RESEARCH ARTICLE OPEN ACCESS

Manual And Automation Method For Software Application Testing Process,Types,Techniques

Vamsi Krishna.V *, Karthikeyan.T **

* (M.Phil Computer Science Student Department of Computer Science
Dr.M.G.R Chockalingam Arts College Arni-632317,Tamilnadu,India)
**(Associate Professor Department of Computer Science
Dr.M.G.R Chockalingam Arts College Arni-632317,Tamilnadu,India)

--************************--

Abstract :

Software testing is the one of the critical phase in the Software Development Life Cycle. Software under test goes through different phases, which as per the study are test analysis, test planning, test case/data/environment preparation, test execution, bug logging, tracking and test closure. The research which has been done in past to optimize overall testing process with intent of improving quality of software in a short period. After verifying and validating all available testing processes it has been found that different development models are used for different types of software applications and different software testing techniques are performed to test the same concept. Based on the research during the survey of this paper, it has been analyzed that each company select their testing process as per the requirement and performs testing based on the critical environment 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 teach about ensuring the quality of all types of software applications by performing different types of testing techniques and optimized software testing processes. As per the survey and research done testing types can be followed 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.

Keyword - **Functional, Performance and Security Testing Analysis, Execution and Closure, Software Testing Techniques, Software Testing Life Cycle (STLC), Software Development Life Cycle(SDLC).**

--************************--

I. 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 , Planning and Preparation , Execution and Closure. 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. The last section deals with the conclusion, which shows relevance of our optimized software testing process and it as a basis for testing methods.

II. 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. 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.

III. 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.

IV. 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. 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 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 has various steps which start with Test case generation , Test case selection , Evaluation, and Test case prioritization. There are various algorithms which are used to generate and optimize test cases. 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. 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. 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 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. 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.

V. SOFTWARE TESTING TYPES

There are various software testing techniques as per the research and study like black box, white box, grey box, regression, 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.

VI. FUNCTIONAL TESTING

Functional testing is a software testing process used within software development in which software is tested to ensure that it conforms to all requirements. Functional testing is a way of checking software to ensure that it has all the required functionality that's specified within its functional requirements. Functional testing is primarily is used to verify that a piece of software is providing the same output as required by the end-user or business. Typically, functional testing involves evaluating and comparing each software function with the business requirements. Software is tested by providing it with some related input so that the output can be evaluated to see how it conforms, relates or varies compared to its base requirements. Moreover, functional testing also checks the software for usability, such as by ensuring that the navigational functions are working as required. Some functional testing techniques include smoke testing, white box testing, black box testing, unit testing and user acceptance testing.

VII. UNIT TESTING

Unit testing refers to tests that verify the functionality of a specific section of code, usually at the function level. In an object-oriented environment, this is usually at the class level, and the minimal unit tests include the constructors and destructors. These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to ensure that the building blocks of the software work independently from each other.Unit testing is a software development process that involves a synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. It is performed by the software developer or engineer during the construction phase of the software development lifecycle. Unit testing aims to eliminate construction errors before code is promoted to additional testing; this strategy is intended to increase the quality of the resulting software as well as the efficiency of the overall development process.Depending on the organization's expectations for software development, unit testing might include static code analysis, data-flow analysis, metrics analysis, peer code reviews, code coverage analysis and other software testing practices.

VIII. INTEGRATION TESTING

Integration testing is any type of software testing that seeks to verify the interfaces between components against a software design. Software components may be integrated in an iterative way or all together ("big bang"). Normally the former is considered a better practice since it allows interface issues to be located more quickly and fixed.

Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

IX. COMPONENT INTERFACE TESTING

The practice of component interface testing can be used to check the handling of data passed between various units, or subsystem components, beyond full integration testing between those units. The data being passed can be considered as "message packets" and the range or data types can be checked, for data generated from one unit, and tested for validity before being passed into another unit. One option for interface testing is to keep a separate log file of data items being passed, often with a timestamp logged to allow analysis of thousands of cases of data passed between units for days or weeks. Tests can include checking the handling of some extreme data values while other interface variables are passed as normal values. Unusual data values in an interface can help explain unexpected performance in the next unit. Component interface testing is a variation of black-box testing, with the focus on the data values beyond just the related actions of a subsystem component.

X. SYSTEM TESTING

System testing tests a completely integrated system to verify that the system meets its requirements. For example, a system test might involve testing a logon interface, then creating and editing an entry, plus sending or printing results, followed by summary processing or deletion (or archiving) of entries, then logoff.

XI. OPERATIONAL ACCEPTANCE TESTING

Operational acceptance is used to conduct operational readiness (pre-release) of a product, service or system as part of a quality management system. OAT is a common type of non-functional software testing, used mainly in software development and software maintenance projects.

XII.PERFORMANCE TESTING

Performance testing, a non-functional testing technique performed to determine the system parameters in terms of responsiveness and stability under various workload. Performance testing measures the quality attributes of the system, such as scalability, reliability and resource usage.

XIII. LOAD TESTING

It is the simplest form of testing conducted to understand the behavior of the system under a specific load. Load testing will result in measuring important business critical transactions and load on the database, application server, etc., are also monitored.

XIV. STRESS TESTING

It is performed to find the upper limit capacity of the system and also to determine how the system performs if the current load goes well above the expected maximum.

XV.SOAK TESTING

Soak Testing also known as endurance testing, is performed to determiQQne the system parameters under continuous expected load. During soak tests the parameters such as memory utilization is monitored to detect memory leaks or other performance issues. The main aim is to discover the system's performance under sustained use.

XVI. SPIKE TESTING

Spike testing is performed by increasing the number of users suddenly by a very large amount and measuring the performance of the system. The main aim is to determine whether the system will be able to sustain the workload.

XVI. SECURITY TESTING

It is a process intended to reveal flaws in the security mechanisms of an information system that protect data and maintain functionality as intended. Due to the logical limitations of security testing, passing security testing is not an indication that no flaws exist or that the system adequately satisfies the security requirements. Typical security requirements may include specific elements of confidentiality,integrity, authentication, availability, authorization and non-repudiation. Actual security requirements tested depend on the security requirements implemented by the system. Security testing as a term has a number of different meanings and can be completed in a number of different ways. As such a Security Taxonomy helps us to understand these different approaches and meanings by providing a base level to work from.

XVII. TEST EXECUTION

There are four different phases are explained in software test life cycle Each phase required various types of testing to be performed. In test analysis and test preparation phase only verifying the requirement documents. Verification involves all different types of reviews, inspection and walkthroughs. It is mainly perform before the validation. Once Test Preparation is perform and all artifacts are reviewed and base lined, then Test execution starts, where in actual validation is performed.

XVIII. CONCLUSION

The main goal of this paper was to research on different phases of STLC and various types of testing. After reviewing and analyzing different phases of software life cycle it is found that they are important four phases in testing life cycle that will be explained as Analysis, Planning and Preparation, Execution and Closure. software testing life cycle-APEC is proposed in this paper. Also most latest failures are learned, which happen due to lack of performance and security testing. maximum amount of time is work on Functional testing and there is rarely any software which got crashed and hanged due to lack of functional testing. So in this paper proposed a new process of testing which include both performance and security testing concepts in addition to functionality testing for to improve quality of software. Further to this paper deals with research and survey can be perform on the software testing to propose testing framework and techniques to support functional, performance and security testing for object oriented development framework and other different platforms using some algorithm(s) with/ without use of automation tools in minimum amount of time.

REFERENCES

[1] Pressman, R.S. 1997. Software Engineering: A practitioner Approach.4th Edition. Tata McGraw Hill.

[2] Sommerville, I. 1998. Software Engineering. 5th edition. Addison-Wesley.

[3] Myers, G.J. The Art of Software Testing. New York: John Wiley and Sons.

[4] McGraw.Chess,B. Seven pernicious kingdom.2005.A Taxonomy of Software Security Errors.WISSTWorkshop on Software Security Assurance Tools, Techniques and metrices.

[5] G.McGraw.2004.Software SecurityTesting .IEEE Security and Privacy.2,2(Sept/Oct 2004),(80-83).

[6] D,Verndon. G.McGraw.2004.Risk Analysis in software Design. IEEE Security and Privacy.2,4.32-37 (July/August 2004).

[7] Sarma, M. D,Kundu.Mall, R. 2007. Automatic Test Case Generation from UML Sequence Diagram. International Conference on Advance Computing and Communication.

[8] Bertolino, A. Basanieri, F. 2000. A Practical approach to UML-based derivation of integration tests. In Proceeding of the Fourth International Software Quality Week Europe and International Internet Quality Week Europe(QWE), Brussels, Belgium.

[9] Boghdady, P,N.Badr, ,L.Hashem, M.Tolba, M,.F. 2011. A Proposed Test Case Generation Technique based on Activity Diagrams.IJET-IJENS:11.No:3.(37-57)

[10] Treharne, H. Draper, J. Schneider, S. 2006. Test Case Preparation Using a Prototype. In: B'98: Recent Advances in the Development and Use of the B Method .Lecture Note in Computer Science.(1393,1998(293-311).(22May 2006).

[11] Swain,Kumar,Santosh, Mohapatra, Durga, Prasad Mall,Raj ib.2010.Test Case Generation Based on Use

case and Sequence Diagram. International Journal of Software Engineering (IJSE). Swain et al.3,2(July 2010). [12]Akhilesh, Babu, Kolluri.K, Tameezuddin. Kalpana, Guddika dula.2012.Effective Bug Tracking Systems.Theories and Implementation", IOSR Journal of Computer Engineering ISSN:2278-0661 Volume 4,Issue 6(Sept- Oct 2012), pp 31-36. [13] Rina, DCSK, KU, Haryana, INDIA, Tyagi, Sanjay.DCSA,Haryana.2013.AComparative Study Of Performance Testing Tools.IJARCSSE. 3,2(May 2013).

[14] Karen ,Scarfone.2012. Intro to Information Security esting&Assessment.ScarfonecyberSecurity Csr.nist.gov.(7June 2012).

[15] B,Beizer.1990. Software Testing Techniques. Technology Maturation and Research Strategies Carneige Mellon UniversityPittsburg,USA.

[16]B.Beizer 1995. Software Testing Techniques. 2006.VanNostrandReinhold,NewYork.1990.ISBN.0-442-20672- 0.(31.Oct.2006).

[17] A,Bertolino.2001.Chapter 5: Software Testing . IEEE SWEBOK trial version 1.00.IEEE(May 2001).

[18] Khan, Mohd. Khan, Farmeena. 2012. A Comparative Study of White Box, Black Box and Grey Box Testing Techniques.2012. International Journal of Advanced Computer Science and Applications(IJACSA). Vol. 3.No.6.(2012).

[19] Tarika,Bindia. Computer Programmer CSE,GNDEC,Ludhiana,Punjab-India.IJRITCC.2,1 .68-72.(2321-8169).

[20] B,Swarnendu.R,Mall.CSeDeptt,IIT Kgp.2011.Regression Test Selection Techniques, A Survey-Informatica 35 .2011.

[21] Swain,S.k.Mohapatra,D.P.Mall,R.2010.Terst Case Generation Based on Use Case and Sequence Diagram.International journal of Software Engineering (IJSE).3, 2.(July 2010),(289-321).

[22] Thakre, Sheetal. Chavan,savita. Chavan,P.M. 2012.Softwa re Testing Strategies and Techniques.International Journal of Emerging Technology and Advanced Engineering .Website: www.ijetae.com .2, 4.(April 2012), (2250-2459).

[23] An Approach to Cost Effective Regression Testing in Black-Box Testing Environment - IJCSI International Journal of Computer Science Issues. 8, 3, 1(May 2011),(1694-0814).

[24] Chauhan,Kumar,Vinod.2014.Smoke Testing-International Journal of Scientific and Research Publications4,2 (February 2014),(2250-3153).

[25] Gupta, Varuna.Sen, Saxena,Vivek.2013. Software Testing: Smoke and Sanity- International Journal of Engineering Research & Technology (IJERT).2,10(October 2013) (2278-0181).

[26] Liskin, olga.Hermann, christoph.Knauss, Eric.Kurpic, Tho mas.Rumpe, Bernhard.Schneida, Kurt.2012.Supporting Acceptance Testing in Distributed Software Projects with Integrated Feedback Systems: Experiences and Requirements. IEEE Seventh International Conference on Global Software Engineering.(2012).

[27] Yoo,Shin.Harman,mark2012.Regression Testing Minimisation, Selection and Prioritisation .A Survey. King's College London.Centre for Research on Evolution, Search &Testing.Strand, London, WC2R 2LS, UK.22,2 (March 2012) (67-120).http://onlinelibrary.wiley.com/resolve/doi?DOI=10.1002/stv.430

[28] Sumalatha,Mary.Raju,G.2013. Object Oriented Test Case Generation Technique using Genetic Algorithms.International Journal of Computer Applications(0975-8887). 61,20 (January 2013).

[29] Ostrand,T,J.Balcer,M, J.1988.The category-partition method for specifying and generating functional tests. Communications of the ACM 31 ,6(June 1998) (676- 686).doi>10.1145/62959.62964

[30] Nirpal,B,Premal.Kale,K,V.2011.Using Genetic Algorithm for Automated Efficient Software Test Case Generation for Path Testing. Int. J. Advanced Networking and Applications.(911-915).02,06 (January 2011).

[31] Malhotra,Ruchika.Garg,Mohit.2011. An Adequacy Based Test Data Generation Technique Using Genetic Algorithms. Journal of Information Processing Systems.7,.2, (June 2011).Doi:10.3745/JIPS.2011.7.2.363.

[32] Bhasin,Harsh.Khanna,Esha.Sudha.2014.Black Box Testing based on Requirement Analysis and Design Specifications International Journal of Computer Applications.(0975 – 8887) .18,18(February 2014).

[33] Mohamad Kassab ,"Software Testing Practices in Industry: The State of the Practice" 2016 IEEE

[34] Marcel Bohme and Soumya Paul,"A Probabilistic Analysis of the Efficiencyof Automated Software Testing", Ieee transactions on software engineering, vol. 42, no. 4, april 2016. Ieee transactions on software engineering, vol. 42, no. 4, april 2016.