A Serious Study on Software Testing

[1]Sabavath Raju, [2]Rupani Ranjith Kumar

[1,2] Department of Computer Science Engineering, Pallavi Engineering College, Hayathnagar, Kuntloor Village, Telangana, India.

Abstract – Software testing is the process of executing the software product or application with the intent of finding errors, defects, faults, and failures so that rectification mechanisms can be applied against errors, defects, faults, and failures for developing an application bug free or defect free by aiming to produce 100% bugfree software to the customer, testing team carries out various levels of testing. Software testing is a useful process to assess the quality of the software. The various facilities provided by software testing are i). Itreduces bugs in the software, decreases the cost of the software and aims at providing software with low maintenance cost. Butthe major challenge in testing software product is to find the appropriate test cases to test software. There are number of advanced methods in testing butstill the software has to be completely tested before it is deliveredto the customer. Therefore, many techniques and goals and objectives are involved in software testing. In this paper, various software techniques, process involved in software testing, SDLC (softwaredevelopment life cycle), the steps involved in SDLC, therole of software testing and its significance is explained indetail fashion.

Index Terms – Software testing, techniques in software testing, SDLC, test planning, test execution, test monitoring, software testing tools.

1. INTRODUCTION

With the aim of producing high quality software, software testing was initially introduced software testing is the process of detecting the error. There are basically two ways for testing the software products or applications i). Manualtesting and ii). Automation testing. The main objective of software testing is to produce a bug free software product. Software testing is usually done by software tester, developer, project manager and clients or end users. Software testing is done to ensure that thesoftware meets the specified requirements. Software quality assurance is provided by software testing. The primary goal of software testing is to detect possible number of defects at early stage. Softwaretesting is done at different stages in the development environment. It is to be discussed in brief. Verification and validation are important concepts in software testing. A simpleexample can explain both verification and validation.

Verification is the process of building the product right. Validation isthe process of building the right product.

2. STRATAGIES OF SOFTWARE TESTING

A technique in software testing is well planned and several stepsare followed. Thus forming a strategy for testing is important, the various types of software testing is done in the developmentenvironment or in place of the customer during maintenance. There are four levels in software testing.

Unit testing or Module testing

It is the lowest level of testing. Here a small portion or module of the software system or software application is tested. Each module is tested separately or individually. Thus, any defects at module level of the software system can easily be detected, unit testing is generally done by a developer, it is the simplest testing where each code is tested individually. It is just one of the parts in the testing cycle, unit testing is usually considered as the white box testing.

2.1. Integration testing

It is the amalgamation of unit testing together. In integration testing two or more modules are combined together and the process of testing is carried out as usually. Data flow is checked from one module toanother module in integration testing. Integration testing is usually performed by a tester, here two modules are combined and tested to ensure that the integrated module functions without defects.

2.2. System testing

In system testing, the whole or entire system is tested. All the components or modules are integrated and tested to check whether the specified requirements are met or not. It is an end to end process to ensure that the quality of the software. The functional requirementsof the software system are checked in system testing. This testing is usually takes place in the production or developer environment. Both the business requirements and application architecture are tested and validated here.

2.3. Acceptance testing

It is executed at the last stage of testing where the entire or whole software product is delivered to the customer and this testing is done in the user or client location or environment. The main objective of acceptance testing is to ensurethat the software works and functions properly as per the requirements of the client rather than finding defects.

3. PHASES OF SDLC

Software development life cycle is a process to be used to develop a software product. Requirements elicitation, requirements analysis, design, implementation or coding, testing, deployment and maintenance are the main phases to SDLC provides a blueprint for designing the code and creating software products with high quality in a specified time period be used in the software development process. Increasing the quality of the software. The various phases in SDLC are to be discussed one by one in a detailed fashion.

Figure: Software development Life Cycle

3.1. Requirements elicitation

The first phase in SDLC is Requirements elicitation. All the necessary requirements are collected or gathered by the developer as he usually interacts with the client in a formal way or approach.

3.2. Requirement analysis

The second phase in SDLC is requirement analysis, the requirements of the software are gathered together in requirement analysis, and customers have to tell the initial requirements. Then at the developer site requirements are analyzed in three aspects a). feasibility b).budget c). time period. The main aim of gathering requirements to make everyone get an overview of the project and to meet the scope of the project. Different methodologies are used in different phases of software development life cycle.

3.3. Design

The third phase in software development life cycle is design.In this phase technical architects and software developers forma blueprint and further provide a high-level design to be produced. In phase of design, various constraints are discussed with the client and important decisions are made. To implement the process further time, budget, potential of the team, variousproblems in the project are discussed and best design is framed.

3.4. Implementation

After the designing phase, the next phase is implementation. Actual coding is done in this phase. The team and development. Actual product is created by the development team in the implementation phase. In this phase software developers build complete code.

Testing is the last process in the software development life cycle. In this phase, testers start to test the software aim at producing bug free software. The main objective of testing is
to check whether the actual result matches with the expected result. In testing phase, test script is formed and different test cases are written and result is verified. There is a possibility offinding defects and the defects found are rectified by the development team, the process is continued until all the requirements have been satisfied and testing is done.

3.5. Deployment and Maintenance

After the testing stage, deployment and maintenance is done. There is no priority issue in this phase. The product is releasedand maintenance of the software is checked periodically at the customer site. The customer uses the

products and gives better suggestion for producing the product with better quality in future. Maintenance is to do rectifications in the software after delivery and usage by the customer at his site.

4. EVOLUTION OF TESTING

The concept of testing had no importance during early days of software design and development. The developers have to produce the product. Glenford Myen was the one who introduced the concept of testing. He started testing with all possible combinations of test cases by considering positive and negative approaches. The evolution of testing has 5 phases.

In phase 0, the testing was based on debugging. It was considered as the positive testing.

In phase 1, test cases were written and testing was performed. But phase 1 failed due to the possibility that with increasing test cases, the software may fail

In phase 2 testing were performed to check whether the softwareworks in normal conditions and does not fail to work under abnormal conditions.

In phase 3 testing was generally based on risk of working withthe acceptable value.

Phase 4 is the last phase of testing and the testing that is performed in recent times. The bugs found in testing was said as lucanac and testing team will work on finding the defects.

5.TEST PLANNING

The entire functions of the software are scheduled in a test plan.There are different sub-functions in test planning like test toolselection, estimation, resource scheduling. The outputs of test planning are test plan document and estimation document.

5.1. Test plan

The following details should be there in a test plan.

Test plan Identifiers gives a unique version number to a test plan.

Test scope gives a clear idea about the extent to which the testing is being performed.

Objective of testing gives the aim of testing and also provide with additional guidelines.

Assumption and Dependencies provides with documentation thus helps to give clarity about testing. Any negative impacts like cost of testing, quality improvement can have changes if required.

Risk analysis tells about possible number of risks during testing activity.

Roles and Responsibilities - The role of each tester is explainedin a detailed fashion. The roles and responsibilities of a testing team is scheduled here.

Test strategy - The testing can be black box or white box testing. Here planning about which strategy best suits here is done.

Test schedule - The sequence of testing, the approximate time and dependencies of testing is planned in test schedule.

Test Environment

The software requirements for testing and procedures to do software testing and to locate tester's artifacts is done in testingenvironmental.

Entry and Exit criteria - The commands are given to regulate testing process. Separate commands will be given to starttesting and stop testing.

Communication plan - It provides communication with customer in the aspect of documentation and other formal procedures. The tools used for testing is discussed in this platform.

5.2. Test design

In test designing, there are many phases including, starting testdesign by getting inputs, creating different test scenarios, forming test cases, constructing test data. These phases are executed using techniques like equivalence class partitioning, boundary value analysis. The main objective of this phase aimsat how to perform testing. The functions of these phases involves getting inputs, constructing test cases, work on alterations in test cases and test

scripts, the outputs of these phases are test data and test cases.

Once the testing team completes test planning they get clarity about the scope of testing and entire testing is easily done. Thekey role of a testing team is to form relevant and proper test cases.

5.3. Test scenario

Test scenario is a high level testing. The functionality of testingis prevailed here. All features of testing is included in the test scenario identification. If the test scenario is not identified in aparticular application, it may result in test not being performedin that area. Test scenario helps to test the entire application from end to end. Test scenario is an important function in testing, but it need not be formal.

5.4 Test case

Test case gives clear cut idea about testing a particular situation. Every detail to test a situation/application from getting input data to performing output expectations are explained in a test case. Test case is mainly used to find the original behaviour of an application.

Process of creating a test case

Test conditions have to be identified and documentation of thetest conditions is the first step in creating a test case. Test condition is usually identified using two formal ways.

In the first way unique conditions will be identified starting from breaking test scenario to error flow. In the second way, test conditions are satisfied by checking with use case to functionality testing in normal flow.

5.5. Test case Review

After completing the test cases, a Review need to be conductedto check the validation of testing. Review can be conducted inmany ways, peer to peer, within team members or a superior can initiate the review process.

5.6. Test Data

The subset of test case is a test data. While undergoing testing,test cases can be executed many times with the combination ofdifferent sets of test data. This is done to check the actual function of a particular application.

Requirements Tractability Matrix (RTM)

Requirements Tractability Matrix is a tool to check whether thescope of the software requirements gathered and outputs of theapplication are in a sequence. The main use of this tool is to check all requirements are utilized and to regulate the change of requirements.

5.7. Test Implementation

Test scripts are written with the help of automation tools in test implementation. The outcome of test implementation is test scripts.

5.8. Test execution

After the completion of test design phase, the next logical phaseis test execution. To check whether the outcome of the test caseis an expected behavior, test execution is performed. Based onthe instructions in the test plan execution of test is beingperformed.

5.9. Test closure

The last phase in the testing cycle is test closure. Here, data is collected to check whether the actual result match with theexpected result. After consolidation process, summary report is prepared and handed over to the maintenance team to assist with changes during maintenance.

5.   SOFTWARE TESTING TOOLS

Software testing tools are innumerable in market. The selectionof software tools is based on the particular project. The necessity of a project may differ, so selecting tools are based on project necessity. There are certain limitations for selectingtools. Free testing tools are available in the market. But they are with certain limitations. Software testing tools are divided into three sub categories.

- Test management tools
- Functional testing tools

- Load testing tools

6.1. Test management tools

TET: TET (Test environment tool kit) provides a test driver aiming at testing according to the needs of present and future test development community. To achieve this goal, inputs from different examples are taken into specification and functionality of the software application is specified.

TET ware: TET ware is a test execution management system that gives us with different features like administration, sequencing and reporting the test result in a beneficial manner.

Test Manager: Test Manager is used for day to day testing activity and is an automated testing tool. Test manager tool is developed using java programming language and mostly used for regular software development activities.

RTH: RTH is a Requirement and testing hub. It is an open source and used as a requirement management tool.

6.2. Functional testing tools

Ranorex: It is a simple efficient and cost effective automated testing tool. It is a better source when compared with other testing tools because it is user friendly and uses standard language like C+, vb.Net etc. A lot of commercial software companies are mostly using this tool. Future version of Ranorax is accessible and is a highly documented interface for the customers. This tool is based on record-replay functionality which is known as ranorex recorder. This tool provides features for the client to perform test automation process in client's environment. It is built on the .Net framework which is a purely coded format.

Rational Function tester: This tool was developed by IBM in early 1990s. Rational function tester is an automated testing tool used for testing equipment in a well written format. It gives quality assurance for the maintenance and execution of test plans. It also assists in non testing activities rather than testing activities. The error caused by humans is reduced in this tool.

6.3. Load testing Tool:

Load tracer: Load tracer is an automated web performance testing tool. It is a user friendly tool and mostly used for load testing, stress testing etc. this tool will have additional privilege in internet application performance testing.

6. CONCLUSION

Many studies in the field of software testing will be useful in future because we can prevent defects earlier. Thus this paper provides details about software testing techniques and about the different testing tools. Software testing is usually less formal because of the tough practice and methodologies of testing. Software testing is a team based project work and every individual have to perform their role for producing 100% bug free software. Though testers aim at producing 100% bug free software there will be defects found during the maintenance. Efforts are taken to remove bSugs and produce a quality software.

REFERENCES

[1] S.Kaushal and J.K.Bajwa,Analytical Review of User Perceived Testing Techniques,International journal of advanced Research in Computer Science and Software Engineering,vol.2,issue 10,pp 213-216,2012. http://technav.ieee.org/tag/1579/software-testing

[2] Irena,J;(2008) "Software Testing Methods and Techniques"

[3] IEEE (1990),IEEE Standard Glossary of Software Engineering Terminology,Los Alamitos, CA: IEEE Computer Society Press

[4] Kelkar ,S.A,Information Systems-A Concise Study,2nd edition,PHI Learning,New Delhi, 2009

[5] Perry,William E,Effective Methods for Software Testing,3rd Edition,Wiley,Indiapolis,2006

[6] ANSI/IEEE Standard 829-1983.Software Test Documentation

[7] Anderson,R.B.Proving Programs Correct.New York: John Wiley & Sons, 1979.

Boehm,B.W.Software and its impact: a quantitative assessment.Datamation