A Survey of Effective and Efficient Software Testing Technique and Analysis

D. Bhatt[1]

[1]L.J.Institute of Engineering and Technology

Abstract— ***the overall scenario of the software testing is to confirm delivery of quality product to the end user. It is required for high quality software that possibilities of errors and bugs occurrence in production are very less. Development of software contains complexity and errors which occur at any stage. Sometimes these errors or bugs could be effect in terms of losses respect to time and money if not identified and solved as early as possible. Software Testing ensures that software meets user specifications and requirements which works as expected as it should be. In any case, the field of software testing has various issues like efficacious generation of test data or test cases, prioritization of test data and so on which need be handled. Testing also include as a major part in Software development life cycle. Effective and Efficient manner in testing can help to fix the issues in time and make software stable.***

Index Terms – Software Testing, Testing, Methods of Testing

I. INTRODUCTION

The testing of software is an important for evaluation of the software in terms of its nature, quality ability to perform functions. Since testing typically consumes 40~50% of development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering [1]. Software testing is one of the important and primary part for achieving high quality software. It can also be defined as the process of verifying and validating software to check whether it meets the technical and business specification as required. In other words testing means verifying the actual outcome with the expected outcome. Effective testing is the relative ability of testing strategy to find bugs in the software product. Test efficiency is the relative cost of finding a bug in the software under specific environment. One of the major issue in software industry is time taken for software testing. Time means money and resources, hence the entire process is costly. Therefore to reduce testing cost it is necessary to reduce manual repeatedly testing time and number of test cases generated for particular system. Testing is a never ending process, we could not say that the particular software is 100% bug free. Therefore we have to define when to start and when to stop testing. For maintain and manage entire testing process it is always required to follow a Software Testing Life Cycle inside the Software Development Life Cycle. Though in the formal technique and verification method, a system still needs to be tested before it is used. Testing remains the truly effective means to assure the quality of a software system of non-trivial complexity [3], as well as one of the most intricate and least understood areas in software engineering [9]. Testing, an important research area in computer science and engineering which become even more important in the future.

Both verification - a form of static testing - and validation - a form of dynamic testing - play a major role in software testing. Verification is performed to ensure that the developed software meets specification and is close to structural testing whereas validation is close to the functional testing and is done by executing test cases in dynamic manner. Testing techniques primary include functional (black box) and structural (white box) testing. Black box testing is based on functional specification whereas white box testing is based on code itself. Verification makes sure that the product is designed to deliver all functionality with respect to the customer’s specification. Verification is done at the starting of the development process. It includes reviews of all customer requirements and meetings, inspection to evaluate documents, code and specifications. Validation is about to determine if the system complies with the requirements and performs functions for which it is requested for and meets the institute’s goals and user needs. Validation is done at the end of the development process and takes place after verifications are completed [2]. Testing can be done either automatically or manually. It is found that automation software testing is better than manual testing. However, very less test data generation tools are commercially available today.

II. CLASSIFICATION OF TESTING TECHNIQUES

Testing is an extremely wide region, which includes many other specific technical or non-technical aspects. For example, specification of requirement, design and implementations, development, maintenance with other management issue. Our focus area is techniques used for testing with respect to latest methods which gives to future direction of this area.

A. The Goal of Testing

Different people can define differently the aim of software testing, but the common thing is to ensure the functioning of software run as expected as specified. Finding bugs or defects which may remain in the program while developing the software, to prevent defects, make sure that the end result meets the business and user requirements are also includes its responsibility.

B. The Testing Platform

Testing is involved in every stage of software development life cycle and it needs to be performed at each level of software development cycle which is different in nature and has different objectives.

Unit testing is performed at the lowest level in testing stages. It tests the basic unit level of functionality of software product or application, which is the smallest testable block, and is often called “unit”, “module”, or “component” testing.

Integration Testing is performed when two or more components or modules are integrated into a larger structure. Testing is often done on both the interfaces of the modules, It occurs after unit testing and before validation.

System Testing executes to ensure the end-to-end quality of the whole system. System test is often based on the functional specification of the system. Non-functional quality attributes, such as reliability, security, and maintainability, are also need to be checked. The purpose of this test is to evaluate the system's compliance with the specified requirements.

Acceptance Testing is run when the developed system is handed over from the developers to the customers or users. The purpose of acceptance testing is rather to give confidence that the system is working as expected behaviour as required.

Unit Testing >> Integration Testing >> System Testing >> Acceptance Testing

C. Static vs. Dynamic Analysis in Software Testing

There are several approaches or we can say analysis of Software testing, performed at different stage during development life cycle. Based on whether or not the particular execution of developed software under analysis is required or not, there are two major analysis of quality assurance activities provided: Static Analysis and Dynamic Analysis.

Static analysis is executed in a very non-runtime environment. Usually a static analysis tool can examine program code for all potential run-time behaviors and search out coding flaws with potentially malicious code. Static testing is carried out without executing the code. This type of testing comes under Verification. It is done before code deployment. It gives assessment of code as well as documentation. Here a checklist is prepared for testing process which include walkthroughs, code review.

Dynamic analysis adopts the opposite approach and is executed whereas a program is in operative mode. A dynamic test will monitor system memory, functional behavior, time interval, and overall performance of the system. This method is not fully dissimilar to the style within which a malicious third party could act with associate application. Having originated and evolved separately, static and dynamic analysis have, at times, been mistakenly viewed con. Dynamic testing is carried out with executing the code. This type of testing consider in Validation. It is done after code deployment, it gives bottlenecks of the software system. Here the test cases are executed have involves functional and nonfunctional testing. There are number of strengths and weaknesses with each approaches to consider.

III. MANUAL TESTING AND AUTOMATED TESTING

If define manual testing and automated testing in a simple manner, executing the test cases manually without any use of tools or scripts is known as manual testing. Taking the support of automation tools, scripts and software to executing the test cases is known as automated testing. Manual testing and automated testing are different segments with respect to requirements and specification. Within each method, specific testing techniques are offered that are black box testing, white box testing, integration testing, system testing, performance testing, and load testing. Some of the techniques are better suited to manual testing, and few of them are best performed through automation tool. Following two sections will discuss the differences in detail with their advantages and disadvantages.

A. Manual Testing

In manual testing code developers or software testers run test cases manually, comparing program requirements and actual outcomes in order to find code defects or bugs. Manual testing is a good fit for smaller projects as well as the projects which have not stable product's specification or changes continuous basis in terms of design, functionalities and financial resources.

Manual testing is not accurate at all times as test cases are executed by human so chance of human error increase and it is less reliable. It's time-consuming as well required investment for taking up human resources. Manual testing is only practical when the test cases are run once or twice, as well with system where designs, functions changes frequently. Software automation tools are expensive. Short-term cost is much lower with manual testing. Here directly involved human observation which may more useful if the goal of product is user-friendly and improved customer experience. Automated tests some of the issues the end user might encounter might be overlooked in an automated test. A human user might handle the program in a way which might give rise to errors that are more likely to be skipped in an automated test but caught with manual testing. It's flexible than automated tools where automated tools run according to a set of rules. Test cases are set up and programmed into the automated tool, and then the tests are run. When a change occurs in the project the whole process might have to be repeated. But with manual testing this can be easily incorporated into testing routine [2]. In terms of result or quick response upon priority issues or bugs manual testing more effective for reporting and fixing compare to automated testing.

Certain tests are tough to be done manually, e.g. low level interface regression testing is extremely tough to be performed manually. As a result, it's prone to mistakes and overseen when done by manually. As major consideration manual testing will be repetitive and boring – nobody needs to stay filling out a similar forms time to time. As a result, several testers have a tough time staying engaged during this process, and errors are a lot of doubtless to occur. Compare to machine driven re run all testing execution with manual testing gives time consuming result.

B. Automated Testing

Automated testing is the process in which testing tools or scripts run tests that repeat pre-defined actions, comparing a developed program functionalities and actual outcomes. If the program functionalities and outcomes result matches your project or product behaves as expecting and bug free, if the results don't match, however, there is an issue that needs to be addressed. You'll have to take a look at your code, alter it, and continue to run tests until the actual and expected outcomes align [5].

Automated testing runs tests quickly and effectively, increases your test coverage, scope and depth of test cases helps to improve software quality. Numbers of test cases which are lengthy that are avoided during manual testing can be performed here. They can even be run on multiple computers with different testing parameters and configurations. Results of execution test cases can see by everyone where as in manual testing can't. Automated testing is more reliable, as it is performed by tools and/or developed scripts. It’s a practical option when the test cases are run repeatedly over a long period of time period.

The onetime cost of automation tools can be expensive. It is important to only use the ones that will give you full, or as close to full coverage, as you can find. When scripts or tools not able to run in such as situation testing could not be performed, few changes at UI level also need to be change in scripts against manual testing. Few scenario may be change takes time to develop its automation script. A considerable amount of time goes into developing the automated tests and letting them run. Different tools have different limitations and they executes with respective environment and configurations.

.

IV.CONCLUSION

This paper presents the introductory survey on software testing, its taxonomy and its behaviour with respects to types, techniques and methods of testing. Here we present different pros and cons how manual testing and automated testing helps to understand the overall testing behaviour for project and product stability, reliability with respect to time and resource, also where in future we can more specifically elaborate with automated tools comparisons.

REFERENCES

[1] Luo, Lu. "Software testing techniques: technology maturation and research strategy." Class report for (2001).

[2] Mailewa, Akalanka, Jayantha Herath, and Susantha Herath. "A Survey of Effective and Efficient Software Testing." The Midwest Instruction and Computing Symposium. Retrieved from http://www.micsymposium. org/mics2015/ProceedingsMICS_2015/Mailewa_ 2D1_41. pdf. 2015.

[3] E. F. Miller, “Introduction to Software Testing Technology,” Tutorial: Software Testing & Validation Techniques, Second Edition, IEEE Catalog No. EHO 180-0, pp. 4-16

[4] Maha alzabidi et. al., “Automatic software structural testing by using evolutionary algorithms for test data generations”, International Journal of Computer science and Network Security 9 (4), 2009, pp. 390 – 395.

[5] Base36,. 'Automated Vs. Manual Testing: The Pros And Cons Of Each'. N.p., 2013. Web. 21 Mar. 2015.

[6] J. A. Whittaker, “What is Software Testing? And Why Is It So Hard?” IEEE Software, January 2000, pp. 70-79

[7] Sharma, Chayanika, Sangeeta Sabharwal, and Ritu Sibal. "A survey on software testing techniques using genetic algorithm." arXiv preprint arXiv:1411.1154 (2014).