

Review of Software Testing Techniques and Strategies

Department of Computer Technology
KG college of Arts and Science
Email: sathyavathy.v@kpcas.com

Abstract- Software testing plays a major role in the software development life cycle .The major issue in software testing phase is to detect bugs .Software testing is essential to reduce the errors and overall development costs. The major problem in the software testing is to generate the test cases. There are various number of software testing techniques available which performs its own functionality. The goal is to find the effectiveness in software testing with the least number of test cases. The main objective is to compare the software testing techniques or combination of testing techniques can be used to test the quality of the software product

Index Terms: Software Testing, Software testing Techniques, Testing Strategies

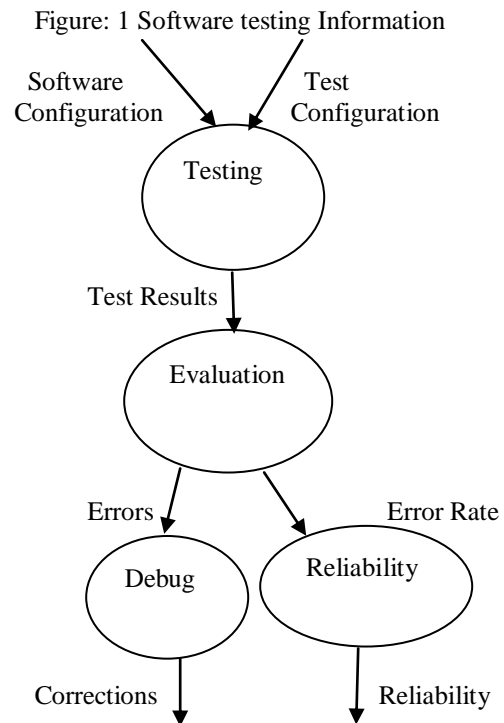
1. INTRODUCTION

Advancement and application of information technology are ever changing in SDLC [9].Software testing is the process of evaluating and detecting the errors. It performs verification and validation to check whether the developed software is correct. The software defects are identified, detected and isolated and subjected for rectification. Software Testing is an important area in research in the computer science is likely more important in the future. Testing occupies most of the development efforts and time that requires quality and reliability in the software engineering process. So various software techniques and tools are available to evaluate the system. The software testing techniques can be used to assess the software to determine its quality.

2. TAXONOMY OF SOFTWARE TESTING

2.1. Goal of software testing

The main aim of software testing technique is to ensure the quality of the software, by systematically applying various software testing strategies .Testing should focus on the intent of finding error. A good testing is the one that has high probability of finding an undiscovered error [6]. A test can be either successful or unsuccessful. A successful test is the one that identifies the uncovered errors. The category of software testing can be further divided.



2.2 Testing Levels

Software Testing is involved in every phase of the software development lifecycle [1]. There are various testing process that is carried out at each phase in the software testing.

2.2.1 Unit Testing

It is the smallest unit or piece of code that is being isolated. It is the basic unit which is done at the lower level. It is considered as a single module or unit or component within the software.

2.2.2 Integration Testing

This testing is performed mainly to expose defects occurring in the interfaces and the interactions within the interfaces of the components within the software. Integration testing is performed after the unit testing.

2.2.3 System Testing

System Testing is a black box testing technique performed to evaluate the complete system against the specified requirements. It includes the functional and nonfunctional requirements.

2.2.4 Acceptance Testing

It is a testing technique performed to analyze whether the software system has met the requirement specification. The main purpose is to verify the system with the user. The acceptance test cases are generated against the test data or using the acceptance test script and the results are compared with the existing ones.

3. TESTING TECHNIQUES

There are two types of testing techniques:

3.1. Structural testing technique

Structural testing is a white box testing technique in which it is used in all the phases of testing technique such as analysis, requirements and design [1]. Structural testing technique ensures the coding part of the program, which is very much essential.

3.2. Functional Testing Technique

Functional Testing technique is also known as black box testing technique. It is used to test the features/functionality of the software which includes all the paths and boundary conditions. There are various structural testing techniques. They are

Statement Coverage: This technique is aimed at exercising all, programming statements with minimal test set.

Branch Coverage: This technique is running a series of tests to ensure that all branches are tested at least once.

Path Coverage: This technique focuses on all possible combinations of paths which mean that each statement and branches are covered.

Statement Testing = (Number of Statements Exercised / Total Number of Statements) x 100 %

Branch Testing = (Number of decisions outcomes tested / Total Number of decision Outcomes) x 100 %

Path Coverage = (Number paths executed / Total Number of paths in the program) x 100 %

4. TEST SELECTION

The test case selection is one of the important parts in software testing process. It is known fact that the different test criteria will lead to detect many faults [2]. Fault removal technique is one of the major goal in the software testing and also to achieve the specified quality. Test sets that are good at one would not be suited for another.

4.1 Test selection on code

This strategy is called path based because they select each unique path as the test input. The basis of the test is to execute the path at least once, Test path selection that is executed could be a success or failure. It is not possible to cover all the paths within the program that leads to the infinite number of loops as the inputs. It is not all paths of the graph leads to actual program paths. The finding of input in the selected path is not only the problem but also it is a difficult task in practice. Additional tests are to be exercised, if some elements are to be uncovered to evaluate the effectiveness of the test suite.

4.2 Test selection on specification

In specification testing strategy, it is derived from the documentation relative to program specification. Specification based testing focuses on testing from UML models [2]. Early approaches of the testing is expressed in relation with the Input/output of the program and derived manually on boundary conditions, equivalence classes and cause effect graphs.

5. SOFTWARE TESTING PRINCIPLES

Different software testing principles are:

Test as a program: Testing is a process of executing a program with the intent of finding errors and faults. Testing becomes more efficient and effective when more number of failures is present.

Early Testing: When the testing activities are done at the early stages of testing the available time can be utilized better. When the defects are found at the early stage within the software development life cycle, the process becomes much easier and cheap.

Testing is context dependent: The testing, types and methodologies are related to the type and nature of the application [3]. Certain software applications need more number of testing process. For example in case of website testing it has to go through various

performances based testing that is not affected by the load on the servers.

Absence of errors

- Testing must be done by an independent party
- Assign best personnel to the task
- Test for invalid and unexpected input conditions as well as valid conditions
- Keep software static during test
- Provide expected test results if possible

6. SCOPE OF THE STUDY

6.1 Technical scope

In this paper technology of testing techniques which includes both the functional and structural testing techniques that have been most widely used in practice. The methodology and strategy are implemented that is used to select the test cases and analyze the test results. Software testing techniques research are roughly divided into two broad categories, methodological and theoretical, the growth of both the categories put the testing technology together.

Some of the areas include in the scope of the study is to ask some questions as:

- How testing is involved in the software development life cycle
- How testing is performed at different levels
- What are the testing process models used
- What are the management responsibility and testing policies used
- When to stop testing criteria and software testability

6.2 Goal and Standard of progress

The main goal of software testing is to help the analyst, software designers, developers and managers construct systems with high quality. Thus research and development in testing aims effective testing to find more errors in the phases of requirement, analysis, design and implementation and to improve the system with the high quality. The testing technique leads to the destination of testing methods and tools[6].The standards in the research includes the amount of technology inside, amount of dependability in other areas of software engineering, technical feasibility being used in practical and spread of technology in classes, trainings and management attention.

6.2.1 Test execution

The test input that provides the entry-exit paths and the corresponding program paths are to be identified. The test cases are the path to the sequences

of events that are specified in the abstraction level of the specifications. The specified test cases can be translated into executable runs, another one is the ability to include the system into the state from which the specified test is launched [8].The specific set of test involves the actions to implement in a subsystems. It also includes the test cases of input/output sequences and the external interfaces.

6.2.2 Test Oracle

An important testing component is the test oracle. An oracle is the one which decides the program that works correctly on the given set of tests. The oracle is specified to output that is either accepted or rejected. In this approach only limited numbers of test cases are included is the regression testing, in which the test output is compared with the earlier versions of the executions. The test oracle is derived from the specifications of the expected behavior.

6.2.3 Analysis of test results

The software testing effectiveness is based on the entire test suite and test case. The real software testing should be related to the program and should allow the testers to measure the effect of testing the program's attribute. The test criteria that found the highest number of defects would be more useful. If the testing is continued for a while then it implies that the program is correct or the tests are effective. The most objective measure of software testing is the statistical one [4].The statistical analysis of next set of test can be predicted based on the sample of the program's behavior. The analysis and documentation of test results require discipline and effort.

6.2.4 Notion of software reliability

Software reliability is one the important quality factor, in which the software will execute without failure in given environment [5] .After the tests are executed, the software reliability, is measured by obtaining the test results against the reliability model. Various models are used for reliability to present the situation that the program is tested and the failures are encountered.

7. PROCESS MODEL

A Process model is the way of presenting the phase of software in the concepts of verification and validation to prevent the delay between the defects. In this model each phase of the software management is characterized by the following:

Entry Criteria which specifies when the test can be started and also includes that are inputs for the phase.

Tasks or steps that are to be carried out in that phase with the set of measurements that characterize the tasks.

Verification that specifies the method of checking that the tasks are carried out correctly.

Exit criteria, that specify the conditions under which one can consider the phase as done and also includes the outputs for the phase.

8. EVALUATION OF SOFTWARE TESTING TECHNIQUES

Software testing should be effective enough to resolve critical damages on the entire system for the users, by taking into account of potential failures of the program and its environments [Kurokawa and Shinagawa, 2008]. One of the best ways to prevent such failures is to go for entire software testing of the system, which tests the entire system with all possible combinations of sets of inputs which includes both valid input and invalid test cases [7]. However, excluding test cases, whole system testing is an impractical thing for the most software systems. Besides, it is often faced with lack of time and resources, which can limit the ability to effectively complete testing efforts. A tester do not want to go for software testing, rather wants to select a testing technique in relation with the selected test strategy and techniques that will detect maximum possible faults and errors that brings the product to an acceptable level while consuming less resources and time. Whether the option is for static or dynamic testing, there is a selection of testing methods to choose from which the testing efficiency is detected. In each testing method there are so many software testing techniques that are used to test a system. Each software testing technique is meant for testing has its own characteristics i.e. for what purpose it is used, what aspect it will test, what will be its deliverables etc. Different approaches to software development require different testing methods and techniques [Tawileh et al., 2007]. This limits the ability to use a generic software testing technique for testing a specific system.

Table: 1 Comparison of testing technique

S . N o	Type	Testing Environment	Effectiveness	Testing Technique
---------	------	---------------------	---------------	-------------------

1	Random Testing	Black	Least Effective	Specificati on based
2	Function Testing	Black	Effective	Code based
3.	Control Flow	White	Effective	Code based
4	Data Flow	White	Effective	Code based
5	Mutation Testing	White	Most Effective	Fault based
6.	Regression Testing	White/ Black	Most Effective	Validation

9. CONCLUSION

Quality is the main focus of any software engineering project. Without measuring, we cannot be sure of the level of quality in software. So the methods of measuring the quality are software testing techniques. This paper relates various types of testing technique that we can apply in measuring various quality attributes. Software testing research is the driving element of development and application. In this era of new and higher demand of software testing, it is important to constantly summarize new achievements, fresh hotspots and propose different ideas in order to promote the study on software testing system engineering, to facilitate the rapid development on software testing field and industry. With the availability of so many software testing techniques and the very inadequate quantitative and qualitative knowledge about them, it is strongly believed that there is a necessity to further evaluate the testing techniques.

Currently the relative ordering of software testing techniques are unknown and yet to make the software testing more effective by selecting the appropriate testing techniques on an ordinal scale. Current situation calls for replication and further process of evaluation of testing technique is needed to acquire the knowledge about the effectiveness and efficiency of software testing technique for both detecting the errors and reliability criteria. To achieve the goal the experiment is to be done on a large scale but that needs to be compared, that will have no contradictions. For that it is necessary to set the common and standard set of parameters such that there are few variations in the experimental goals.

REFERENCES

[1] Stacey, D. A.(2004), Software Testing Techniques Guide to the Software Engineering Body of

- Knowledge, Swelok – A project of the IEEE Computer Society Professional Practices Committee.
- [2] R.S. Pressman & Associates, Inc. (2005). *Software Engineering: A Practitioner's Approach*, 6/e; Chapter 14: Software Testing Techniques,
- [3] Myers, Glenford J.(1979), IBM Systems Research Institute, Lecturer in Computer Science, Polytechnic Institute of New York, *The Art of Software Testing*, by John Wiley & Sons, Inc.
- [4] Redmill, Felix (2005), *Theory and Practice of Risk-based Testing*, Vol. 15, No. 1. IEEE(1990), *IEEE Standard Glossary of Software Engineering Terminology*, Los Alamitos, CA: IEEE Computer Society Press.
- [5] Zhang Hongchun, *Research on New Techniques and Development Trend of Software Testing*. Nancy Bordelon, A comparison of automated software testing tools
- [6] D. Janzen, H. Saiedian, and L. Simex. Test-driven development concepts, taxonomy, and future direction. *Computer*, 38(9):43–50, Sept. 2005.
- [7] H. Zhu, P. A. V. Hall, and J. H. R. May. Software unit test coverage and adequacy. *ACM Comput. Surv.*, 29(4):366– 427, 1997.
- [8] M.Lyu., *Handbook of Software Reliability Engineering*. McGraw-Hill, New York, and IEEE CS Press, Lo Alamos, 1996.
- [9] kavitha.M, *Innovative Trends in Information Technology* International Society for Research in Computer Science, Feb. 2016.