

# “A COMPARATIVE STUDY OF VARIOUS SOFTWARE TESTING TECHNIQUES

Mrs. Shalin Bhola<sup>1</sup>, Mrs. Namrata Jain<sup>2</sup>, Mr. Shakti Kumar<sup>3</sup>

Department of Computer Science  
Govt. P. G. College Ambala Cantt  
Haryana, India

e-mail-shalinbhola@gmail.com<sup>1</sup>, namratagc@gmail.com<sup>2</sup>, shaktikumarbajpai@gmail.com<sup>3</sup>

**Abstract**— Software Testing is a process in which we investigate about the quality of the software being tested. The main purpose is to identify the bugs in the software to prevent any future failure and to enhance the reliability and robustness of the software. In other words, we refer to validate and verify the software. Different commonly used testing techniques are discussed to test the software- black box testing, white box testing and gray box testing. In this paper we are going to provide the comparative difference between these techniques, as well as we will discuss the various other techniques which are used to test the software.

**Keywords**—Software development models; Scheduling Jobs; Cost Benefit Analysis; Risk Calculation; Comparison of various models.

## I. INTRODUCTION

Software Development Life Cycle is a process which explains the overall life of development of the software in terms of phases and also explains the order in which they are to be executed. Software testing is one of the phases of this cycle which refers to test the quality of the software. It is very important activity which accesses the functionality and correctness of a program through execution or analysis[1]. The main purpose of testing is to validate and verify the software. By the term validate we ensure that the product actually meets the user needs and the specifications. At the same time verification refers to ensure that the product is built according to the user's requirements and specifications. Testing does not guarantee a high quality system as it cannot identify all the defects in the software. It almost take 40-50% of development effort to test a software.

### A) Goals of software testing

Software testing is referred to a very important activity in life of software development as it aims to

1. System meeting the requirements and specifications. Responds correctly to all kinds of inputs.
2. Is sufficiently usable.
3. Can be installed and run in its intended environments.
4. Performs its functions in acceptable time.[2]

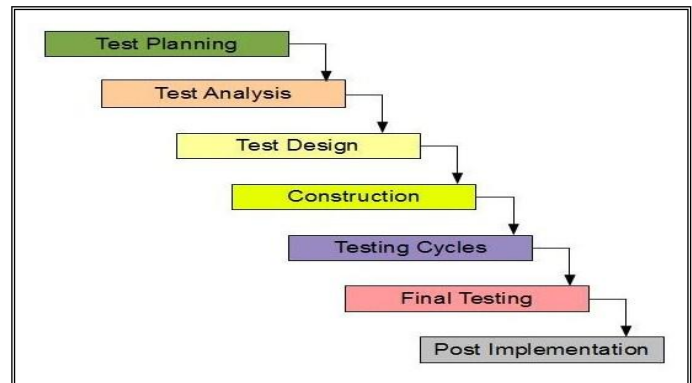


Fig.1 Software Testing Life Cycle

### B) levels of testing

#### 1) Unit Testing-

It refers to test a single module's functionality. It focuses on the working of a particular function. With the help of this testing, it is checked whether the modules of software can work independently of each other or not. [3]

#### 2) Integration Testing-

As the name implies the unit modules are combined and tested as a group for their compatibility.

#### 3) Component Interface Testing-

The type of testing technique that focuses on the interface between the components.

#### 4) System Testing-

It tests the complete integrated system whether it meets the specifications and requirements.

#### 5) Operational Acceptance Testing-

It tests whether the system is ready for operation or not.

### C) Software testing methods

#### 1) Black Box Testing-

It is basically a dynamic test approach in which focus is not on the internal details but at the functioning of the software. It totally deals with the user's requirements and specifications. The testers are

concerned of what the software does not how it does it. This testing is applicable to all levels of testing.

The various techniques available in black box testing are:

- **Equivalence Partitioning-** This technique partitions the data of the software into equivalence classes from which test cases are derived. In this, test cases are designed to cover each partition at least once. It reduces the time required to test the software because of lesser number of test cases.[4]

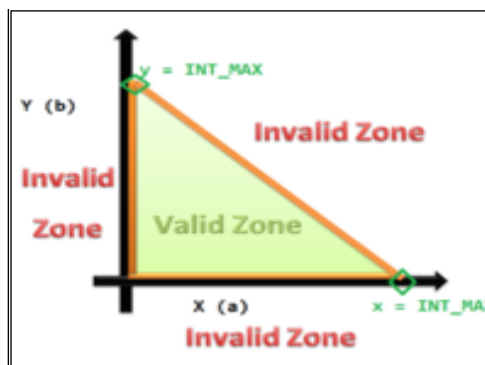


Fig. 2 Evaluating test cases

- **Boundary Value Analysis-** This technique is the refinement of the equivalence partitioning technique. It focuses on testing at boundaries, or where the extreme boundary values are chosen. It includes minimum, maximum, just inside/outside boundaries, error values and typical values [1]. Here focus is on the boundary of the input space to identify the test cases.

- **Cause Effect Graphing-** This technique establishes relationships between logical input combinations called causes and corresponding actions called effect. The cause and effect are represented with the help of a graph which is further transformed to decision table where each column represents a test case.

- **All Pairs Testing-** In this technique, test cases are designed by checking all possible combinations of each pair of input values.

- **Use Case Testing-** A use case is a list of actions. It develops the test cases on a system or acceptance level.

- **State Transition testing-** In this technique test cases are designed by changes to the state of the system.

### 2) White Box Testing-

In this technique, focus is on the internal working and structure of the system. In this programming skills

are used to design the test cases. A programmer should have complete knowledge regarding the internal functioning of the system. The various techniques used in white box testing are:

- **Statement Coverage-** In this type of testing, each statement of the code is executed at least once. It is a weak form of testing as it does not handles control structures fully.

- **Branch Coverage-** It focuses on executing each branch of each decision at least once. The weak point is that it may ignore branches within a Boolean expression.

- **Condition Coverage-** it tests all logical conditions in a program module.

- **Multiple Condition Coverage-** in this test cases are designed for all possible combinations of conditions.

- **Basis Path Testing-** Basis path testing allows the test case designer to produce a logical complexity measure of procedural design and then uses this measure as an approach for outlining a basic set of execution paths [1].

- **Loop Testing-** In this proper attention is given to loop constructs validation.

- **Control Flow Testing-** This technique focuses on control of data flow.

### 3) Grey Box Testing-

It combines the functionalities of both black box and white box testing. This testing uses the white box approach to design the test cases and uses the black box approach for execution purposes.

Comparative Analysis

S. No.	Black Box Testing	White Box Testing	Grey Box Testing
1	Internal working is Hidden and unknown	Internal working well known and clear	Internal working partially known
2	Can be done by trial and error method.	Data can be better tested.	Data can be tested if known.
3	Least time consuming and exhaustive.	Time consuming and mostly exhaustive.	Partly time consuming and exhaustive.
4	Performed by end users, testers and developers.	Performed by testers and developers.	Performed by end users, testers and developers.
5	Not suited for algorithm	Suited for	Not suited for

testing.	algorithm testing.	algorithm testing.	[3]
----------	--------------------	--------------------	-----

Table 1.Comparison Between Various Testing Techniques

## II. CONCLUSION

Software testing is the activity which accesses the functionality and correctness of a program through execution or analysis. It is a very important activity as the software functionality depends on it. To carry out software testing in a more effective manner, this paper provides a comparative study of three main techniques of software testing[7]

## REFERENCES

- [1] Anju Bansal, "A Comparative study of Software testing Techniques" IJCSMC, Vol. 3, Issue. 6, June 2014, pg.579 – 584, www.ijcsmc.com
- [2] Maheshwari & Jain "A Comparative Analysis Of Different Types Of Models In Software Development Life Cycle", Vol.

- 2, Issue 5 pg 286, www.ijarcse.com , ISSN: (Online) 2277-128X May 2012
- [3] Khurana & Gupta, "Study & Comparison Of SoftwareDevelopment Lifecycl Models", pg 1515-1516, 2012
- [4] A. K. Shukla, "Which model is best for software project? A Comparative Analyis Of Software Engineering Models" Vol. 76 pg-11, www.ijca.com ISSN: (Online) 0975-8887 Aug 2013
- [5] Boehm, Barry, "A SPIRAL MODEL OF SOFTWARE DEVELOPMENT AND ENHANCEMENT," from Proceedings of an International Workshop on the Software Process and Software Environments, 1985.
- [6] C. Ebert, P. Abrahamsson, N. Oza, "LEAN SOFTWARE DEVELOPMENT", IEEE Software, vol. 29, no. 5, pg. 22-25, 2012,
- [7] Vishal Sharma & preeti yadav, RISK STUDY OF VARIOUS SOFTWARE DEVELOPMENT METHODOLOGIES International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No.02, Special Issue No. 01, ISSN (online): 2348 – 7550 ,September 2014