieved and at the same time, software can be delivered on time.

#### E. Quality-Oriented

Great testers are always passionate and quick learners. They keep a positive attitude. Quality-oriented nature means a tester should always think about how quality of software can be enhanced. A tester must be user-oriented, he should always think about user requirements so that proper software quality attributes can be achieved.

Other traits of a good software tester can be creative mind, Out of box thinking, great attitude, intellectual curiosity etc.

### V. PRINCIPLES OF SOFTWARE TESTING

Principles are basic ideas or rules that explains or control how something happens or works. There are seven principles of Software Testing-

#### A. Testing Shows Presence of Defects

The first principle states that testing talks about the presence of defects and don't talk about the absence of defects. The goal of testing is to make the software fail. In case, testers are unable to find defects after repeated regression testing does not mean that software is bug-free.

#### B. Exhaustive Testing Is Impossible

The process of testing the functionality of a software in all possible inputs (valid or invalid) and preconditions is known as exhaustive testing. Instead of exhaustive testing, risk analysis and priorities should be used to focus testing efforts. For example, if we are testing a text box that accepts numbers between 0 to 100, we would test for boundary values, on less than boundary value, one more than boundary value, middle number, few random numbers, that's it and assume that if it is working fine for these numbers, it will work for other numbers also. We can't test for each number, that's very impractical and complicated.

#### C. Early Testing

To find the defect in the software, early test activity shall be started. The defect detected in early phases of Software Development Lifecycle will be less expensive. So, conducting early testing reduces the cost of fixing defects.

#### D. Defect Clustering

It states that, in a project, a small module can contain most of the defects. Pareto principle states that 80% of the problems are found in 20% of the modules. Most of the reported defects are related to small number of modles within a system.

### E. Pesticide Paradox

If you keep running the same set of tests over and over again, chances are no more new defects will be discovered by those test cases. To overcome this, the test cases need to be regularly reviewed and revised, adding new and different test cases to help find more defects.

### F. Testing Is Context Dependent

Testing approach depends on context of software developed. Different types of software need to perform different types of testing. For example- The testing of E-commerce site is different from the testing of the Android application.

### G. Absence of Errors- Fallacy

99% of bug-free software may still be unusable, if wrong requirements were incorporated into the software and the software is not addressing the business needs. In this case, finding and fixing defects does not help. "No Amount of Testing Can Prove a Software Right a Single Test Can Prove a Software Wrong"

## VI. LEVELS OF TESTING

A level of testing is the stage at which the software must be tested. There are four recognized levels of testing-

### A. Unit Testing

In Unit testing, the smallest testable parts of an application called units are tested independently. It focuses on smallest element of the software system called module. That's why, it is also called module testing. It is done by developers.

#### 1) Example

Consider the example of Calculator. To conduct unit testing on the calculator system, individual modules like Addition, subtraction, multiplication, division must be tested independently.

### B. Integration Testing

The purpose of integration testing is to expose faults in the interaction between integrated units. It is the process of testing the interface between two software units. It can be done in three ways- Big-bang approach, top down approach and bottom up approach.

### C. System Testing

System testing is nothing but the testing of complete system. It is the last test that the developer performs before delivering the software to public for acceptance testing. Different types of system testing are followed like usability testing, stress testing, regression testing etc. First and important step in system testing is to prepare System test plan.

### D. Acceptance Testing

It is a level of testing where a system is tested for acceptability. It has various types' like- Alpha testing, beta testing, user acceptance testing and business acceptance testing. It is done by end users. Its outcome provides an important quality indication for the customer to determine whether to accept or reject the product.

## VII. EXEMPLAR ILLUSTRATION OF SOFTWARE TESTING

For practical and better understanding, some test scenarios of WHATSAPP- Messaging application are provided below. These are not the actual test cases in a test case format but a general layout how manual testing is performed. There would be numerous test cases for Whatsapp, but here, tests are limited to some of the high level features according to their functionality.

### A. Installation Testing

This type of testing is performed to verify if the software has been installed with all the necessary components or not. Whatsapp will be tested whether it can be installed or not.

- Check that user can download the Whatsapp application from play store or not.
- Check that user can register with a new mobile number.
- Check that user is getting a verification code on his mobile.....

### B. Test Case on Status

- Check that user can set DP (Display Picture).
- Check that user can set status on Whatsapp.
- Check the status privacy and settings.
- Check that recent updates are available or not.

### C. Test Case on Chats

- Check that Chats window contains the entire chat list.
- Check that Chats window shows the last updated chat time.
- Check that Chats window contains the group chat list.
- Check that Chats window contains the users whose DP is not available.
- Check that user can send and receive media files in chat to individuals.
- Check that user can check all messages delivered and received.
- Check that CREATE NEW GROUP is available or not.
- Check that user can give group name.

### D. Test Case on Calls

- Check that call history is available or not.
- Check that call history shows the called date and time.
- Check that search is working fine or not.
- Check that user can call and receive calls from person in his contact list.

### E. Design based Testing

Design based testing is the testing in which test cases are designed according to the detailed or architectural design of the system.

- Does the Whatsapp follow the design specification?
- Check that call section is available or not.
- Check that Chat section is available or not.

### F. Non-Functional Testing

It is the testing of software for its non-functional requirements.

- Test whether the application looks like an application.
- Look and feel must be good.

G. *Negative Test Cases*

- Check that how Whatsapp perform when user try to send data to blocked group.
- Check that how Whatsapp perform when app gets crashed.
- Check that how Whatsapp perform when user(Not admin) try to remove a person from group.

Hence, the above mentioned test scenarios prove that Testing is a broad concept. We not only need to write POSITIVE test cases, but NEGATIVE test cases as well. Therefore, each and every functionality must be tested in order to attain favourable testing results. So, this was just a clear-cut example that summarised the significance of Software testing.

VIII. SOFTWARE TESTING LIFE CYCLE (STLC)

Software testing lifecycle is a sequence of activities conducted to perform software testing in a systematic and planned manner. In STLC, different activities are carried out to improve the quality of product. STLC is a subset of Software Development life Cycle (SDLC). The phases of STLC are explained below-

A. *Requirement Analysis*

In this first phase of STLC, test team studies the requirements and checks whether the requirements are testable or not. Business requirement specification and Software requirement specification play a very important role in this phase.

B. *Test Planning*

Once the test team is clear with the requirements, the testers can start making test plans. Test plan is nothing but a proper strategy or approach with the help of which, testers conduct their testing.

C. *Test Case Development*

This phase involves the creation of test cases and test cases are also reviewed after their finalization. Test team also prepares the Requirement Traceability Matrix in this phase.

D. *Test Environment Setup*

Test environment decides on which conditions software is tested. This phase can run in parallel with design phase. The deliverables in this phase are test environment and smoke test results.

E. *Test Execution*

After test environment setup, execution of test cases is performed based on the defined test plan. All the positive and negative test cases must be executed and documented in a proper format. Defect report should be prepared for failed test cases and should be reported to the development team for rectification.

F. *Test Closure*

Once testing is completed, matrix, reports and results are documented. It is the last phase of STLC.

STLC is a very important phase of SDLC and final product cannot be released without passing through testing process. The different phases of Software Testing Life cycle must be executed in the same order in which they are defined

from start till end, otherwise desired testing results will not be noticeable. Moreover, to conduct testing well, the various terminologies involved in the testing process must be crystal clear to the testers.

IX. SOFTWARE TESTING TOOLS

Software testing is of two types- Manual and automation. Automation testing can be done with the help of several tools. There are many software testing tools available, so it becomes quite difficult to select the best tool for the project and these tools are also classified on the basis of certain parameters like Test management tools, load testing tools, mobile testing tools, defect tracking tools, security testing tools etc. Some of the general testing tools are summarised below-

A. *Selenium*

Selenium is a free open source portable framework for testing web applications. It is written in Java. It has many components like Selenium IDE, Selenium Client API, Selenium web driver, Selenium remote control etc.

B. *Testpad*

Testpad is a simple test management tool. There are many features of testpad like it is mobile and tablet friendly, easy in, easy out, natural way to test, drag and drop organisation etc.

C. *Zephyr*

Zephyr, a test management software provides a suite of tools to optimise speed and quality of testing. When the test load increases, agile team expands, Zephyr ensures reliability and testability.

Apart from the above tools, there are many other tools such as qTest, QMetry, QAComplete, Ranorex etc.

X. A COMPARATIVE STUDY OF PRESENT SCENARIO OF SOFTWARE TESTING WITH REGARD TO SOFTWARE TESTING IN THE EARLY DAYS

	TESTING IN THE EARLY DAYS	PRESENT SCENARIO OF TESTING
1	Management style was command and control.	Management style is leadership and Collaboration.
2	Risk was high.	Risk is low.
3	Scope of testing in the early days was limited.	Scope of testing in the present world is Mature and widespread.
4	Reuse was low.	Reuse is high.
5	Planning of testing was predictive.	Planning of testing is adaptive.
6	Communication between developers and testers was formal.	Today, the communication is informal.
7	Primary objective was high safety.	Primary objective is quick value.
8	Remodelling was expensive.	Remodelling is less expensive.
9	User involvement was low.	User involvement is high.



10	Knowledge management was explicit.	Knowledge management is tacit.
11	INPUT+PROGRAM+COMPUTATION→RESULTS	INPUT+RESULT+COMPUTATION→PROGRAM

Table 1:

#### CONCLUSION

Software Testing is an activity that is performed for evaluating software quality and also for improving it. It is one of the broader topics that wants immediate attention in this era of new and higher demand of Quality Software. To carry out testing in a more effective manner, this paper presented a comprehensive account of all the terminologies that are related to software testing. Generally, it happens that people know the best techniques, methodologies, tools that are involved in testing, but they don't know the basic testing goals and this is reflected in their bad testing reports. So, to perform testing efficiently and accurately, everyone involved in testing should be familiar with basic software testing goals, objectives, principles and concepts. Then only, a robust, reliable, accurate, flexible and efficient Software can be delivered to public. It is the need of the hour to focus more on WHY TO TEST rather than HOW TO TEST because when the WHY is clear then, the HOW is easy.  
KEEP CALM AND TEST ON...!

#### REFERENCES

- [1] Glenford J. Myers, Corey Sandler and Tom Badgett, 2011. The Art of Software Testing(3<sup>rd</sup> edition), Wiley Publishing.
- [2] Rasneet Kaur Chauhan & Iqbal Singh, Latest research & development on Software Testing techniques and tools, Article published in International Journal of Current Engineering and Technology, Vol.4, No. 4(Aug-2014).
- [3] Naik & Tripathy,"Software Testing & Quality Assurance",(Ist edition),Wiley Publishing, 2008-08-18.
- [4] Mohd. Ehmer Khan & Farmeena Khan, Importance of Software Testing in software Development Life Cycle, International Journal of Computer science Issues, Vol.11, No.2, March 2014.
- [5] Priya Shukla & Deepesh Mishra, A Composition on Software testing, International Journal of Technical Research and Applications, Vol.2, (March-April 2014).
- [6] Bindia Tarika, Review on Software Testing Techniques, International Journal on Recent and Innovation Trends in Computing & Communication, Vol.2, (January-2014).