

Quality Assurance Framework for Software Testing

Shwetha M N

M.Tech, Department of Computer Science
GSSSIETW, Metagalli, Mysuru , Karnataka , India.
Vishweshvara Technological University

Abstract-- Software testing is both an order and a procedure. The essential goal is to comprehend the idea of software Testing and the Software Testing methods which organizations use to accomplish Quality Assurance. Software testing is an Iterative process. The Iterative process consists of software requirement specification, designing test cases, executing the test cases, identifying the defect and providing the solution for that defect.

Keywords— Unit testing, Integration testing, system testing, manual testing, automation testing

I. INTRODUCTION

Software testing is both a control and a procedure. Despite the fact that product testing is a piece of the product Improvement process, it ought not be viewed as part of programming advancement. It is a different order from programming advancement. Programming advancement is the way toward coding usefulness to meet characterized end user needs. Programming testing is an iterative procedure of both approving usefulness, and, considerably more critical, endeavoring to break the product. The iterative procedure of programming testing comprises of Planning tests, Executing tests, Identifying issues, getting issues fixed. The goal of programming testing is to discover issues and fix them to improve quality. Programming testing normally speaks to 40% of a programming advancement spending plan. Main aim of this paper is to provide the quality assurance for the software testing.

II. GOALS OF SOFTWARE TESTING

Regularly the meaning of testing contrasts dependent on it procedure, reason and the dimension of testing. The principle target of testing is to discover absconds that are made by the developer while building programming. It likewise ensures that it meets the business and the client necessities [2]. It additionally persuades the Software necessity detail. It should pick up the certainty of the clients by conveying the quality item.

A. The Testing Spectrum

Software testing includes Software life cycle in each organize, however in each dimension the product improvement contrasts in develop and has distinctive points.

- Unit testing will be accomplished for the low dimension. It tests the principal unit of the product. Nearby information are Scrutinize to ensure that trustworthiness can be looked after [2].

- Integration testing is done when at least two units are joined into a more noteworthy structure. It is accomplished for the two interfaces between the segments and the structure that is being developed.

- System testing attests the end – to-end nature of the framework. The framework depends on the practical detail of the framework. The non-practical properties, for example, security, unwavering quality are too checked [2].

- Acceptance testing is examined when entire framework is given over from the engineer to the client .The principle goal of this testing is to give a certainty that framework is attempting to discover the blunders.

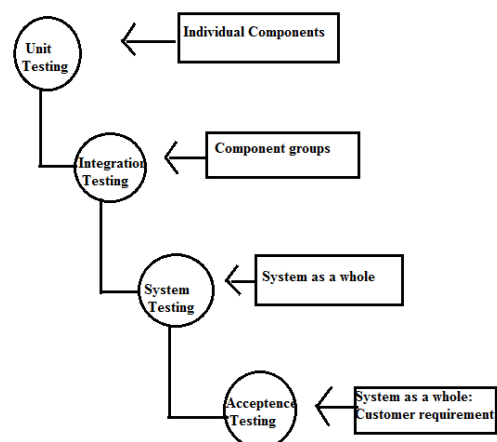


Figure:1 levels of Testing

B. Methods of Software Testing

There are two fundamental techniques for performing programming testing: Manual Software Testing - As the name would suggest, manual programming testing is the procedure of an individual or people physically testing programming. Robotized Software Testing-Automated programming testing is the way toward making test contents

that would then be able to be run naturally, tediously, and through much cycle. Accomplishing the Right Blend of Programming Testing A powerful programming testing process is normally a blend of test types, executed through a blend of manual and mechanized testing. The blend what's more, number of tests is dictated by the quality prerequisites of the application. Every strategy (robotized or manual) is utilized for what is proper. Manual testing is best utilized for those tests which require immediacy and innovativeness, too a decent arrangement of subjectivity, UI or ease of use testing and exploratory/impromptu testing .While computerized testing is best utilized for tests which are express and dull, general QA and usefulness tests (for example does each module do what the prerequisites state it should? How does the application react to off base information sources?) , 'End to end' situation tests (reenacting a 'genuine world' utilization of the product in a creation domain), execution, burden, and stress testing.

III. CONCEPTS OF TESTING

A. Definition of Testing

Software testing is performed to confirm that the finished programming bundle capacities as per the desires characterized by the prerequisites/determinations. The general goal to not to discover each product bug that exists, however to reveal circumstances that could contrarily affect the client, ease of use or potentially maintainability.

B. History

The separation of debugging from testing was initially introduced by Glenford J. Myers in 1979. Although his attention was on breakage testing ("a successful test is one that finds a bug") it illustrated the desire of the software engineering community to separate fundamental development activities, such as debugging, from that of verification. Dave Gelperin and William C. Hetzel classified in 1988

The phases and goals in software testing in the following stages:-

- Until 1956 - Debugging oriented
- 1957–1978-Demonstration oriented
- 1979–1982 - Destruction oriented
- 1983–1987 - Evaluation oriented
- 1988–2000 - Prevention oriented

C. Successful Testing Strategies

The following are a portion of the tips which each analyzer should remember before Testing any application:-

The last test outcome might be pass fail however investigating the underlying driver of fail will lead us to the arrangement of the issue. Analyzers will be regarded on the off chance that they log the bugs as well as give arrangements.

- To guarantee greatest test inclusion break our application under test (AUT) into littler practical modules.
- Make our experiments accessible to designers before coding.
- Ward off designers from test condition. This is expected advance to identify any setup changes missing in release or sending file. Every so often architects complete a couple of framework or application arrangement changes yet neglect to make reference to those in organization steps. On the off chance that engineers don't approach testing condition they won't do any such changes inadvertently on test condition and these missing things can be caught at the ideal spot.
- Its a decent practice to include testers directly from programming prerequisite and configuration stage. These way analyzers can get learning of use reliability bringing about nitty gritty test inclusion.
- On the off chance that conceivable recognize and gathering our experiments for relapse testing. This will guarantee fast and viable manual relapse testing.

IV. ADVANTAGES OF TESTING

To improve quality: As PCs and programming are utilized in basic applications, the result of a bug can be extreme. Bugs can cause colossal misfortunes. Bugs in basic frameworks have caused plane accidents, permitted space transport missions to go amiss, stopped exchanging on the securities exchange, and more regrettable. For Example: The alleged year 2000 (Y2K) bug has brought forth a cabin industry of specialists also, programming instruments committed to ensuring the current world doesn't go to a sudden end on the first day of the following century. Quality methods the conformance to the predefined plan prerequisite. Being right, the base necessity of value, implies executing as required under determined conditions.

For Verification and Validation (V&V) Another critical motivation behind testing is check and approval (V&V). It

is intensely utilized as an instrument in the V&V process. Analyzers can make claims dependent on elucidations of the testing results, which either the item works under specific circumstances, or it doesn't work.

For unwavering quality estimation Programming unwavering quality has imperative relations with numerous parts of programming, including the structure, and the measure of testing it has been exposed to. Testing is required to ensure that Application created is in agreement with Client and have executed with all the necessities from the customer. Testing can't build up that an item capacities legitimately under all conditions however can just set up that it doesn't work appropriately under explicit conditions.

V METHODS OF SOFTWARE TESTING

Software development is a complex process and involves a series of tasks to be carried out including code testing. Testing can be carried out through automated or Manual Testing Method. Testing is a specialized skill or domain; it can be carried out by only seasoned professionals who are well versed with various testing methods and tools. Numerous automated testing tools and methods are available for testers; however, Manual Testing method has its own importance. It is the process of manually testing the software functionality for errors in programming. In this method of testing a tester plays the role of an end user and works on all the options provided by the application to ensure correct behaviour.

A. Definition of Manual Testing is a process where in a tester often follows a written test plan that leads them through a set of important test cases. A test case in software testing is a set of conditions under which a tester will determine whether an software application is working correctly or not. In order to fully test that all the requirements of an application are met, there must be at least two test cases for each requirement: one positive test and one negative test

B. Advantages of Manual Testing

Useful for all tasks: Manual testing can be use for both little and huge activities. We can undoubtedly include and evacuate the experiments as indicated by task developments.

- Financially savvy: It is very practical. It is very less expensive to oversee.
- Straightforward: Fresh analyzer can see very effectively the procedure of manual testing.
- Dependable: Manual testing is more solid than computerization testing (by and large robotized not cover all cases). It enables the analyzer to perform all the more specially appointed (irregular testing). A bigger number of bugs are found by means of irregular

testing than by means of mechanization. Manual testing isn't connected with any programming dialects.

C. Drawbacks of Manual Testing

- Time Consuming: Manual testing can be very time consuming as everything has to be done manually.
- It does not have concept of re-usability.
- More human involvement.
- Each time there is a new build, the tester must re-run all required tests - which after a while would become very dull and tiresome.

D. Automated Testing

The guideline of mechanized testing is that there is a rogram (which could be an occupation stream) that runs the program being tried, bolstering it the correct information, and checking the yield against the yield that was participated. When the test suite is composed, no human intercession is required, either to run the rogram or to hope to check whether it worked; the test suite does all that, furthermore, by one way or another monstrates whether the program's yield was obviously. We, for example, have more than two hundred test suites, which can all be run medium-term by executing one occupation stream accommodation direction; after they run, another direction can demonstrate which test suites succeeded and which fizzled.

E. Advantages of Automated Testing

- a) Reliable: Tests perform precisely the same operations each time they are run, thereby eliminating human error
- b) Repeatable: You can test how the software reacts under repeated execution of the same operation
- c) Comprehensive: You can build a suite of tests that covers every feature in your application.
- d) Reusable: You can reuse tests on different versions of an application, even if the user interfaces changes.
- e) Better Quality Software: Because you can run more tests in less time with fewer resources
- f) Fast: Automated Tools run tests significantly faster than human users.
- g) Cost Reduction: As the number of resources for regression test are reduced.

F Disadvantages of Automated Testing

Though the automation testing has many advantages, it has its own disadvantages too. Some of the disadvantages are:

- a) Proficiency is required to write the automation test scripts.
- b) Debugging the test script is major issue. If any error is present in the test script, sometimes it may lead to deadly consequences.
- c) Test maintenance is costly in case of playback methods. Even though a minor change occurs in the GUI, the test script has to be rerecorded or replaced by a new test script.

- d) Maintenance of test data files is difficult, if the test script tests more screens.

VI. LIFE CYCLE OF TESTING

Software testing has its own life cycle that meets every stage of the SDLC. The software testing life cycle diagram can help one visualize the various software testing life cycle phases. They are:

- 4) Test Design
- 5) Test Verification and Construction
- 6) Test Execution
- 7) Result Analysis
- 8) Bug Tracking
- 9) Reporting and Rework
- 10) Final Testing and Implementation
- 11) Post Implementation
- 1) Requirement Stage
- 2) Test Planning
- 3) Test Analysis

VII CONCLUSION

Software testing is an art. Testing is something other than troubleshooting. Testing isn't just used to find defect and correct them. It is moreover utilized in approval, confirmation procedure, and dependability estimation. 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. The stopping time can be decided by the trade-off of time and budget. Or if the reliability estimate of the software product meets requirement.

REFERENCES

- [1] <http://www.logigear.com/newsletter-2006/271-software-testing.html>
- http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/introduction
- [2] <http://www.3riverstech.com/atcasestudy.php>
- [3] <http://www.onestoptesting.com/ntp/testing-process.asp>
- <http://www.softwaretestinggenius.com/download/introlr.pdf>
- <http://www.outsourcingdotnetdevelopment.com/manualtesting.html>
- [4] http://bazman.tripod.com/what_testing.html
- [5] Software Testing Techniques by Shivkumar Hasmukhrai Trivedi
- [6] Software Testing Techniques by Lu Luo
- [7] Research and Development on Software Testing Techniques and Tools by Rasneet Kaur Chauhan* and Iqbal SinghB
- [8] Software Testing Methods and Techniques by Jovanovich, Irena
- [9] https://scholar.google.co.in/scholar?q=research+paper+on+software+testing+techniques&hl=en&as_sdt=0&as_vis=1&oi=scholar&sa=X&ved=0ahUKEwjaw7evyOrMAhVKGZQKHVYqC7AQgQMIGTAA
- [10] Shepperd, M., & Ince, D. (1990). The Use of Metrics in the Early Detection of Design Errors. Proceedings of Software Engineering.
- [11] A Study of Software Metrics Gurdev Singh, Dilbag Singh, Vikram Singh IJCEM International Journal of Computational
- [12] Cem Kaner, Senior Member, IEEE, and Walter P. Bond, "Software Engineering Metrics: Why Do They Measure and How Do We Know?"
- [13] S. S. Stevens, "On the Theory of Scales of Measurement," Science, vol. 103, pp. 677-680, 1946.

- [14] S.S. Stevens, *Psychophysics: Introduction to its Perceptual, Neural, and Social Prospects*. New York: John Wiley & Sons, 1975.
- [15] L. Finkelstein, "Theory and Philosophy of Measurement," in *Theoretical Fundamentals*, vol. 1, Handbook of Measurement Science, P. H. Sydenham, Ed. Chichester: John Wiley & Sons, 1982, pp. 1-30.
- [16] N. E. Fenton and S. L. P. Fleeger, "Software Metrics: A Rigorous and Practical Approach," 2nd Edition Revised ed. Boston: PWS Publishing, 1997.
- [17] IEEE, "IEEE Std. 1061-1998, Standard for a Software Quality Metrics Methodology, revision." Piscataway, NJ.: IEEE Standards Dept., 1998.
- [18] Hericko, M, Rozman, I, Horvat, R, Domjinko, T, Gyorkos, J, "OO Metrics data gathering environment, in: Proceedings of Technology of Object Oriented Languages" (Tools 24), 1998, pp. 80-85