ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)
Vol. 5, Issue 3 (July - Sept. 2018)
International Journal of Advanced Research in Education & Technology (IJARET)

A Study on Software Testing

[I]Deepthi S, [II]Sree Raj M P
[I]M.Tech Graduate, Dept. of ECE, University of Kerala, Kerala, India.
[II]Assistant Professor, Dept. of Mechanical Engineering, Rajadhani Institute of Engineering & Technology, Kerala, India.

Abstract
Software Testing is a process of finding errors while executing a program so that we get a zero defect software. It is aimed at evaluating the capability or usability of a program. Software testing is an important part in software development process. Complex systems are being built and testing throughout the software development cycle is valid to the success of the software. Effective Testing produces high quality software.

Keywords
Software Testing, SDLC, STLC, Manual Testing, Automation Testing, Defect Life Cycle and defect.

I. Introduction

Software testing is a process of executing a program which is used to find the correctness, completeness and the quality of software or the application. Software testing is the procedure of executing the application under positive and negative condition by manually or automatically. Software testing is done with the intent of locating and finding the software bugs. Software Testing is a very crucial in software development life cycle (SDLC).The main aim of software testing is to uncover as many errors or bugs as possible. It is used to see that the software product is matching the requirements or not. It is also used to validate the quality of software. Testing is the process of making certain secret that a program does what it is took as probable to do.

II. Need For Testing

Software Testing is necessary because we all make mistakes. Some of those mistakes are unimportant, but some of them are expensive or dangerous. We need to check everything and anything we produce because things can always go wrong –humans make mistakes all the time.

Software testing is very important because of the following reasons:

1. Software testing is really required to point out the defects and errors that were made during the development phases.
2. It is essential since it makes sure of the Customer's reliability and their satisfaction in the application.
3. It is very important to ensure the Quality of the product. Quality product delivered to the customers helps in gaining their confidence. (Know more about Software Quality)
4. Testing is necessary in order to provide the facilities to the customers like the delivery of high quality product or software application which requires lower maintenance cost and hence results into more accurate, consistent and reliable results.
5. Testing is required for an effective performance of software application or product.
6. It is important to ensure that the application should not result into any failures because it can be very expensive in the future or in the later stages of the development.
7. It is required to stay in the business.

III. Priciples of Testing

There are seven principles of testing. They are as follows:

1. Testing shows presence of defects:
   Testing can show the defects present, but cannot prove that there is no defect. Even after testing the application or product thoroughly we cannot say that the product is 100% defect free. Testing always reduces the number of undiscovered defects remaining in the software but even if no defect is found, it is not a proof of correctness.
2. Exhaustive testing is impossible:
   Testing everything including all combinations of inputs and preconditions is not possible. So, instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts. For example: In an application in one screen there are 15 input fields, each having 5 possible values, then to test all the valid combinations you would need 30 517 578 125 (5^{15}) tests. This is very unlikely that the project timescales would allow for this number of tests. So, accessing and managing risk is one of the most important activities and reason for testing in any project.
3. Early testing:
   In the software development life cycle testing activities should start as early as possible and should be focused on defined objectives.
4. Defect clustering:
   A small number of modules contains most of the defects discovered during pre-release testing or shows the most operational failures.
5. Pesticide paradox:
   If the same kinds of tests are repeated again and again, eventually the same set of test cases will no longer be able to find any new bugs. To overcome this "Pesticide Paradox", it is really very important to review the test cases regularly and new and different tests need to be written to exercise different parts of the software or system to potentially find more defects.
6. Testing is context dependent:
   Testing is basically context dependent. Different kinds of sites are tested differently. For example, safety – critical software is tested differently from an e-commerce site.
7. Absence – of – errors fallacy:
   If the system built is unusable and does not fulfil the user's needs and expectations then finding and fixing defects do not help.

IV. SDLC

There are various software development approaches defined and designed which are used/employed during development process of software, these approaches are also referred as "Software Development Process Models" (e.g. Waterfall model, incremental

model, V-model, iterative model, RAD model, Agile model, Spiral model, Prototype model etc.). Each process model follows a particular life cycle in order to ensure success in process of software development.

Software life cycle models describe phases of the software cycle and the order in which those phases are executed. Each phase produces deliverables required by the next phase in the life cycle. Requirements are translated into design. Code is produced according to the design which is called development phase. After coding and development the testing verifies the deliverable of the implementation phase against requirements. The testing team follows Software Testing Life Cycle (STLC) which is similar to the development cycle followed by the development team.

There are following six phases in every Software development life cycle model:

1. Requirement gathering and analysis
2. Design
3. Implementation or coding
4. Testing
5. Deployment
6. Maintenance

1. Requirement gathering and analysis
   Business requirements are gathered in this phase. This phase is the main focus of the project managers and stake holders. Meetings with managers, stake holders and users are held in order to determine the requirements during a requirements gathering phase. After requirement gathering these requirements are analyzed for their validity and the possibility of incorporating the requirements in the system to be development is also studied.
   Finally, a Requirement Specification document is created which serves the purpose of guideline for the next phase of the model. The testing team follows the Software Testing Life Cycle and starts the Test Planning phase after the requirements analysis is completed.
2. Design
   In this phase the system and software design is prepared from the requirement specifications which were studied in the first phase. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture. The system design specifications serve as input for the next phase of the model.
3. Implementation or coding
   On receiving system design documents, the work is divided in modules/units and actual coding is started. Since, in this phase the code is produced so it is the main focus for the developer. This is the longest phase of the software development life cycle.
4. Testing
   After the code is developed it is tested against the requirements to make sure that the product is actually solving the needs addressed and gathered during the requirements phase. During this phase all types of functional testing like unit testing, integration testing, system testing, acceptance testing are done as well as non-functional testing is also done.
5. Deployment
   After successful testing the product is delivered / deployed to the customer for their use. As soon as the product is given to the customers they will first do the beta testing. If any changes are required or if any bugs are caught, then they will report it to the engineering team. Once those changes are made or the bugs are fixed then the final deployment will happen.
6. Maintenance
   Once when the customers starts using the developed system then the actual problems comes up and needs to be solved from time to time. This process where the care is taken for the developed product is known as maintenance.

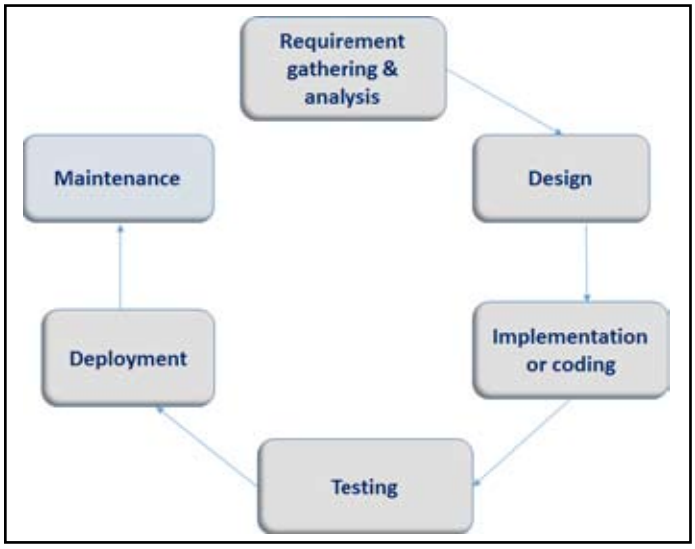

Fig.1: Software Development Life Cycle (SDLC)

V. STLC

In STLC process, every operation is executed in a planned and orderly way. Each phase has different goals and things to be given. Different organizations have different sides and different phases in STLC however the basis remains the same. Software Testing Life Cycle mainly consists of various steps, and are:

1. Requirement analysis
2. Test Planning
3. Test case development
4. Environment Setup
5. Test Execution
6. Test Cycle Closure

1. Requirement Analysis
   In this phase testing team goes through the Requirement document with both Functional and non-functional details in order to identify the testable requirements. In case of any confusion the QA team may setup a meeting with the clients and the stakeholders (Technical Leads, Business Analyst, System Architects and Client etc.) in order to clarify their doubts. Once the QA team is clear with the requirements they will document the acceptance Criteria and get it approved by the Customers.
2. Test Planning
   Test Planning phase starts soon after the completion of the Requirement Analysis phase. In this phase the QA Manager or QA Lead will prepare the Test Plan and Test strategy documents. As per these documents they will also come up with the testing effort estimations.
3. Test case development
   In this phase the QA team write test cases. They also write scripts for automation if required. Verification of both the

test cases and test scripts are done by peers. Creation of Test Data is done in this phase.

4. Environment Setup
   This phase includes the setup or installation process of software and hardware which is required for testing the application. In this phase the integration of the third party application is also carried out if required in the project. After setting up the required software and hardware the installation of build is tested. Once the installation of build is successful and complete then the Test Data is generated. After the creation of Test data the Smoke testing is executed on the build in order to check whether the basic functionalities are working fine or not. This phase can be done in parallel with the Test Case Development phase.
5. Test Execution
   Before starting the Test Execution phase the Test Environment setup should be ready. In Test Execution phase the test cases are executed in the testing environment. While execution of the test cases the QA team may find bugs which will be reported against that test case. This bug is fixed by the developer and is retested by the QA.
6. Test Cycle Closure
   In order to start the Test Cycle Closure activity the Test Execution phase should be completed. In Test Cycle phase the QA team will meet and discuss about the testing artifacts. The whole intent of this discussion is to learn lessons from the bad practices. This will help in future projects.

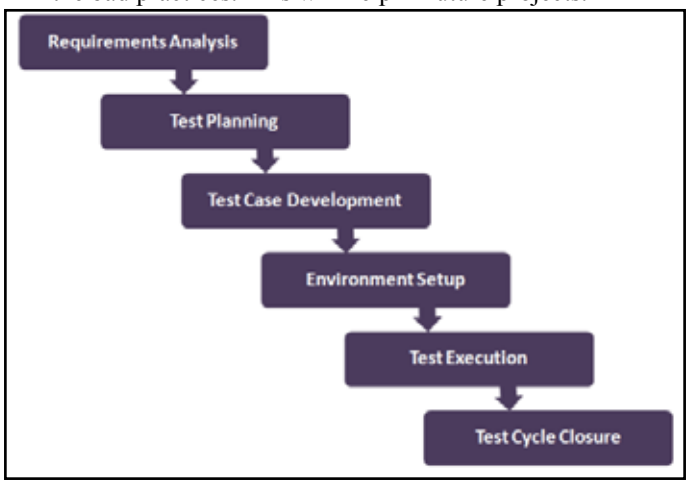

Fig.2: Software Testing Life Cycle (STLC)

VI. Manual Testing

Manual Testing is a type of Software Testing where testers manually execute test cases without using any automation tools. Manual Testing is the most primitive of all testing types and helps to find bugs in the software system. Any new application must be manually tested before its testing can be automated. Manual Testing requires more effort, but is necessary to check automation feasibility. Manual Testing does not require knowledge of any testing tool.

The key concept of manual testing is to ensure that the application is error free and it is working in conformance to the specified functional requirements. It also makes sure that reported defects are fixed by developers and re-testing has been performed by testers on the fixed defects. Basically, this testing checks the quality of the system and delivers bug-free product to the customer.

VII. Automation Testing

Automation Testing means using an automation tool to execute your test case suite. The automation software can also enter test data into the System Under Test, compare expected and actual results and generate detailed test reports. Test Automation demands considerable investments of money and resources. Successive development cycles will require execution of same test suite repeatedly. Using a test automation tool it is possible to record this test suite and re-play it as required. Once the test suite is automated, no human intervention is required.

Goal of Automation is to reduce the number of test cases to be run manually and not to eliminate Manual Testing all together.

Automated software testing is important due to the following reasons:

- Manual Testing of all work flows, all fields , all negative scenarios is time and cost consuming
- It is difficult to test for multi lingual sites manually
- Automation does not require Human intervention.
- Automation increases speed of test execution
- Automation helps to increase Test Coverage
- Manual Testing can become boring and hence error prone.

VIII. Defect Life Cycle

Defect life cycle is a cycle which a defect goes through during its lifetime. It starts when defect is found and ends when a defect is closed, after ensuring that it is not reproduced. Defect life cycle is related to the bug found during testing.

The bug has different states in the Life Cycle. The Life cycle of the bug can be shown diagrammatically as follows:

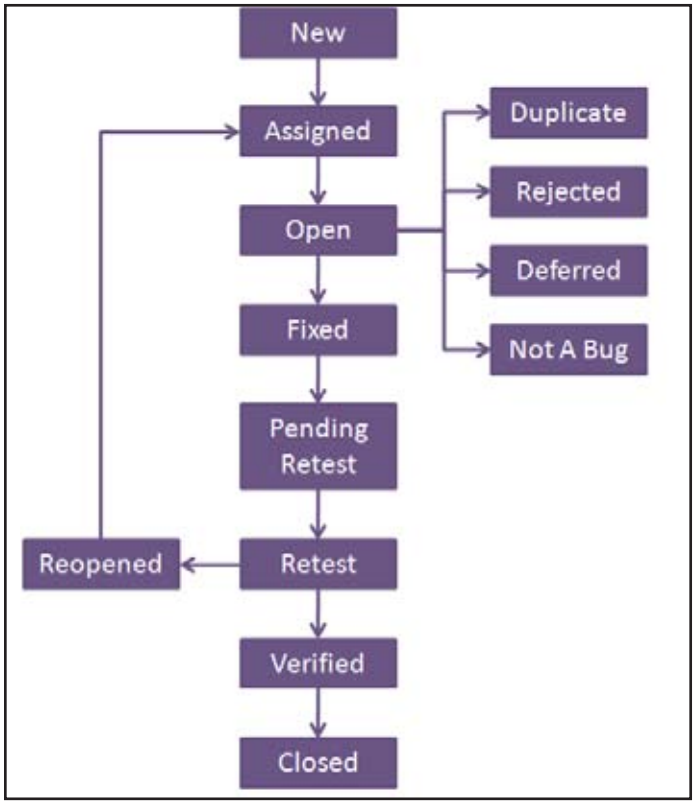

Fig.3: Defect Life Cycle

Bug or defect life cycle includes following steps or status:

1. New: When a defect is logged and posted for the first time. Its state is given as new.
2. Assigned: After the tester has posted the bug, the lead of the

tester approves that the bug is genuine and he assigns the bug to corresponding developer and the developer team. Its state given as assigned.

3. Open: At this state the developer has started analyzing and working on the defect fix.
4. Fixed: When developer makes necessary code changes and verifies the changes then he/she can make bug status as 'Fixed' and the bug is passed to testing team.
5. Pending retest: After fixing the defect the developer has given that particular code for retesting to the tester. Here the testing is pending on the testers end. Hence its status is pending retest.
6. Retest: At this stage the tester do the retesting of the changed code which developer has given to him to check whether the defect got fixed or not.
7. Verified: The tester tests the bug again after it got fixed by the developer. If the bug is not present in the software, he approves that the bug is fixed and changes the status to "verified".
8. Reopen: If the bug still exists even after the bug is fixed by the developer, the tester changes the status to "reopened". The bug goes through the life cycle once again.
9. Closed: Once the bug is fixed, it is tested by the tester. If the tester feels that the bug no longer exists in the software, he changes the status of the bug to "closed". This state means that the bug is fixed, tested and approved.
10. Duplicate: If the bug is repeated twice or the two bugs mention the same concept of the bug, then one bug status is changed to "duplicate".
11. Rejected: If the developer feels that the bug is not genuine, he rejects the bug. Then the state of the bug is changed to "rejected".
12. Deferred: The bug, changed to deferred state means the bug is expected to be fixed in next releases. The reasons for changing the bug to this state have many factors. Some of them are priority of the bug may be low, lack of time for the release or the bug may not have major effect on the software.
13. Not a bug: The state given as "Not a bug" if there is no change in the functionality of the application.

IX. Conclusions

Quality is the main focus of any software engineering project. Without measuring, we cannot be sure of the level of quality in a software. So the methods of measuring the quality are software testing techniques. This paper relates to the basics and fundamentals of Software Testing. Software testing research is the driving element of development and application. In this era of new and higher demand of software testing, it is important to constantly summarize new achievements, fresh hotspots and propose different ideas in order to promote the study on software testing system engineering, to facilitate the rapid development on software testing field and industry.

AcknowledgEment

We wish to express our deep sense of gratitude to all members of faculty in the Department, our parents and all our friends for their whole hearted cooperation and encouragement. Above all we bow our mind before the grace from beyond the blues that bestowed on us, sense in amplitude, to realize this venture.

References

[1]. https://en.wikipedia.org/wiki/Software_testing
[2]. http://istqbexamcertification.com/
[3]. https://www.guru99.com/manual-testing.html
[4]. https://www.guru99.com/automation-testing.html