Software Testing: Affirming Software Quality

Dr. Poonam Chaudhary
System Programmer, K.U.K.

Seema Sangwan
Lecturer, *DCSA, Pt. N.R.S. Govt. College, Rohtak (Haryana)*

Abstract - Today, in Computerized and highly automated era, software process/products are vital aspects in our lives which therefore requires software systems to be highly reliable, usable, functional, efficient and of high quality. Hence testing of software systems from evolving phase and at every stage is imperative to attain above qualities. Software testing aims at evaluating the capability of a program/system and to affirm the quality of software systems by systematically exercising the software in carefully controlled circumstances. Software testing ensures quality assurance, reliability estimation, verification and validation. Software testing executes program with actual test data which produces expected results with the intention of finding the errors. Testing illustrates differences between the expected behavior specified by the requirements and the observed behavior of the system. A good software test has a high probability of finding an as yet undiscovered error and a successful test discloses undiscovered error. This paper discuses about Software testing and elaborates its objectives and testing techniques.

Keywords: SDLC, Software Testing, Software quality, Software Testing Techniques

I. Introduction

Software testing is both a discipline and a process. 'Testing' can be defined as an activity which is planned and carried out methodically. Testing is integral party lifecycle of the software. Testing requires more project effort than any other software engineering activity.

Software testing is analyzing a system or a component by providing defined inputs and comparing them with the desired output to check the discrepancies between the desired and actual outputs and correct them.

Software testing integrates software test case design techniques into a well planned series of steps that result in the successful construction of software.

II. Software Testing Objectives

1. To Improve Software Quality:

The effects of bugs are very severe; bugs errors cause huge losses and disasters. Few examples of bugs causing critical issues are airplane crashes, missing of space shuttles; stock market trading was stopped and so on. The quality of software is a vital aspect in the computerized world. Quality is conformation to the specified design requirement. Debugging is conducted to find out defects in design, developed by the programmer. It is not so easy for a developer to develop a program which is error free, we ensure good quality by finding the problems and fixing them.

2. For Reliability Estimation:

We can associate software, and the amount of testing it has been subjected to with software reliability. Testing serves as statistical sampling method to gain failure data for reliability estimation based an operational profile.

3. For Verification & Validation:

Software testing is often used in association with the term verification and validation. Verification is a process of testing or checking the deviation of actual results from the required ones. This activity is carried out in a simulated environment so that actual results could be obtained without taking any risks.

Validation refers by process of using software in a live environment in order to find errors. The feedback from the validation phase produces changes in the software to deal with bugs and failures that are uncovered.

Validation and Verification should be implemented at every stage in the software process.

Validation: are we doing the right job?

Verification: are we doing the job right?

Objective of Validation and Verification are:

- Discover defects in system.
- Evaluate system weather it is usable in operational situation or not.

III. Test Information Flow

Test information flow is an arrangement to check and maintain the right flow of information during software test[9]. In fig.1 we feed the software and test configuration specification to the testing block which generate test result. Then, these result are fed into the evaluation block , along with expected results and error rate data. The expected results and the test result are compared and error generated. Then there error are fed into the debug block. The debug block corrects the errors and producer corrected result.

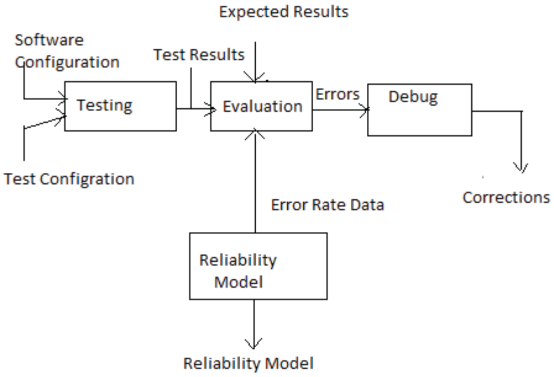

Fig:1 Representation of Test Information Flow

IV. Test Case Design

Test case design is an important activity in the testing process. Test case is an instance of a specific testing technique. Test case specification specifies the steps and the special requirements that are necessary for executing a set of test cases.

Advantages of Test Case Design:

1. To assure that the set of test cases used is of good quality.
2. The overall result of unit testing and effect of total set of test cases is visible to the tester.
3. The test cases are optimized, as evaluation of test set might indicate that few test cases are redundant.

V. Testing Methodologies

There are three methods to test software [9]:

1. Black Box Testing
2. White Box Testing
3. Grey Box Testing

Black Box Testing

Black box testing covers under unit testing in which components are tested individually. It is an integral part of ‘Correctness Testing’. Black box testing is also called functional testing. Black box testing involves only observation of the output for certain input values and there is no attempt to analyze the code which produces the output.

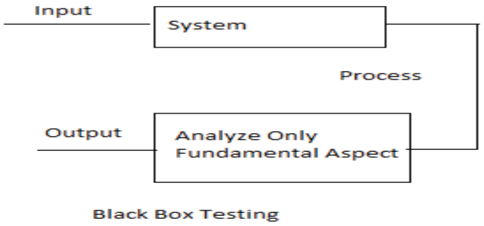

Fig:2 Processing Representation in Black Box Testing

Some of the Black Box testing techniques are:

1. Equivalent partitioning
2. Boundary Value Analysis
3. Cause-Effect Graphing Techniques
4. Comparison Testing
5. Fuzz Testing
6. Model Based Testing

Advantages of Black Box Testing:

1. Tests are done according to user's point of view.
2. Tester and programmer are independent of each other.
3. It is more efficient on larger units of code than white box testing.
4. Tester needs no knowledge of implementation
5. Test cases can be designed on the basis of specifications.

Disadvantages of Black Box Testing:

1. Many program paths remain untested.
2. Only small numbers of possible input can actually be tested.
3. Test cases are difficult to design without clear and concise specifications.

White Box Testing

White Box Testing is a kind of testing where the test groups should have complete knowledge of the internal structure of the software. It is also called structural testing. White box testing is considered as a security testing method which can be used to validate whether code implementation follows intended design, to validate implemented security functionality and to uncover exploitable vulnerabilities [5]

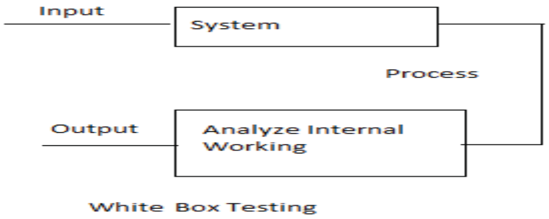

Fig: 3 Processing Representation in White Box Testing

Type of White Box Testing:

1. Basis Path Testing
2. loop Testing
3. Control Structure Testing

Advantages of White Box Testing:

1. This testing is more efficient on larger units of code
2. Errors in hidden codes are revealed
3. Developers give reason about implementation.
4. Side effects of such testing are beneficial.
5. Approximate the partitioning done by execution equivalence.
6. Internal structures will be timely checked to maintain their validity.
7. All loops will be exercised

Disadvantages of White Box Testing:

1. It is very expensive
2. Missed out the cases omitted in the code
3. It is not possible to check every bit of code to find out hidden errors.

Gray Box Testing

Gray Box testing is a mix combination of black box testing and white box testing [6]. It is a testing process required to test a piece of software against its specifications using some knowledge of internal working.

Gray Box Testing Methodology

Methodology involves ten steps as mentioned [5]:

1. Identify Inputs
2. Identify Outputs
3. Identify Major Paths
4. Identify Sub function (SF)Y
5. Develop Inputs for SF Y
6. Develop Outputs for SF Y
7. Execute Test Case for SF Y
8. Verify Correct Result for SF Y
9. Repeat Steps 4:8 for other SF
10. Repeat Steps 7&8 for Regression

VI. SOFTWARE TESTING STRATEGY

Software testing strategies integrates software test case design methods into a series of steps resulting in a successful construction of software. Testing strategies are defined by project managers, software engineer and testing specialist. Software testing includes four types of testing:

1. Unit Testing
2. Integration Testing
3. Acceptance / Validation
4. System Testing

Unit Testing

It is the most micro scale of testing, mainly focuses on each unit (component) of the code. It is done by the programmers. It requires knowledge of the internal program design and code.

Methods involved in Unit testing:

1. White Box Testing Techniques
2. Test Coverage Techniques

Integration Testing

It is the systematic technique for constructing the program structure. All the components are integrated to form complete software package.
Different integration testing strategies are:
Top down integration

Develop the skeleton of the system and populate it with components.

Bottom up integration:

Integrate infrastructure components then add functional components.

Validation/Acceptance Testing

It is also known as user acceptance testing which is carried out in order to verify if the product is developed as per standards and specified criteria and as per user requirements.
Types of Acceptance Testing:

1. User Acceptance Testing
2. Alpha Testing & Beta Testing
3. Operational Acceptance Testing
4. Contract and Regulation Acceptance Testing

Methods involved in Integration Testing is Black Box techniques

System Testing

System testing verifies whether all elements are interconnected correctly, and it checks that the entire system performance is achieved or not. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system
Types of system testing are as follows:-
1. Recovery testing
2. Security testing
3. Graphical user interface testing
4. Compatibility testing

VII. Conclusion

The main attribute for a reliable and efficient software system is its quantified high quality in design and functionality. Quality measurement is done by software testing at each phase from the inception of software system.
In this paper I have discussed the objectives of software testing and basic principles of Black Box testing, White Box Testing and Gray Box testing with their pros and cons. Also have described different software testing strategies. This paper serves as a guide for developers who are new to testing of software. This objective is to put all the relevant issues into a unified content, due to wide concept of the field, each topic problem and approaches are briefly tackled.
It is concluded that best selection and implementation of strategies is the key in development of high quality assured software system, which also facilitates in productive future researches.

References

[1] Vishal D Rampure, Automated Software Testing, Advanced Software Engineering CSC 532
[2] http://www.automated-testing.com/PATfinal.htm

[3] IMPROVING QUALITY USING TESTING STRATEGIES, Sahil Batra*1, Dr. Rahul Rishi2, Volume 2, No. 6, June 2011,Journal of Global Research in Computer Science

[4] Boris Beizer, “Software testing techniques”, Second edition, (1990).

[5] Software Testing Techniques and Strategies, Abhijit A. Sawant, Pranit H. Bari, P. M. Chawan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.980-986

[6] Software Testing Techniques, Shivkumar Hasmukhrai Trivedi, Volume 2, Issue 10, October 2012 ISSN: 2277 128X, International Journal of Advanced Research in Computer Science and Software Engineering

[7] Software testing, by Ron Patton

[8] Software engineering, by Roger Pressman

[9] Software Testing Methods and Techniques Jovanović, Irena