A Survey on Software Testing

Parasur Puja Popat[1], Ghule Akanksha Ramdas[2], Pokale Tejal Dinkar[3], Bhavar Payal Ashok

[1,2,3,4]Student, Computer Technology, Amrutvahini Polytechnic,Sangamner, Maharastra, India

---***---

Abstract - Software testing is the process of developing an application bug free or defect free. Aiming to produce 100% bug free software, testing team carries out various levels of testing. Software testing is a useful process to access the quality of the software. The various facilities provided by software testing are it reduces bugs in the software, decreases the cost of the software and aims at providing a software with low maintenance cost. But the major issue in testing is to find the appropriate test cases to test a software. We have a number of advancement in testing. But still the software has to be completely tested before it is delivered to the customer. Therefore, many techniques and goals are involved in software testing. In this paper, various software techniques, process involved in software testing, SDLC (software development life cycle), the steps involved in SDLC, the importance of software testing and its significance is explained in detail

Key Words: Software testing, techniques in software testing,SDLC, test planning, 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. Testing can be done in two ways manual testing and automation testing. The main objective of software testing is to produce a bug free software. Software testing is usually done by software tester, developer, project manager and end user. Software testing is done to ensure that the software meets the specified requirements. Software quality assurance is provided by software testing. The primary goal of software testing is to detect possible defects at early. Software testing is done in different stages in the development environment. It is to be discussed in brief. Verification and validation are important concept in software testing. A simple example can explain both verification and validation. Verification is are you building the product right. Validation is are you building the right product.

1.1 Verification

Verification is the process of evaluating work-products of a development phase to determine whether they meet the specified requirements. Verification ensures that the product is built according to the requirements and design specifications..

1.2 Validation

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs.
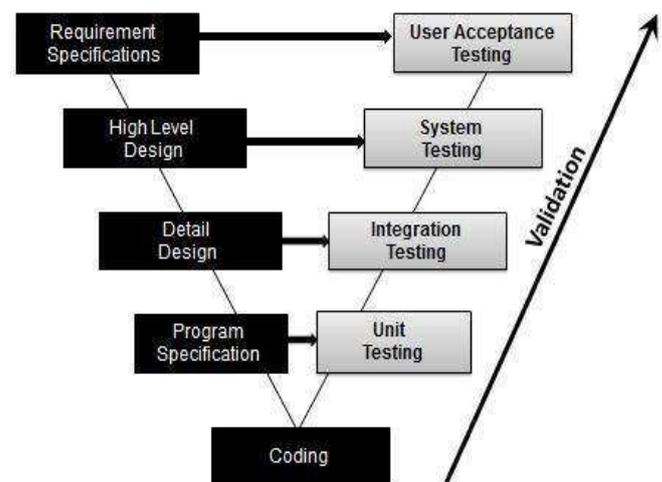

Fig:1 V-Model

2. SOFTWARE TESTING STRATEGIES

2.1 Unit testing

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

2.2 Integration testing

It is the combination of unit testing. In integration testing two or more modules are combined together and the process of testing is carried out. Data flow is checked from one module to another 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.

2.3 System testing

In system testing, the whole system is tested. All the components are integrated and tested to check whether the specified requirements are meet. It is a end to end process to ensure the quality of the software. The functional requirements of the software system are checked in system testing. This testing is usually done in the production environment. Both the business requirements and application architecture are tested and validated here.

2.4 Acceptance testing

It is the last stage of testing where the whole software is delivered to the customer and this testing is done in the user location. The main objective of acceptance testing is to ensure that the software works and functions properly rather than finding defects.

3. EVOLUTION OF TESTING

The concept of testing had no importance during early days. The developers have to produce the product and it will be finalized without testing. Glenford Myen was the one who introduced the concept of testing. He started testing with all possible combinations. The evolution of testing has 5 phases.

3.1 phase 0

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.

3.2 phase 1

But phase 1 failed due to the possibility that with increasing test cases, the software may fail

3.3 phase 2

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

3.4 phase 3

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

3.5 phase 4

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

4. TEST PLANNING

The entire functions of the software are scheduled in a test plan. There are different sub-functions in test planning like test tool selection, estimation, resource scheduling. The

outputs of test planning are test plan document and estimation document.

4.1 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. Test objective gives the aim of testing and also provide with additional guidelines. Assumption and Dependencies provides with documentation 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 risks during testing activity.

Role and Responsibilities

The role of each tester is explained in detail. 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 suit 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 testers artifacts is done in testing environmental.

Entry and Exit criteria

The commands are given to regulate testing process. Separate commands will be given to start testing 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.

4.2. Test design

In test designing, there are many phases including, starting test design 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 aims at 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. The key role of a testing team is to form relevant and proper test cases.

4.3 Test scenario

Test scenario is a high level testing. The functionality of testing is prevailed here. All features of testing is included in the test scenario identification. If the test scenario is not identified in a particular application, it may result in test not being performed in 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.

4.4 Test case

Test case gives clear idea about testing a particular application. Every detail to test an application from getting input data to performing output expectations are explained in a test case. Test case is mainly used to find the original behavior of an application. Process of creating a test case Test conditions have to be identified and documentation of the test 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.

4.5 Test case Review

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

4.6. Test Data

The subset of test case is a test data. While undergoing testing, test cases can be executed many times with the combination of different 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 the scope of the software requirements gathered and outputs of the application are in a sequence. The main use of this tool is to check all requirements are utilized and to regulate the change of requirements.

4.7 Test Implementation

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

4.8. Test execution

After the completion of test design phase, the next logical phase is test execution. To check whether the outcome of the test case is an expected behavior, test execution is performed. Based on the instructions in the test plan execution of test is being performed.

4.9. Test closure

The last phase in the testing cycle is test closure. Here, data is collected to check whether the actual result match with the expected 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 selection of 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 selecting tools. Free testing tools are available in the market. But they are with certain limitations. Software testing tools are divided into three sub categories.

1. Test management tools

2. Functional testing tools

3. Load testing tools

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 bugs and produce quality software.

REFERENCES

[1] Pavithra L, Sandhya S."Survey on Software Testing"s (JNCET) Volume 9, Issue 3, March(2019).

[2] 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 213216,2012.

[3] http://technav.ieee.org/tag/1579/softwaretesting.

[4] Irena,J;(2008) "Software Testing Methods and Techniques"IEEE (1990),IEEE Standard Glossary of Software Engineering Terminology,Los Alamitos, CA: IEEE Computer Society Press.