Implementing Different Types and Variants for Software Testing Process and Techniques

N. Sudheer[1*], V. Sarma[2] , N. Ahmad[3]

[1*]Department of CSE, Siddharth Institute of Engineering & Technology, Puttur, India
[2] Department of Computer Engineering, Professor in Aligarh Muslim University, Aligarh, India
[3] Department of Computer Engineering, Professor in Aligarh Muslim University, Aligarh, India

Corresponding Author: nidamanuri.sudheer@gmail.com

www.ijcseonline.org

Abstract— Software testing is the most critical phase of the Software Development Life Cycle. Software under test goes through various phases, which as per the study are test analysis; test planning, test case/data/environment preparation, test execution, bug logging and tracking and closure. There is lot of research which has been done in past to optimize overall testing process with intent of improving quality of software in a minimum amount of time. After evaluating all available testing processes it has been found that different development models are used for different types of applications and different testing techniques are performed to test the same. Based on the research during the study of this paper, it has been analyzed that each company modifies their testing process as per the needs and performs testing based on the criticality of the applications. The most critical components of each application have to be tested thoroughly to ensure their functional, performance and security features are behaving as expected. This paper talks about ensuring the quality of all types of software applications by performing certain types of testing techniques and optimized software testing processes. As per the study and research done testing types can be categorized under three major testing techniques which are Functional, Performance and Security Testing and major software testing process called as Analysis, Preparation and Execution and closure.

Keywords— Functional, Performance and Security Testing (FPS), Analysis, Planning and Preparation, Execution and Closure (APEC),), Software Testing Techniques, Software Testing Life Cycle (STLC), Software Development Life Cycle(SDLC).

1. INTRODUCTION

Software testing is the main activity of evaluating and executing software with a view to find out errors. It is the process where the system requirements and system components are exercised and evaluated manually or by using automation tools to find out whether the system is satisfying the specified requirements and the differences between expected and actual results are determined. This paper at a high - level is divided into two sections. The first section covers optimized testing process, which elaborates all phases of the testing life cycle and the second section covers testing types. The first section emphasizes the main activities, which are Analysis [A], Planning and Preparation [P], Execution [E] and Closure[C]. Where closure includes release and root cause analysis activities and execution phase goes hand in hand with bug logging and tracking.

The software bug life cycle explained in the paper in the coming section highlights the mandatory steps for bug logging and tracking. The test preparation phase includes test case preparation, test case selection, test case optimization and test data preparation which is going to be elaborated

later in this paper. There are lots of available testing types like black box testing, white box testing, state based testing, security testing, look and feel testing, acceptance testing, system testing, alpha and beta testing, and configuration based testing, verification and validation testing. Based on the research and study done this paper categorized all of them under three high - level testing types, which is Functional, Performance and Security (FPS).The last section deals with the conclusion, which shows relevance of our optimized software testing process and FPS as a basis for testing methods.

2. OPTIMIZED SOFTWARE TESTING PROCESS

STLC phases deals with detecting and rectifying any error by using various software testing techniques. This paper presents the required phases of testing lifecycle without which no software life cycle would be completed efficiently. Testing basically furnishes a criticism or a comparison that determines the state behavior of the system against its specifications, mechanisms, principles, characteristics and relevant standards. Software testing process can be customized according to the customer or the project needs.

The optimization process which one can use while testing software is analysis, planning and preparation, execution and closure. The software process provides the flow of the system and enhances the assurance of the product to be produced. There are various methods of testing of software that can be referred from different research journals, books and published papers but based on study, research and considering all the critical testing types, this paper talk about the key findings that Functionality, Performance and Security testing are three main software methods that a software tester needs to be tested to provide software according to specifications and with good quality.

2.1. Test Analysis Phase

The first phase which is a Analysis phase is the basic phase of the software testing process. This phase includes the analysis of functional and non functional requirements e.g. business requirements, functional specification document and technical specification document etc.

The requirements collection and is to be done for elucidation with customers to identify actual and expected results of testing like Identification of requirements and gaps, which are basically non functional requirements such as usability, scalability, testability, maintainability, performance and security. All requirements that cannot be tested due to system and test environment constraints should be communicated to the business team. During this phase, the testing team reviews and analyses the requirements and identifies the tests, which are to be performed and sets priorities for testing - team members. The test environment requirement includes the hardware and software requirements under which the required software is to be tested and in parallel software developers start by planning and development activities.

2.2. Test Planning and Preparation Phase

The test preparation phase includes test plan preparation, test case, test data and test environment preparation. The test plan is the first document to be prepared, which outlines the scope, objectives, features to be tested, features not to be tested, types of testing to be performed, roles and responsibilities of testing team, entry and exit criteria and assumptions[1-10]. Simultaneously the testing teams start preparing test cases and test data. A test case is a document, which outlines steps required to test any functionality with expected and actual result. If actual result doesn't matches with expected result, then a bug is opened. For each requirement, positive and negative test cases are prepared, which is ensured by requirement traceability matrix (RTM). RTM is a document which maps requirements with test cases to ensure 100% testing is done

All valid and invalid test data sets are to be prepared for each test case and a test data document is prepared. Test data is

also generated based on some algorithm and tools. Test case preparation [11] has various steps which start with Test case generation [12], Test case selection [16], Evaluation, and Test case prioritization [5]. There are various algorithms which are used to generate and optimize test cases.

Swain et.al proposed a technique to generate test cases using corresponding sequence diagrams and also specifies the constraints across the defined artifacts. At the same time test case generation techniques are helpful for detecting synchronization and dependency of use cases and messages, object interaction and operation faults [12].

Test environment preparation is one of the most important phases which are usually prepared by separate team handling environments. After completion of coding part, the code is checked by configuration management tool and then test build is prepared where testers have to start test execution.

2.3. Test Execution Phase

In this phase testers execute software as per test cases. Wherever actual and expected results don't match then tester open bugs and assign the same to developers. Bug logging and tracking [13] follows complete life cycle of bug. There is already a lot of work which has been done in past that focuses on main steps to be taken to report valid fault. The routine reports can be discussed on weekly/ daily basis along with the projects progress on project delivery, acceptance and approvals are monitored to analyze pilot project.

2.4. Test Closure

Test Closure is a important phase which includes all test reports ensuring that all system, integration, user acceptance testing passed and decision is taken whether all requirements are tested and there is no critical bug pending to be fixed OR verified. A review of all test artifacts is done by Manager. Once all artifacts are reviewed and approved then software release is done. Further root cause analysis is being done to brainstorm on what went well, what did not go well and areas of improvement. There are various root cause analysis tools and methods available on which a lots of research has been done in past.

3. SOFTWARE TESTING TYPES

There are various software testing techniques as per the research and study like black box, white box, grey box[19][21], regression [22], reliability, usability, performance, unit, system, integration, security, smoke, sanity and object oriented testing etc. It is impossible to perform all types of testing on a software as there is always fixed amount of time allocated for testing. Functional testing is very common and lots of research is done on them in past that's why only in rare cases a site crashes due to lack of functional testing. The most recent failures

happened in past are due to lack of Performance and Security testing. In 2014 Indian Railway site got crashed as it was not able to handle load of customers. Another failure in 2014 is of Delhi University (DU) online application form web site crash on last day of submission due to excessive load on site. Then there were instances in 2013 when Indian government sites were hacked by some external agencies. After analyzing and survey of all these techniques it is found that a right mix of testing types should be performed on a given software to ensure quality and overall reliable software. This paper will focus on the main testing techniques like Functional [F], Performance [P] and Security testing[S]. The right mix of testing should be included from all headers of F, P and S. Functionality is first and foremost aspect of software testing which ensure quality of software.

Verification and Validation is done using Static and Dynamic testing respectively. Static testing involves all types of reviews, inspections, and walkthroughs. Dynamic testing or actual validation involves all functional and non-functional testing types.

3.1. Functional Testing

The main quality factor in software is to meet its required functionality and behavior. The functional part of software includes the external behavior that mainly specifies all user requirements. The high level design of the software is produced so that the customer would be satisfied at an early stage of design and development. The functional testing revolves around the basic work flows and alternative flows of software. These flows can be represented by various use case diagrams like sequential diagrams, class diagrams, component diagrams etc [9][10]. Automated Test cases are also generated by UML models [8]. There are different types of functional testing methods and techniques [18] which could be performed at various levels of testing i.e. unit testing, integration testing (top down and bottom up testing) and system testing. There are lots of testing performed at various levels of testing like black box testing[19], white box testing, grey box testing [21] regression testing [22][2], fuzz testing, use case testing, exploratory testing, smoke testing, sanity testing, acceptance testing [23], alpha, beta testing etc. Test cases are built around specifications and requirements i.e., what the application is supposed to do. The functional testing method basically focuses on "What" is supposed to do but not on "How".

Unit testing is usually done by developers. Integration testing and System testing is performed by testing team and user acceptance testing is mainly performed by end users or business team.

The main types of Functional Testing are explained in brief in below table 1:

S.No.	Testing Type	Definition
1	Unit Testing	The lowest level of testing mainly performed by developer to test the unit of code
2	Integration Testing	This is to test the communication between various modules to make sure data is flowing across various components correctly. done following either top-down approachOR approach.
3	System Testing	The overall system is tested to ensure that it is behaving or functioning as intended and as specified in requirement document. Regression performed to ensure that nothing is broken in system after fixing bugs and testing bugs. Overall Smoke and Sanity testing is performed to ensure all links and features are working environment is stable.
4	Acceptance	Pre acceptance testing performed mainly known as alpha and beta testing to ensure the customers are able to perform intended functionality and feedback is taken to further enhance quality of software.
5	White box and Black Box Testing	Black box testing is performed to ensure output of application is as correct for all various types of positive and negative inputs. There are various types of Black box testing types like Equivalence Class partitioning, Boundary value analysis, error guessing etc. White box deals with internal working of code to ensure there is no redundant code written in software. This involves testing of line of code, program, flow, logic, loop, structure, functions, communication testing and ther internal testing of program.

3.2. Performance Testing

This is one of non-functional testing types which test performance of software under all favorable and non-favorable conditions. This includes all time related parameters like Load time, access time, run time, execution time etc. This also includes success rate, failure frequency, mean time between failures and overall reliability of software. The most popular types of testing performed in Performance testing[14] are Stress testing and Load testing. Stress testing is performed to find and understand the upper

limits of capacity within the system. Extreme load is given to the application to determine the robustness of the system. Soak testing is also performed which is called as endurance testing. This testing is done to determine if the system can sustain the continuous expected load. Potential leaks are detected by monitoring the memory utilization.

Soak testing ensures that the throughput and response time after some long period of sustained activity are as good as or better than at the beginning of the test. The main goal of performing soak test is to discover the system behavior under its sustained use. Mean Time To Repair (MTTR) and Mean Time Between Failures (MTBF) are calculated for the system efficiency and robustness.

Spike testing is done by suddenly increasing the number of users or load generated by users by a very large amount and observing the behavior of the system. Mainly tools are used to test performance of software, as it is very difficult to test the load manually. There are lots of freeware tools like Soap UI and Jmeter used to test performance of software. The most popular tool used to test performance of software is Load runner tool and there are various IBM tools also available to test the performance of software [15]. Most of the recent failure occurred in software industry are due to lack of performance testing.

3.3. Security Testing

As per the current scenarios timing and buffer overflow attacks are most common. In object oriented systems, design level problems include error handling. Some other design level problems like sharing and trust issues, unprotected data channels, incorrect or missing access control mechanisms, lack of auditing, incorrect logging and timings and ordering errors also lead to security risks. The software is required to test for the security features like strong authentication, cryptography and access control and some other security mechanisms.

[Gary McGraw] suggests that vulnerability is an error that an attacker can exploit [6]. The system is tested for areas like authentication, authorization and different kinds of threats. Security testing of software is important as to protect the information, services, skills and resources of adversaries and the cost of potential assurance remedies [15].

Security testing basically follows two types of approaches:
a. Testing software regarding software's functional mechanisms
b. Performing risk based approach according to attackers mindset.
Penetration Testing is a security testing in which evaluators attempt to circumvent the security features of a system based on their understanding of the system design and implementation.

Also a other type which is Fuzz Testing was given by Barton Miller, university of Wisconsin in 1988. It is software testing technique in which automatically invalid, random and unexpected data to the software is given to find out the reaction of software. It is good for testing that software where inputs have no control over the predefined data. This testing technique is only used to find only the simple features of the software but not the complex software code.

Now to sum up on the above sections of the testing, Table 2 below categorizes all Testing types

Table 2

Testing Types	Methods
1.Functional Testing	Black box, White box, all pair testing, state transition tables, decision tables, model based testing, use based testing, exploratory testing, specification based testing, regression testing, smoke testing, sanity testing
2.Performance Testing	Load, Spike, Stress, Soak and configuration testing.
3.Security Testing	Static Analyzer, Brute Force Attack, SQL Injection and Cross Site Scripting (XSS), Penetration testing and Fuzz testing.

4. MAPPING OF SOFTWARE TEST PROCESS AND TESTING TYPES

There are four major phases in software test life cycle which are earlier explained in this paper. Each phase require some sort of testing to be performed. In test analysis and test preparation phase only verification of requirement documents and other test documents is done. Verification involves all types of reviews, inspection and walkthroughs. It is mainly done before validation. Once Test Preparation is done and all artifacts are reviewed and base lined, then Test execution begins, where in actual validation is performed. In this phase all types of testing is performed which are shown in Figure 1 below at a high level.
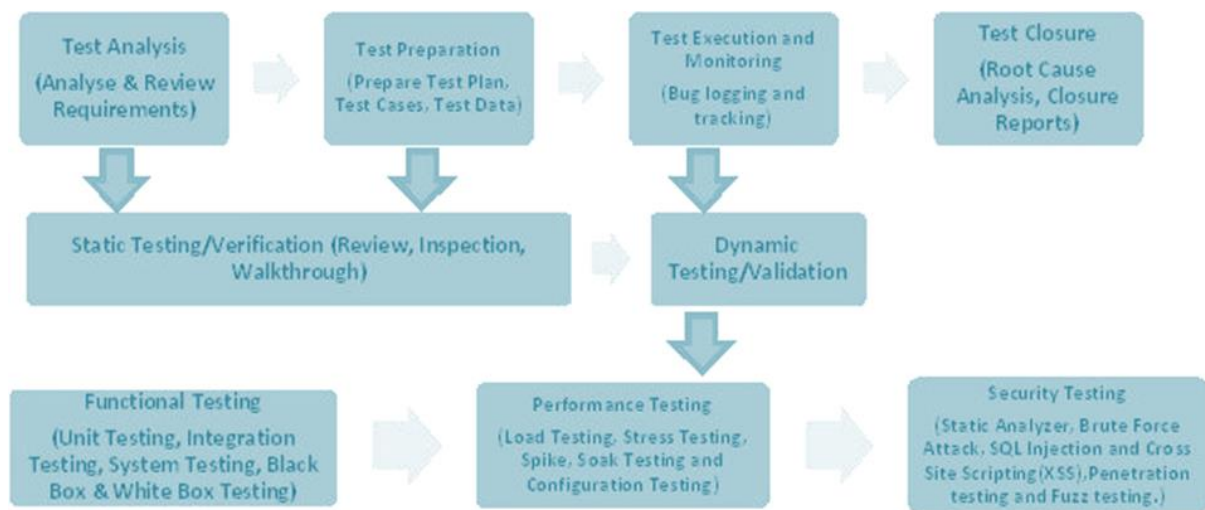
Figure 1: Mapping of Software Test Phases and Testing Types

5.   CONCLUSION

The intent of this paper was to research on various phases of software testing life cycle and different types of testing. After reviewing various phases of software life cycle it is found that there are main 4 phases in testing life cycle that could be categorized as Analysis, Planning and Preparation, Execution and Closure. A generic software testing life cycle- APEC is proposed in this paper. Also most recent failures are studied, which happened due to lack of performance and security testing. A lot of time is spent on Functional testing and there is rarely any software which got crashed due to lack of functional testing in recent past. So this paper proposed a new right mix of testing which should include some performance and security testing checks in addition to functionality testing for better quality of software. As there is always a scope so Further to this paper a research and study can be done on the software testing to propose a generic testing framework and techniques to support functional, performance and security testing for object oriented development framework and other platforms using some algorithm(s) with/ without use of tools in minimum amount of time.

6.   REFERENCES

[1]. SK. Swain, DP. Mohapatra, R. Mall, "*Test Case Generation Based on Use Case and Sequence Diagram*", International Journal of Software Engineering, Vol.7, Issue.5, pp.289-321, 2010.

[2]. A. Babu, K. Kolluri, L. Tameezuddin, D. Guddika, "*Effective Bug Tracking Systems.Theories andImplementation*", IOSR Journal of Computer Engineering, Vol. 4, Issue.6, pp.31-36, 2012.

[3] G. McGraw "*Software Security Testing .IEEE Security and Privacy.2,2*" IJDR ,Vol.5, Issue.8, pp. 80-83, 2004.

[4] D,Verndon. G.McGraw.2004,"*Risk Analysis in software Design. IEEE Security and Privacy.2,4*", Vol.3, Issue.5, pp.32-37, 2004.

[5] PN. Boghdady, L. Badr, M. Hashem, Tolba, "*A Proposed Test Case Generation Technique based on Activity Diagrams*", IJET-IJENS, Vol.3, Issue.7, pp.37-57, 2011.

[6] B. Nirpal, P. Kale, "*Using Genetic Algorithm for   Automated Efficient Software Test Case Generation for Path   Testing*", Int. J. Advanced Networking and Applications, Vol.2, Issue.6, pp.911-915, 2011.

[7] K.K. Mohbey, G.S. Thakur , "*A Clustering Framework for Large Document Datasets*", International Journal of Computer Sciences and Engineering, Vol.1, Issue.1, pp.26-30, 2013.

[8] C.P. Patidar, "*Test Case Generation Using Discrete Particle Swarm Optimization Algorithm*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.1, pp.38-42, 2013.

[9] A. Sheeba, A. Chandrasekar, P. Veerasingam, "*Semantically enriched OWL-S Files for Mathematical Web Services*", International Journal of Computer Sciences and Engineering, Vol.1, Issue.2, pp.1-4, 2013.

[10] N.K. Rao, R.M. Reddy, B.K. Rao, "*Defect Prediction in Software Entities Classified in Terms of Level Dependencies*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.1, pp.20-25, 2013.

[11] P. Padmavathy, C. Balakrishnan, "*Smart Tracking of Human Location and Events Based on WPS using Android Technology*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.1, pp.30-34, 2014.

[12] Aanchal, S. kumar, "*Metrics for Software Components in Object Oriented Environments: A Survey*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.2, pp.25-29, 2013.

[13] MidhyaMohan E, Filma Mathew, Glemin George, Harikrishnan V, Vaneza Benny, Bineesh M, "*Lab Tracer- The Remote Desktop Technology*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.10, pp.62-66, 2014.

[14] Chandraprakash Patidar, "*A Report on Latest Software Testing Techniques and Tools*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.4, pp.50-52, 2013.

[15] Latika Kharb, "*CCTF: Component Certification & Trust Framework*", International Journal of Scientific Research in Computer Science and Engineering, Vol.1, Issue.6, pp.4-6, 2013.

[16] RB. Pansare, I.R.Shaikh, "*Min-Max and Limited Knowledge Algorith mic Approach for Load Balancing*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.11, pp.50-59, 2014.

[17] V.O. Ikenna, C.B. Asiegbu, N.E. Gloria, C.N. Obi, "*Group Membership Prediction of an Outsourced Software Project: A Discriminant Function Analysis Approach*", International Journal of Scientific Research in Multidisciplinary Studies , Vol.3, Issue.4, pp.12-18, 2017.

[18] Poreddy Jayaraju, Vijay Prakash, "*Web Based Interface Implementation*", International Journal of Scientific Research in Computer Science and Engineering, Vol.3, Issue.5, pp.6-11, 2015.

[19] A. Mallick, N. Panda, A.A. Acharya, "*Generation of Testcases from UML Sequence Diagram and Detecting Deadlocks using Loop Detection Algorithm*", International Journal of Computer Sciences and Engineering, Vol.2, Issue.3, pp.199-203, 2014.

[20] SM. Kalyankar, SD. Sawarkar, "*Design and Development of Reconfigurable Multiprocessor Architecture for Embedded Systems*", International Journal of Scientific Research in Network Security and Communication, Vol.5, Issue.1, pp.33-36, 2017.

[21] Hemant Malviya, Sita Solanki, "*Survey of Quality Models for Evaluating Quality in OSS (Open Source Software)*", International Journal of Scientific Research in Computer Science and Engineering, Vol.4, Issue.1, pp.15-20, 2016.

[22] M. Karanam, L. Gottemukkala, "*Software Fault Detection Using Improved Relief Detection Method*", International Journal of Scientific Research in Computer Science and Engineering, Vol.4, Issue.5, pp.1-4, 2016.

[23] M.B. Shinde, S.G. Tamhankar, "Review: Software Defined Networking and OpenFlow", International Journal of Scientific Research in Network Security and Communication, Vol.1, Issue.2, pp.18-20, 2013

AUTHORS PROFILE

Dr. N. Sudheer born in Ongole Prakasam Dt, AP. Presently working as an Associate professor (CSE) in Siddharth institute of Engg.& Technology ,Puttur and completed Ph.D (CSE), in Aligarh Muslim University, Aligarh, Uttar Pradesh, India. My Research interest includes Software Engineering and Testing Methodologies & Published number of Journals.
Email:nidamanuri.sudheer@gmail.com

Dr. V. Sharma born in Aligarh, UP. Worked as a Professor (CSE), in Aligarh Muslim University,Aligarh, Uttar- Pradesh, India. He Guided Number of Ph.D Scholars in Aligarh Muslim University & Published number of Journals. Research interest includes Software Engineering, Networks and Testing Methodologies & Published number of Journals.
Email: dr.vidhushisarma@rediffmail.com

Dr. Nesar Ahmad Presently Working as a Professor (CSE), in Aligarh Muslim University,Aligarh, Uttar- Pradesh, India. He Guided Number of Ph.D Scholars in Aligarh Muslim University & Published number of Journals. Research interest includes Software Engineering, Networks and Testing Methodologies & Published number of Journals.
Email: nesarahmad@gmail.com