Software Testing-Three P'S Paradigm and Limitations

Sandeep Dalal Assistant Professor Department of Computer Science and Applications, M.D.University, Rohtak, (Hr)-India-124001

ABSTRACT

Software Testing is an important area of research in the field of software engineering because it is the deciding factor for successful project release and directly proportion to software quality and reliability. A lot of research has been already done in this field, likely to be leveraged in future as software has become integral part of each and every domain to make it faster, easier and smarter. Software testing is regarded as the most important phase of software development process, as without testing it is impossible to diagnose errors. This paper explains the phases, principles and limitations of software testing. We tried to accumulate principles after detailed analysis of phases of testing, based on available methodologies, users involved & importance of software testing. All the terms which are considered as building blocks of software testing are discussed in this paper to enhance the overall quality of a system and to ensure successful project release.

General Terms

Software Engineering, Software Testing

Keywords

Software Testing, Testing Importance, Testing principles, Testing Limitations.

1. INTRODUCTION

Software Testing is the process of executing a program or system with the intent of finding errors [2]. It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software testing is about the quality of the software product under test [1]. It can be stated as process of validating and verifying that a software application meets the requirements and works as expected. Software testing depends on the testing method used, can be implemented at any phase in the development process. Software testing is also used to test the software for other software quality parameters like reliability, usability, integrity, security, capability, efficiency, portability, maintainability and compatibility [4]. The purpose of testing can be quality assurance, verification and validation or reliability estimation. Testing certifies that software works properly under all conditions/environments and also identifies those areas which are not working appropriately [3].Software testing has its own life cycle that intersects with every stage of the SDLC [5]. SDLC is software or system development life cycle which has major phases- Requirement gathering, analysis and elicitation, Designing, Coding, Testing and Implementation. Figure 1 shows all phases of SDLC at a high level following an iterative approach. Testing in the process could lead to change in requirements, designs or code. There are various SDLC models used in the industry depending upon need of the business. All the models use same phases of

Rajender Singh Chhillar, PhD. Professor & Head Department Computer Science and Applications,M.D.University,Rohtak (Hr)-India-124001

development process, only the difference is their representation and flow of information.



Figure 1: Software Development life cycle

Software testing is one of the most important phases in software development life cycle. Importance of testing in software development life cycle is to improve reliability, performance and other important factors, which may define under software requirement specification.

2. SOFTWARE TESTING PHASES:

Software testing has its own life cycle that meets every stage of the SDLC. At high level software testing could be categorized in four phases which are Test Planning, Test Designing, Test Execution and Bug Logging and Tracking.



Figure 2: Software Testing Phases

Figure 2 shows all the phases of software testing which are later leveraged in testing life cycle in this paper.

2.1 Test Planning

Test planning describes how testing should be accomplished [13]. In Test Planning following are the major tasks:

- Define scope and objective of testing
- Identify Test Strategy or approaches
- Define risk
- Resource Allocation
- Time Schedule
- Effort Estimation
- Types of Testing to be performed
- Exit criteria etc.

2.2 Test Designing

This phase includes Test Case and Test data preparation. Standard template should be used to prepare test cases and multiple test data sets should be prepared for all the test cases. Positive and Negative test cases and data is prepared for all the requirements in this phase.

2.3 Test Execution and Status Reporting

System and Integration test cases are executed in this phase. Test Execution is done by tester with intent of finding bugs. In this phase test cases are executed and results are updated as Pass/Fail in the test case template. Bugs are opened for the failed cases and after fix test cases are again executed which is termed as regression testing. Regression testing ensures that the functionality after fixing the bug is intact. During this phase, the focus of the key stakeholders and management is on the progress and findings of the Testing team. Based on the Test report of the testing team, the management gets confidence about the quality of the product/application. The management takes decision regarding the release of the product/application based on this report. Hence, the test report should be prepared in such a manner so that it helps in taking decisions for deployment[12]. This is an iterative process involving execution of Test Cases, Detection of Bugs, Bug reporting, and Modification of test cases if felt necessary, fixing of bugs by developers & finally repeating the testing cycles.

2.4 Bug Logging and Tracking

A bug is a variance from specification. A bug is defined as: "Any significant, unplanned event that occurs during testing that requires subsequent investigation and/or correction. Bugs are raised when expected and actual test results differ". Bug tracking is the process of finding bugs in a product, (by inspection, testing, or recording feedback from customers), and making new versions of the product that fix the bugs. Software systems may have hundreds of bugs. Bug tracking is the process of identifying and evaluating these bugs followed by prioritizing and managing them.

Bug logging and tracking and Test execution phase go hand in hand. Bugs are opened for the failed cases and assigned to developer to fix them. After fix, tester again re-tests the functionality to make sure the bug is fixed properly and existing functionality in not broken. Bug logging should be done following Ideal Fault reporting system principles [12]. This is done using some bug logging and tracking tool or could be done as part of test management tool.

3. SOFTWARE TESTING PROCESS:

The process of testing a software in a well planned and systematic way following a methodology is known as software testing lifecycle(STLC).STLC starts with analyzing requirements, then preparing the detailed test plan[5] with strategy and scope of testing. Testing team then prepares test scenario and cases following a standard template based on requirements document. Once cases are prepared and test environment is available, then the tester identifies test data sets for all positive and negative cases. By the time development team completes writing code, test team should be ready to execute test cases. When environment gets ready and smoke and sanity test passes [11] then test execution begins and testers start bug logging and tracking. Once bugs are closed then regression testing is performed to ensure nothing is left broken. The testing cycles are repeated until new bugs are not found and regression testing should be passed. Requirements are reviewed by the testing team to make sure requirements are completed and understood properly. Figure 3 explains the complete testing process and also describes how testing and development progresses side by side. Business analyst prepares requirement documents; QA and Development teams' takes walkthrough of requirements from business analyst. Once requirements are completed then testing team starts preparing test plan, test scenario, cases and data. Development team on the other hand starts preparing code. Figure 3 illustrates that by the time development team completes development, testing team should be done with all test planning and should be in a position to execute test cases.

International Journal of Computer Applications (0975 – 8887) Volume 54– No.12, September 2012



Fig 3 Software Testing Life cycle

4. SOFTWARE TESTING PRINCIPLES

When, What, How, Who, and Why of Testing opens all the doors of software testing, answers all questions and solve all puzzles around principles of testing.



4.1. When to Start and Stop Testing?

a) Start testing early and at all phases: Testing should be started at the time of requirement planning & elicitation with the help of business analysts involving static analysis and formal reviews and walkthroughs. Test Planning, Test Scenario preparation and Test Case preparation should be done after requirement phase is completes to ensure that when coding completes testing team used to ready to execute test cases[14].

b) Criteria for stopping testing: Stop testing when pre-defined scope of testing is achieved, all stakeholders are demoed the product to be released and all agree to the functionality and if there are any known bugs then they are put forward and there is no show stopper to the application.

4.2. What Should Be Tested?

a) Critical Functionality: Test cases needs to be drafted around all requirements and all defined requirements should be tested using set of positive and negative data sets.

b) Usability of Application: Ensure that the application is easy to use and end customers would feel it easy to navigate and perform actions. Application should be self guiding and intuitive in nature.

c) Performance Testing: All non-functional aspects of software which covers performance parameters should be validated appropriately under all defined environments & browsers. Example- Access time, load time, run time etc.

d) Look and Feel Testing: Need to make sure the look and feel of testing is in sync when the monitor settings are modified and application runs of various browsers.

4.3. How to ensure the Bug free Software?

a) Qualified Test Team: Ensure that test team is qualified enough to test the application effectively. Team should have necessary skills to work on project.

b) Ensure that positive and negative cases are drafted to test basic and alternate flows of application using all test data sets generated based on equivalence partitioning to ensure that application works fine in adverse condition and open bugs for all the irregularities. Please note that the productivity of tester is counted on number of bugs opened & tester job is to breaking the code in all possible ways to make the application full proof. [9].

c) Ensure that test case walkthrough is given to all the stakeholders to make sure no test case is missed and all requirements are tested. This activity needs to be done to take approval from all the stakeholders on test cases.

d) Make sure that test data is prepared in advance prior to test execution for all positive and negative test cases. Make sure to have 3 to 4 test data sets for each and every test case.

e) Ensure that a bug is opened for each and every anomaly or deviation from requirement be it high priority, severity or vice versa. Decision on deferring of bug could be taken by business or project managers prior releasing the project.

f) Ensure that all bugs are closed and regression tested prior to project release.

g) Follow proper bug logging and tracking mechanism [6].

h) Prepare Requirement Traceability Matrix to make sure that test cases are designed and executed for each and every requirement and to ensure 100% test coverage.

i) Never assume anything and perform testing as per the requirement and as per the approved test cases.

j)Make sure root cause analysis is done post release to take a note of lessons learnt, what went well, what did not went well and update all the project metrics.

4.4. Who should be involved while testing?

a) Business Team: There should be involvement of business team in making sure no requirement is missed and need to make sure requirement document is clear and complete. Business team would also be involved in UAT (User acceptance testing)

b) Development Team: Dev team should perform unit test plan, test cases and record their results for all unit test cases.

c) Testing Team: Test Team should follow complete test life cycle and bug logging and tracking.

d) Customer and potential users: Ensure that alpha and beta testing is performed on the application to report any bugs in client environment and from end user stand point.

4.5. Why Testing should be performed?

a)Verification and validation of a software product is the process of determining if the system meets its predefined goals and the output is correct. Planning for this phase of testing starts early in the software development life cycle. Verification and validation can be performed by the same organization that developed the product, but are more effective if performed by an independent testing agency. This

is to ensure Scope of testing is covered and all the modules and components are tested as defined in test plan.

b) Computers and software are heavily used in critical fields like medical diagnosis, airplanes and air traffic control, space shuttle missions and stock market reporting. The presence of bugs in the software application can cause irreparable losses. Quality of software is of utmost importance, and making sure the software meets quality standards is the job of the software test engineer.

c) Testing gives confidence to the team that software would not fail after project would go to production or after release.

d) Testing helps business team in deciding a go, no-go of the project release. This would also help decide the team in estimating the effort required to fix the bugs and re-test it so when the project could be deployed.

e) For Reliability Estimation: From the user point of view, reliability means how dependable the software product is. In medical diagnosis, an incorrect suggestion to the doctor can result in the loss of lives. Critical software products are thoroughly checked for all aspects of its functionality.

5. SOFTWARE TESTING LIMITATIONS

It is very important to understand the limitations of testing. Testing does not assure success or even prove that a product will be usable. Even the most strictly conducted formal test cannot ensure that a product will be usable when released. Testing has following limitations:

a) Predefined Testing time not allocated when testing phase begins: One of the limitations of traditional approach is that in most of the cases deployment date is announced prior to initiation of the project and in this process developers consume most of the time from the schedule which leaves lesser time for testing and later due to insufficient testing project fails after deployment or software release is delayed which incurs cost to business.

b) 100% Testing not possible in case of complex systems: When there is a complex project which involves lot of interactions with downstream and upstream applications then there are chances of not performing thorough testing and pass criteria defined are not met due to lots of dependencies on outside systems.

c) Lack of formal testing and reviews at requirement and design stage of software development life cycle.

d) Lack of formal unit testing methodology and assembly testing prior to handling the product to testing team.

e) Lack of efficient and effective automation testing as selection and evaluation of tool is difficult due to availability of lots of tools and high cost of tools which supports different programming languages that becomes difficult for testers to understand and learn the programming language. Also most of the automation tools require writing lots of code rather than

just configuring the tool and feeding stable application to tool to test the critical components and performance parameters.

f) Testing implicit requirements and obvious functionality needs more experienced testers and this is generally not defined anywhere in project plan and test plan. For Ex-Clicking cancel or remove link should show some warning edit with okay and cancel button saying that the unsaved information will be lost. Such requirements are sometimes skipped; however testers ensure that implicit requirements be tested and fixed.

g) In today's scenario where all applications are required to be developed for mobiles, tablets, and traditional systems, their interoperability, compatibility with various hardware, and for all available browsers i.e. all versions of IE, Firefox, Chrome, Mac etc, makes it difficult to ensure the product's stability in all environments and end to end testing.

h) The developer to tester ration has increased to 1:3 for medium sized products, which makes it more expensive to test the product end to end.

i) Testing does not assure success or even prove that a product will be usable. Even the most strictly conducted formal testing cannot ensure that a product will be usable when released. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions [7].

j) Testing can be used to show the presence of errors, but never to show their absence! [8]. Testing is only able to identify the known issues or errors. It gives no idea about defects still uncovered. Testing cannot guarantee that the system under test is error free.

k) In software testing you cannot test every valid input, invalid input and path.

I) It is not a secret that in some cases it is more effectual both in terms of time, cost, and precision to conduct an expert or heuristic estimation of a product rather than test it. This is particularly true in the early steps of a product when gross violations of usability principles abound. It is simply unnecessary to bring in many partakers to reveal the evident [10].

- m) Exhaustive testing is not possible in current scenario.
- n) Complete testing is infeasible. Complexity is the root of the problem. At some point, software testing has to be stopped and product has to be shipped [11].

6. CONCLUSION

The primary role of software testing is not to demonstrate the correctness of software product ,but to expose hidden defects so that they can be fixed and has the potential to save time and money by identifying problems early and to improve customer satisfaction by delivering a more defect free product. This paper briefly assesses the [6] significance of software testing Phases, Process, Principles and Limitation. Without Proper testing there is a lot of risk of software fail due to which there is loss of time, life and money. So to perform testing effectively and efficiently, everyone involved with testing should be familiar with basic software testing Three P'S and Limitations so that defects and there root causes should be traced efficiently.

7. REFERENCES

- [1] Cem Kaner Florida : Institute of Technology, Quality Assurance Institute Worldwide Annual Software Testing Conference, Orlando, FL, November 2006.
- Myers, Glenford J., *The art of software testing*, *Publication info: New York*: Wiley, c1979. ISBN: 0471043281 Physical description: xi, 177 p.
- [3] Kaner, Cem; Falk, Jack and Nguyen, Hung Quoc (1999). Testing Computer Software, 2nd Ed.. New York, et al: John Wiley and Sons, Inc.. pp. 480 pages. ISBN 0-471-35846-0.
- [4] Kolawa, Adam; Huizinga, Dorota (2007). Automated Defect Prevention: Best Practices in Software Management. Wiley-IEEE Computer Society Press. pp. 41–43. ISBN 0-470-04212-5.
- [5] Mogyorodi, G. What is Requirements Based Testing? The Journal of Defense Software Engineering, 12-15 (2003).

- [6] Thomas Zimmermann, Rahul Premraj, Jonathan Sillito, Silvia Breu. Improving Bug Tracking Systems. In Companion to the 31th International Conference on Software Engineering (ICSE Companion 2009), Vancouver, BC, Canada, May 2009.
- [7] S.M.K. Quadri and Shiekh Umar Farooq, "Software Testing- Goals Principles, and Limitations", Intenatonal journal of Computer Applications(0975 - 8887) Vol. 6, No. 9, September 2010.
- [8] Miller, William E. Howden, "Tutorial, software testing & validation techniques", IEEE Computer Society Press, 1981
- [9] PRLog (Press Release) <u>http://www.prlog.org/11590653-limitations-of-software-testing.html</u> Oct 04, 2011
- [10] Myers, Glenford J., The art of software testingl, New York: Wiley, c1979. ISBN: 0471043281
- [11] Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, Inc., New York (2001).
- [12] Sandeep Dalal and Rajender Chhillar "Role of Fault Reporting in Existing Software Industry"CiiT International Journal of Software Engineering and Technology July,2012
- [13] Test Planning by Josh Probert, www.joshprobert.com/Testplan.pdf www.cs.nott.ac.uk/~cah/G53QAT/Reports09/.../QAT09 Report-jxp17u.doc
- [14] Amrit L. Goel, Kazuhira Okumoto,"When to stop testing and start using software?" April 1981 ACM workshop