ISSN 2278-3091

International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 01-04 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

Software Testing, Analysis and Objectives

Dr. Leelavathi Rajamanickam
Senior Lecturer,
SEGI University, Kota Damansara, 47810 Petaling Jaya, Selangor, Malaysia.
leelavathiraj@sedi.edu.my

Abstract*:* Software testing is about testing a feature with varying test data to get a result and then comparing the actual result with expected result. It is not merely finding defects or bugs in the software; it is the completely dedicated discipline of evaluating the quality of the software. Software testing efforts its goals or objectives but it does have certain limitations, still testing can be done more effectively if certain established principles are to be followed when we are doing testing of any software product. The objective of software testing is to identify all defects in a program. However, in practice, even after satisfactory completion of testing phase, it is not possible to guarantee that a program is error free. This is because the input data of most programs is very large and it is not practical to test the program exhaustibly with respect to each value that the input can assume. Even with this obvious limitation of testing process, we cannot underestimate the importance of software testing. Software testing is mainly conducted at three levels: Unit,

Integration and System. In other words, software testing is a process which is used to measure the quality of software developed.

Key words : Software testing; Objectives; Importance; Unit testing; Integration testing; System testing; Black-box; White-box; Non-functional testing; Limitations

INTRODUCTION

Software testing is a set of activities conducted with intent of finding errors in a software product. It also verifies and validate whether the program is working correctly with no bugs or not; it analyse the software finding bugs. In other words, software testing is a process used to identify the correctness, completeness and quality of developed computer software. 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. Testing is important because software bugs could be expensive or even dangerous.

Final Stage

The final outcome defined is based on requirements specification of the test execution.

Actual Stage

The actual outcome received after applying the test data to the software/feature as per steps defined in test case under controlled test environment.

Software testing is a process of verifying and validating that a software application or program meets the business and technical requirements that guided its design and development and works as expected and also identifies

Important errors or flaws categorized as per the severity level in the application that must be fixed. Technical requirements that guided its design and development and works as expected and also identifies important errors or flaws categorized as per the severity level in the application that must be fixed.

Testing a program involves providing the program with set of test inputs (or test cases) and observing if the program behaves expected. If the program fails to behave as expected, then the input data and the conditions under which the failure occurs are noted for later debugging and error correction. The following are some commonly used terms associated with testing.

- An **Error** is mistake committed by the development team during the development phases. The mistake might have been committed in the requirements, design or code. An error is also sometimes referred to as fault, a bug or a defect.
- A **Failure** is a manifestation of an error (or defect or bug). In other words, a failure is symptom of an error.
- A **Test case** is the triplet [I, S, O], where I is the input to the system, S is the state at which the data is input and O is the expected output of the system.
- A **Test suite** is the set of all the test cases with which a given software product is tested.

Table 1: Software Development and Software Testing Phase Correspondence Comparison.

No.	Software developing phase	Corresponding software testing phase
1	User requirements	User requirements verification and confirmation Acceptance test design
2	Requirement analysis and System design	Requirements verification and confirmation System test design
3	Conceptual design	Conceptual design verification and confirmation Integration test design
4	Detailed design	Detailed design and verification Unit test design
5	Coding	Unit test
6	Module integration	Integration test
7	System implementation	System test

ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 01-04 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

IMPORTANCE OF SOFTWARE TESTING

Software testing is a process which is used to measure the quality of software developed, it is also a process of uncovering errors in a program and makes it feasible task. In order to ensure the software quality, conducting software testing in every phase of software development process. A complete software testing should cover the entire life cycle of one software product.

As we all are human beings and human beings commit errors in any process, some of the errors do not impact much on our day to day life and can be ignored; however some errors are so severe that they can break the whole system or software. In such kind of situations you need to take care that such errors are caught well in advance before deploying the system/software in production environment.

For example: Let's take some ABC banks net banking website, if any net banking customer logs into the website and transfers some amount to other account. After transferring he got the confirmation that amount got transferred successfully and it got deducted from his account. After sometime when he confirmed with other person whom he transferred the amount he got to know that he has not received the amount, now you
need to visit your bank to settle this dispute. Also you will get annoyed by the net banking experience of that bank. This is one situation where testing your software is very important before deploying it. Testing cannot be ignored because it impacts all the end users of software.

If the website would have got tested thoroughly for all the possible user operations this problem would have been found well in advance and got fixed before deploying the website. In additions, software testing is important:

- Detect the existence of bugs
- Ensure expected functionality
- Increase customer satisfaction
- Reduce legal liability
- Ensure compliance with specification.

OBJECTIVES AND PRINCIPLES OF TESTING

A. *Objectives of software testing:*
Software testing has following objectives:
- To ensure that application works as specified in requirements document.
- To provide a bug free application
- To achieve the customer satisfaction
- To ensure that error handling has been done gracefully in the application. In situations when user has entered incorrect data, the application display user-friendly messages.
- To establish confidence in software
- To evaluate properties of software
- To discuss the distinctions between validation testing and defect testing
- To describe strategies for generating system test cases
- To understand the essential characteristics of tool used for test automation

B. *Principles of software testing:*
- Testing must be done by an independent party.
- Assign best personnel to the task
- Testing should not be planned under the tacit assumptions that no errors will be found.
- Test for invalid and unexpected input conditions as well as valid conditions.
- Testing is the process of executing software with the intent of finding errors.
- Keeps software static during testing.
- Documents test cases and test results.
- Provide expected test results if possible.

SOFTWARE TESTING STRATEGY

Testing is done for various purposes based on the need, necessity and importance. A complete testing strategy consists of testing at various points in the production process and can be described by the **test vee.**

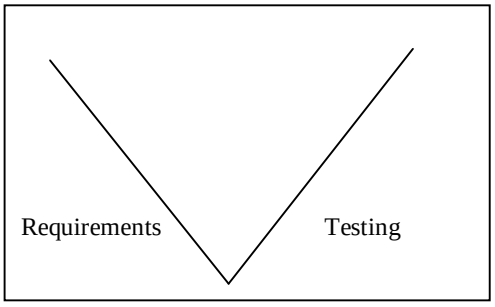

Fig 1: Process of Vee Test

The left hand side of V indicates the processes involved in construction of the software, starting with the determination of the requirements. The designed phase in this in this model is taken to indicate the design of modules and code phase the detailed design and construction of subprograms. The right hand side of V indicates the testing actions which correspond to each stage on the left side. Unit testing is concerned with testing of subprograms. Integration testing is assembly of modules to produce the application. System testing is process of testing the fully developed application.

LEVELS OF TESTING

Testing starts from testing a simple screen, sub-module, module, combination of modules and finally testing the complete system. Broadly, there are three levels of testing in each development cycle:
- Unit testing
- Integration testing
- System testing

A. *Unit Testing:* Unit testing focus on the verification effort of the smallest unit of software design-the unit. The units are identified at the detailed design phase of software

ISSN 2278-3091
International Journal of Advanced Trends in Computer Science and Engineering, Vol.3 , No.5, Pages : 01-04 (2014)
Special Issue of ICACSSE 2014 - Held on October 10, 2014 in St.Ann's College of Engineering & Technology, Chirala, Andhra Pradesh

development life cycle, and the unit testing can be conducted in parallel for multiple units.

B. *Integration Testing:* After unit testing, modules shall be assembled or integrated to form the complete software package. Integration testing is a systematic technique for verifying the software structure and sequences of execution while conducting tests to uncover errors associated with interfacing. Integration testing is subdivided as follows:

- Big-bang approach
- Top-down approach
- Bottom-up approach
- Sandwich (Mixed) approach

C. *System testing:* After the software has been integrated (constructed), sets of high order tests shall be conducted. System testing verifies that all elements mesh properly and the overall system function/performance is achieved. The purpose of system testing is to fully exercise the computer based system. The aim is to verify all the system elements and validate its conformance against Software Requirements Specification (SRS).

CATEGORIES OF TESTING TYPES

All the testing types are explicitly or implicitly part of three high level categories as mentioned below:

- Black-box testing/ Functional testing
- White-box testing/Structural testing
- Non-functional testing

A. *Black-box testing:* In black-box testing, test cases are designed from an examination of the input/output values only and no knowledge of design or code is required. It is an external view of the test object and verifies the functional behavior of the test object in the software product to ensure that system does what its "Expected "to do. Black-box testing is sometimes also called as functional testing or Behavioral testing. It is the test method in which user always test the application functionality and he need not bother about the application structure because the customer always looks at the screens rather than how it is developed. Usually Test Engineers do the Black-box testing.

Black-box testing uncovers errors of the following categories:

- In-correct or missing functions.
- Interface errors.
- Errors in the data structures or external data base access.
- Performance errors
- Initialization and termination errors

The following are two main approaches to design black-box test cases:

- Equivalence class partitioning.
- Boundary value analysis

B. *White-box testing:* White box testing is an internal view of test object and verifies that failures in the code by writing test code against it. White box testing of software is designed for close examination of procedural detail and providing test cases that exercise specific sets of conditions and/or loops tests logical paths through the software. White box testing is a test design method that uses the control structures of the procedural design to derive test cases. The test cases derived from white box testing method s will:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure their validity.

White-box testing is the testing method in which the user will test the application structure for its behavior. Usually Developers perform the white box testing.

A White box testing can either be coverage-based or fault-based.

Fault-based testing: A fault-based testing strategy targets to detect certain types of faults. An example of fault-based strategy is mutation testing.

Coverage-based testing: A coverage-based testing strategy attempts to execute certain elements of a program. Popular examples of coverage-based testing strategy are statement coverage, branch coverage, condition coverage and path coverage.

C. *Non-Functional testing:* Non-functional testing describes the tests used to measure the software characteristics such as response time, page load times, peak load limit, threshold limit for optimum performance of the software product on varying scale etc. Performance testing is carried out to check whether the system meets the non-functional requirements identified in the SRS. Software performance testing is a means of quality assurance (QA). It involves testing software applications to ensure they will perform well under their expected workload. Performance testing is also known as non-functional testing.

The focus of Performance testing is checking a software program:

- Speed – Determines whether the application responds quickly.
- Scalability – Determines maximum user load the software application can handle.
- Stability – Determines if the application is stable under varying loads.

Non-functional testing includes different testing types as mentioned below:

- Performance testing
- Security testing
- Automation testing
- Load testing
- Stress testing
- Volume testing
- Recovery testing

LIMITATIONS OF TESTING

Limitations are principles that limit the extent of something. Testing also has some limitations that should be taken into mind to set realistic expectations about its benefits. In spite of being most dominant verification technique, software testing has following limitations:

1. Testing can be used to show the presence of errors, but never to show their absence. It can only identify the known issue or errors. It gives no idea about defects still uncovered. Testing cannot guarantee that the system under test is error free.

2. Testing provides to help when we have to make a decision to either "release the product with errors for meeting the deadline" or to "release the product late compromising the deadline".

3. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.

4. Software testing does not help in finding root causes which resulted in injection of defects in the first place. Locating root cause of failures can help us in preventing injection of such faults in future.

CONCLUSION

This paper deals with a brief analysis on importance, objectives, types and limitations of software testing. Software testing is very important phase of software development life cycle (SDLC). Software testing is vital element in the SDLC and can furnish excellent results if testing is done properly and effectively. Unfortunately, software testing is often less formal and rigorous than it should and a main reason for that is because we have struggled to define best practices, methodologies, principles, standards for optimal software testing. To perform software testing effectively and efficiently, everyone involved with testing should be familiar with basic software testing goals, importance, types and limitations and concepts. Already lot of work has been done in this field, and even continues today. Implementing testing principles in real world software development, to accomplish testing objectives to maximum extent keeping in consideration the testing limitations will validate the research and also will pave a way for future research.

REFERENCES

[1] S.M.K Quadri, Sheikh Umar Farook, "Software testing-Goals, Principles and Limitations", International Journal of Computer Applications, Volume 6- No. 9, September 2010, pp 7-10.
[2] Fangchun jiang, Yunfan Lu, "Software testing model selection research based on Yin- Yang testing theory", IEEE, International Conference on Computer Science and Information Processing (CSIP), Vol. 9, 2012, pp. 590-594.
[3] Mohd. Ehmer Khan, "Different forms of software testing techniques for finding errors", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No. 1, May 2010, pp.11-16.
[4] Sahil Batra, Dr. Rahul Rishi, "Improving Quality using testing strategies", Journal of Global Research in Computer Science, Volume 2, No. 6. June 2011.
[5] Cem Karner, "Testing Computer Software", 1993.
[6] Sheetal Thakare, Savita Chavan, Prof. P.M.Chawan, "Software testing strategies and techniques", International Journal of Emerging Technology and Advanced Engineering, pp.567-569, Vol. 2, Issue 4 April 2012.
[7] Abhijit A. Sawant, Pranit H. Bari and P.M Chawan,"Software Testing Techniques and Strategies", International Journal of Engineering Research and Applications (IJERA), pp. 980-986, Vol. 2, Issue 3, May-June 2012.
[8] Jitendra S. Kushwah, Mahendra S. Yadav, "Testing for object oriented software", Indian Journal ofComputer Science and Engineering (IJCSE), Vol. 2, No. 1, pp.90-93.
[9] Gaurav Saini, Kestina rai, "Software testing approach for efficient bug finding with yin-yang testing Theory", International Journal of Advanced and Innovative Research (IJAIR), Vol.2 Issue 4, pp.259-262, April 2013.
[10] Rajib Mall, "Fundamentals of Software Engineering", Third edition, prentice hall of India, August 2011.
[11] Dr. S.S Riaz Ahmed,"Studying the feasibility and importance of software testing", International Journal of Engineering Science and Technology, Vol. 1(3,), pp.119-128, 2009.
[12] Software testing Help available at http://www.softwarestestingmentor.com.
[13] Ian Sommerville, "Software testing", Software Engineering, 7th edition, chapter 23.
[14] Introduction to software testing available at http://onestoptesting.com/introduction/
[15] Paper by Lu Luo available at http://www.cs.cmu.edu/~luluo/Courses/17939Report.pdf
[16] Software testing by Jiantao pan available at http://www.ece.cmu.edu/~roopman/des-899/sw-testing/
[17] Software testing help tutorial available at http://www.guru99.com/software-testing.