International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 4 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT195413

A Study on Software Process Framework and Testing Techniques

Dr L C Manikandan[1]*, Dr R K Selvakumar[2]
[1]Professor, Department of CSE, Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, India
[2]Professor, Department of CSE, CVR College of Engineering, Hyderabad, Telungana, India.
*Corresponding Author: lcmanikandan@gmail.comb

ABSTRACT

Software engineering is an engineering division connected with development of software product using well-defined scientific principles, techniques and actions. Two different types of activities like software framework activities and umbrella activities are included in standard software process model. Testing acts important job in software development process, i.e. the specified functionality and requirements are tested. The objective of this paper is to provide the software process framework and testing techniques for young researchers.

Keywords: Software Testing, Black-Box Testing, White-Box Testing, Grey-Box Testing, Verification, Validation, SPF.

I. INTRODUCTION

Software is a collection of commands used to operate computers to execute the specific job or operations. Software engineering is an application of mathematical and science principles used to design, development and maintenance of software. Testing is an integral part of any Development Methodology, many companies use the term Development Methodologies & Testing Methodologies colloquially. Software Testing is a process of executing a program with the aim of finding error. To make our software perform well it should be error free. If testing is done successfully, it will remove all the errors from the software.

II. SOFTWARE PROCESS FRAMEWORK

Framework is a Standard way to build and deploy applications. Software Process Framework (SPF) is a foundation of complete software engineering process. Any standard software process model would primarily consist of two types of activities [8][9][10]: A set of framework activities, which are always applicable, regardless of the project type, and a set of umbrella activities, which are the non SDLC activities that span across the entire software development life cycle. A software common process framework is shown in Figure1.

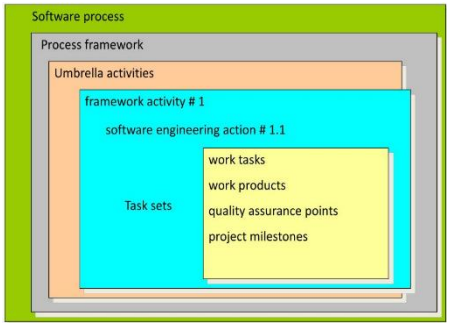

Figure1. Software Process Framework

A. Process Framework Activities

Five activities are included in Process framework, which is briefly explained below.

1) **Communication:** Communication is an initial activity in SPF. This activity deals with overall communication, requirements gathering with customer and other stakeholders.
2) **Planning:** In this activity, technical related tasks, work schedule, risks, required resources etc. are analysed.
3) **Modelling:** Modelling means building symbols/images of things in the real world. In this activity, we produce relevant models for both customers and developers including designing, requirement with respect to development.
4) **Construction:** Construction is the application of group of procedures that are required to assemble the product in software engineering. In this activity, we generate the code and test the product in order to make better product.
5) **Deployment:** The product is handover to the client who has checked it thoroughly and supply the feedback depends on its product. I its required we modify the products based on their feedback for provide better product.

B. Umbrella Activities

Software engineering is a collection of co-related steps. These steps are presented or accessed in different approaches in different software process models. An umbrella is portable, lightweight and easily transportable from place to place. Like each of these umbrella activities is defined by a set of tasks that are adapted to the project type and degree of rigor with which software engineering is to be applied [8][9][10]. These steps of umbrella activities will evolve through the phases of generic view of software development. Umbrella activities are shown in Figure2.

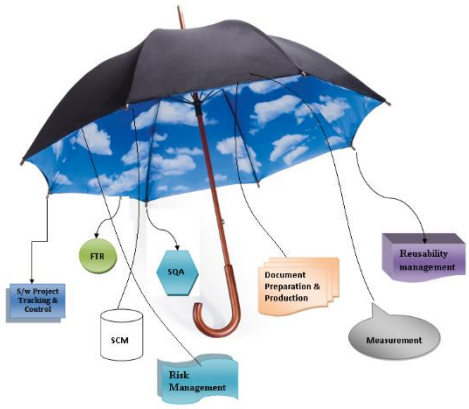

Figure2. Umbrella Activities

1) Software project tracking and control: The developing team accesses project plan and compare it with the predefined schedule is done in this activity. If it's not match with the predefined schedule, then the necessary actions are taken to keep the schedule.
2) Formal Technical Reviews (FTR): FTR is a meeting conducted by the technical staff. The intention of this meeting is to identify quality troubles and propose improvements.
3) Software Quality Assurance (SQA): SQA is well-planned and systematic pattern of activities, which are mandatory to give an assurance of software quality. For example, during the software development meetings are conducted at every stage of development to find out the defects and suggest improvements to produce good quality software.
4) Software Configuration Management (SCM): It manages the effect of change throughout the software process.
5) Document preparation and production: It consists of the activities that are required to create the documents, forms, lists, logs and user manuals for developing software.
6) Reusability management: It defines the criteria for reuse the product. The quality of software is good when the components of the software are

developed for certain application and are useful for developing other applications.

7) Measurement and Metrics: Measurement consists of the effort required to measure the software. The software is measured by direct and indirect. Cost, lines of code, size of software etc. are direct measures. Quality of software, which is measured by some other factor. Hence, it is an indirect measure of software.
8) Risk management: Risk is an event that may or may not occur. If the event occurs, then it causes some unwanted outcome. Hence, proper risk management is required.

III. SOFTWARE TESTING

Testing is the process of evaluating a system to ensure whether specified requirements are satisfied or not. Testing can be divided into two steps [3] shown in Figure3.

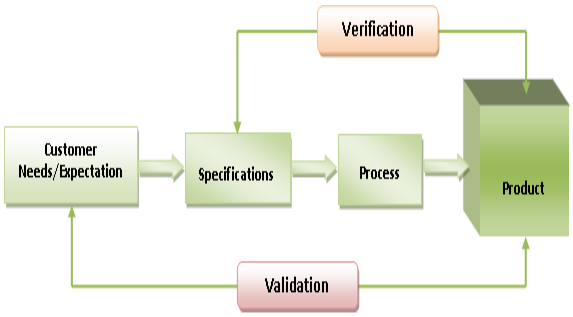

Figure3. Software Verification and Validation

1) Verification: It refers to the set of tasks that ensure that software correctly implements a specific function. "Are we building the product right?"
2) Validation: It refers to a different set of tasks that ensure that the software that has been built is traceable to customer requirements. "Are we building the right product?

A. Principles of Testing

A principle is an absolute rule or method for application in action that has to be. General guidelines are recommending by the Testing Principles for all testing, which helps us in performing testing successfully and professionally. Software testing principles [4][7][10] are:

1) Testing shows presence of defects: As we, all know that no Product is 100% defect free. Testing can show defects are present but cannot prove that there are no defects.
2) Exhaustive Testing is impossible: To understand this better, for software, which deals with a large combination of data it, is not possible to test all the data combination in a set time. Hence, Exhaustive testing is impossible. However, we would opt for testing the application based on Boundary Value and Equivalence partitioning and cover possible combination.
3) Early Testing: The early we detect the error/ defect, the less expensive. Hence, it becomes more important for us to detect gaps/issues in the early phase of the Software Life cycle.
4) Defect Clustering: A small number of modules contain most of the defects discovered in the pre-release testing or shows most operational failures. Sometimes a small module may contain most of the defects as compared with bigger module.
5) Pesticide Paradox: On executing the same test cases repeatedly, we might not be able to find new defects. Hence, we would have to modify the exiting test cases to fetch defects that we not captured earlier.
6) Testing is context dependent: Different sites/application will have to be tested differently. This might be due to the domain/ Functional Requirements/ Testing needs.
7) Absence of errors fallacy: This principle states that if the system built does not satisfy the user's

needs and requirements, then finding and fixing defects is of no use.

IV. TESTING TECHNIQUES

There are three testing methodologies, which are used for testing. They are White Box Testing, Black Box Testing, and Grey Box Testing [5]. These are also called as Testing Techniques. Each of the testing technique is briefed below for your better understanding.

A. Black Box Testing

The technique of testing without having any knowledge of the interior workings of the application is Black Box testing. When performing a black box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon [1][6]. Black Box Techniques include:

1) **Equivalence Partitioning:** Equivalence partitioning (EP) is a specification-based or black-box technique. EP is also known as equivalence classes. The idea behind this technique is to divide (partition) a set of test conditions into groups. In this technique, we need to test only one condition from each partition. This is because we are assuming that all the conditions in one partition will be treated in the same way by the software. If one condition in a partition works, we assume all of the conditions in that partition will work. Similarly, if one of the conditions in a partition does not work, then we assume that none of the conditions in that partition will work so again there is little point in testing any more in that partition.
2) **Boundary Value analysis:** Boundary testing is the process of testing between extreme ends or boundaries between partitions of the input values.
3) **Decision Tables:** Decision table testing is a software testing technique used to test system behavior for different input combinations. This is a systematic approach where the different input combinations and their corresponding system behavior (Output) are captured in a tabular form. That is why it is also called as a Cause-Effect table where Cause and effects are captured for better test coverage. A Decision Table is a tabular representation of inputs versus rules/cases/test conditions.
4) **State Transition Testing:** State Transition testing is defined, as the software testing technique in which changes in input conditions cause's state changes in the Application under Test (AUT). It is a black box testing technique in which the tester analyses the behavior of an application under test for different input conditions in a sequence. In this technique, tester provides both positive and negative input test values and record the system behavior.

Black-box testing approach is shown in Figure4. TABLE1 shows the advantages and disadvantages of Black-box testing.

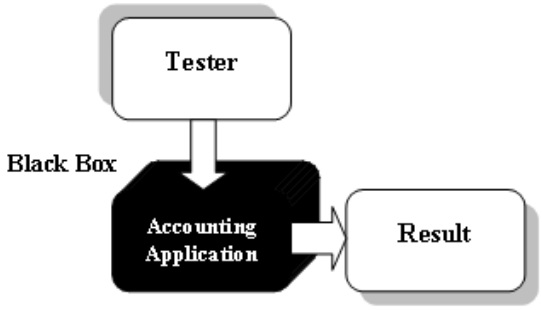

Figure4. Black Box Testing Approach

TABLE 1
Advantages and Disadvantages of Black-box Testing

Sl. No	Advantages	Disadvantages

1.	Testers no need to know the programming language of the application	Only some random paths are covered
2.	Time period is short for tests preparation	It's not possible to predict what part of the code can cause a problem in future
3.	After completing the specification only test engineers starts the test case design	Design test cases without a specifications are very difficult
4.	Test engineers can notice various requirement challenges	Lack of information for detailed testing

B. White Box Testing

White box testing is the detailed investigation of internal logic and structure of the code. White box testing is also called glass testing or open box testing [1][6]. In order to perform white box testing on an application, the tester needs to possess knowledge of the internal working of the code. White-box testing approach is shown in figure 5. Table 2 shows the advantages and disadvantages of White-box testing. These Testing Techniques include [2]:

1) **Statement Coverage:** It is also known as line coverage or segment coverage. Statement Coverage examines all the programming statements. The statement coverage covers only the true conditions. Through statement coverage, we can identify the statements executed and where the code is not executed because of blockage. In this process, each and every line of code needs to be checked and executed.
2) **Branch Coverage:** Branch coverage is also known as Decision coverage or all-edges coverage. It covers both the true and false conditions unlikely the statement coverage. A branch is the outcome of a decision, so branch coverage simply measures which decision outcomes have been tested.
3) **Path Coverage:** Basis path testing involves execution of all possible blocks in a program and achieves maximum path coverage with the least number of test cases. It is a hybrid of branch testing and path testing methods.

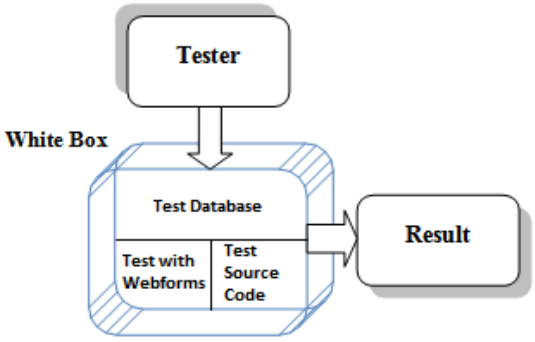

Figure5. White-box Testing Approach

TABLE II
Advantages and Disadvantages of White-box Testing

Sl. No	Advantages	Disadvantages
1.	Testing can be performed in the initial stages of the software lifecycle	Test engineers must familiar with programming language of the software
2.	No need to wait for the GUI implementation	Takes much more time than other approaches
3.	It is possible to conduct more detailed testing with more cases covered	Can be complex and expensive
4.	It is possible to	

	anticipate the potential problems	

C. Grey Box Testing

Grey-box testing is a technique to test the application with limited knowledge of the internal workings of an application. In grey box testing, the tester has access to design documents and the database [1]. Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan [6]. Grey-box testing approach is shown in Figure6.

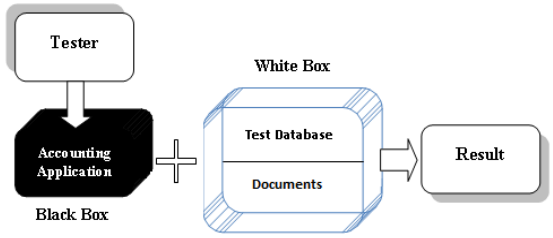

Figure6. Grey-box Testing Approach

The techniques used for grey box testing are:

1) **Matrix Testing:** This testing technique involves defining all the variables that exist in their programs.
2) **Regression Testing:** Testing the software after every change in the software to make sure that the changes or the new functionalities are not affecting the existing functioning of the system. This type of testing is also done when any defect is fixed to ensure that fixing defect has not affected other functionality of the software.
3) **Orthogonal Array Testing (OAT):** it is a black-box testing method. OAT is preferred when maximum coverage is required when there are very few test cases and test data is large. This is very helpful in testing complex applications.
4) **Pattern Testing:** This testing is performed on the historical data of the previous system defects. Unlike black box testing, gray box testing digs within the code and determines why the failure happened.

The grey box testing incorporates advantages and disadvantages of white-box and black-box techniques and can be a good compromise in the question which of the techniques to choose.

V. TESTING TECHNIQUES COMPARISON

The comparison between black-box, white-box and grey-box testing is shown in table 3.

TABLE III
Comparison between testing techniques

Sl. No	Black-Box Testing	White-Box Testing	Grey-Box Testing
1	The internal working knowledge of applications is not required	Testers have full knowledge about internal working of applications	Some knowledge about internal working of applications
2	Also called as closed box testing, data driven testing and functional testing	Also called as structural testing, code based testing and clear box testing	Also called translucent testing
3	Performed by Testers, developers and end-users	Performed by Testers and developers	Performed by Testers, developers and end-users

4	Testing is based on external expectations like functionality	Internal workings are fully known and the tester can design test data accordingly	Testing is done on the basis of high level database diagrams and data flow diagrams
5	Not suited to algorithm testing	Suited for algorithm testing	Not suited to algorithm testing

VI. CONCLUSION

Software process framework and umbrella activities are helpful to develop standard software. Testing is also required to produce and maintain the quality product. Software companies are using two different types of testing, namely Manual Testing and Automation Testing. In this paper we explained the activities of software process framework and examine the comparisons of software testing techniques.

VII. REFERENCES

[1] Dr. L. C. Manikandan, Dr. R. K. Selvakumar, "Software Testing Life Cycle and Approaches for Fault Finding", International Journal of Research and Analytical Reviews, Vo.6, Issue.2, June 2019.

[2] Himanshi Babbar, "Software Testing: Techniques and Test Cases", International Journal of Research in Computer Applications and Robotics, Vol.5, PP.44-53, March 2017.

[3] Prasad Mahajan, "Different Types of Testing in Software Testing", International Research Journal of Engineering and Technology, Vol.3, Issue.4, April 2016.

[4] Anupriya, Ajeta, "Software Testing - Principles, Lifecycle, Limitations and Methods", International Journal of Science and Research, Vol.3, Issue.10, pp. 1000-1002, October 2014.

[5] Abhijit A. Sawant, Pranit H. Bari, P.M.Chawan, "Software Testing Techniques and Strategies", International Journal of Engineering Research and Applications, Vol.2, Issue.3, June 2012.

[6] Mohd. Ehmer Khan, "Different Forms of Software Testing Techniques for Finding Errors", International Journal of Computer Science Issues, Vol.7, No.1, May 2010.

[7] S.M.K Quadri, Sheikh Umar Farooq, "Software Testing – Goals, Principles and Limitations", International Journal of Computer Applications, Vol.6, No.9, September 2010.

[8] Roger S. Pressman, "Software Engineering: A Practitioner's Approach", 7th edition, 2009.

[9] Jovanovic and Irena, "Software Testing Methods and Techniques", May 2008.

[10] Ian Somerville, "Software Engineering", Addison Wesley, 2001.

VIII. AUTHOR BIOGRAPHIES

[1]**Dr.L.C.Manikandan** is working as Professor & Head, Dept. of CSE, at Musaliar College of Engineering and Technology, Pathanamthitta, Kerala, INDIA. He has received Ph.D. degree in Computer and Information Technology from Manonmaniam Sundaranar University, M.Tech. Degree in Computer and Information Technology from Manonmaniam Sundaranar University, M.Sc., Degree in Computer Science from Bharathidasan University and B.Sc. Degree in Computer Science from Manonmaniam Sundaranar University. His main research interest includes Video Surveillance, Image Compression & Video Coding in image processing.

[2]**Dr.R.K.Selvakumar** received his Ph.D. degree in Computer Engineering from Manonmaniam Sundaranar University, India. He has completed the Master degrees M.Sc.(Maths), DOEACC-C(CT), M.C.A., M.Phil.(CS) and M.Tech.(C&IT). He is working as

Professor at CVR College of Engineering, Hyderabad, Telungana, INDIA. He carries out research in Digital Image Processing, Video Surveillance, Video coding, Digital Watermarking and Soft Computing.

Cite this article as :

Dr. L C Manikandan, Dr. R K Selvakumar, "A Study on Software Process Framework and Testing Techniques", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 4, pp. 94-101, July-August 2019.
Available at doi : https://doi.org/10.32628/CSEIT195413
Journal URL : http://ijsrcseit.com/CSEIT195413