

Recent trends of Software Testing Techniques used in IT

Abhay Singh^{#1}

[#] GGSDS College, Chandigarh, India

Submitted: 10-04-2022

Revised: 26-04-2022

Accepted: 28-04-2022

ABSTRACT- With the advancement of new technologies in the software industry, we should focus on the quality of software in order to achieve good developed software. The quality assurance of software in IT sector has been considered as backbone of software development. Quality assurance of software has reached new heights as a result of software applications competencies. Testing can also cut the software cost and decrease the possibility of errors in code. As we all know that testing has played a vital role in software development life cycle. Nowadays in the software industry, we have seen various testing methods and techniques including manual and automated testing. During testing we are using various test cases and test plans to find the errors in code. The goal of this paper is to look at a variety of latest software testing methodologies in order to improve quality assurance.

Keywords: Quality Assurance, Manual Testing, Automated Testing, Test Cases

I. INTRODUCTION:

Software testing is a way to find errors or bugs in the code of software. Testing should ensure that it validates and verifies all sets of user requirements and specifications [1]. It also throws light on other aspects of software like portability, security, maintainability etc. [2]. Software testing aids in the prevention of system problems. It refers to the process of assessing software to determine the source of a mistake. It helped in getting specific goals and objectives. The motive of software testing is to achieve client requirements and needs. In the software industry we find that software testers could do testing rigorously to create test cases, test stubs and test plans to minimize the test sets. Testing can become costly sometimes as it involves risk regarding delay of schedule and overruns of cost. The primary goal of software testing is to deliver a high-quality product in terms of reliability estimation. The secondary goal of testing is to run a program in order to find errors and create a test case capable of detecting the error

that has yet to be identified. This investigation provides stakeholders with precise information about the product's quality. Software testing can also be thought of as a risk-based activity. The software's importance during the testing process cannot be overstated. Testers must understand how to reduce a large number of errors into a manageable set of tests and make sound decisions about the risks that must be tested. The cost of developing software should be significantly reduced if the testing process could be automated. One of the problems involved in software testing is particularly the problem of developing test data. In software testing, test data generation is the process of identifying program input data that meets a predefined testing criterion. A test data generator is a tool that helps a programmer generates test data for a program [10].

II. LITERATURE REVIEW:

In this section Babbar, 2017 emphasized that software testing is an activity that executes the software with the aim of detecting errors or bugs in it. She had focused on important levels of testing techniques. It has been a difficult task to find all the bugs from the code [3]. Berner et al., 2005 proposed that test automation comes from software system development as they are vulnerable to identical issues. We should apply identical, diligence and sensible software engineering practices as in alternative development [4]. Leitner et al, 2007 focused on the two approaches complementing each other: automated testing can perform a large number of tests in a short period of time, whereas manual testing uses the testing engineer's knowledge to target testing to the parts of the system that are assumed to be more error-prone. Despite this complementarity tools for manual and automatic testing are typically distinct, resulting in decreased testing productivity and reliability [5]. Enou et al., 2017 proposed that automated test generation can attain equal liability coverage than manual testing done by software tester in the company. However, in terms of

mutation score, these computer generated test suites are no better than manually developed test suites at detecting faults [6]. Kumbhar, 2020 suggested that it is the process of manually and automatically exercising and evaluating a system or system components to ensure that it meets specified requirements or to highlight differences between original and previous results from various tools. The paper examines a collection of tools that aid in the testing process in a variety of ways [7]. Chauhan et al., 2014 suggested that prevention has to be taken to prevent or reduce the number of errors, to clarify system specifications and to improve system performance. Various methods for avoiding risks and dealing with problems in the future are identified. It demonstrates how the system can be used with varying levels of acceptable risk. It also shows how products are ready for integration or use under special conditions and how functions are demonstrated with special conditions [8]. Bansal, 2014 proposed that when the amount of maintenance and upgrade of existing systems increases so will the amount of testing required to verify systems after changes are made. Despite advancements in formal methods and verification techniques, a system must still be tested before use. Testing is still the most effective way to ensure the quality of a non-trivial-complexity software system, as well as one of the most intricate and least understood areas of software engineering. Testing an important area of research in computer science, is likely to grow in importance in the future [9]. Sharma, 2014 emphasized on manual testing is the process of creating test cases by hand and executing them

without the use of a tool. A human sitting in front of a computer performs manual software testing by carefully going through application screens, experimenting with various usage and input combinations, comparing the results to the expected behavior, and recording their observations. The use of an automation tool to execute a test suite is referred to as automation testing. The goal of automation is to reduce the number of test cases that must be run manually while not completely eliminating manual testing [13]. Asfaw, 2015 proposed that the study shows that automated testing is more efficient, accurate, and cost effective than manual testing. The difference in accuracy between the best manual and the automation test time is 70%. This demonstrates how manual testing is more error prone than automated testing. In fact, as academic skill sets improve, the number of errors committed during the manual testing process decreases, but it never reaches precision [14].

III. METHODOLOGIES :

The realization of software testing could affect the quality assurance of software prominently. After the completing the coding of the software, it's compulsory to test the software to find errors and debug them before the release. It is not possible to identify and debug every error in critical software at each stage, but we strive to eliminate all errors as much as possible. We usually divide the testing techniques into two parts:

1. Static (Manual) Test
2. Dynamic(Automated)Test

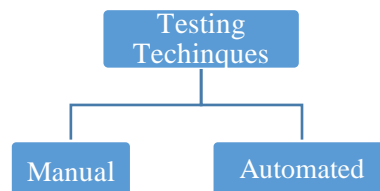


Fig. 1: Testing techniques

A. Manual Testing (Static)

Manual testing is the main type of testing. In manual testing, the tester runs the test case manually. Before running automated tests, you must first manually test each product. Manual testing helps you find product errors. If your project is in early development, you can choose manual testing as manual testing. Even if the project is a short-term project, especially if you

want to test the user interface visually. If your project is short-lived and scripting takes a long time, manual testing is recommended.

1.Steps of Manual Testing

Step1: To successfully complete the manual test, you first need to understand the requirements document. Therefore, by understanding the requirements, you can learn what you need to test

and what you need to classify as a bug or bug. This is an important part of the test.

Step2: Create a test case. Once you understand the requirements, you can create test cases. These test cases basically guide the sequence of tests and test functions and various scenarios described in software products or applications. Creating good test cases is very important as it helps to run all test cases very smoothly and ensure good test coverage.

Step3: Once you have created the test case after that you just execute it. Once the test cases have been created and the test environment is properly prepared, start testing. You can track test cases and mark each test as pass, fail, or skip. It is important to take notes when testing manually. This is very important. The facts cannot be ignored just because the test fails.

Step4: The next step is to log the appropriate bug report. In addition to the actual testing as a tester, I

am also responsible for logging errors. Therefore, if you find a bug, you should provide the development team with log information about the bug. Writing a good bug report will help you and your team later. You can save time later in answering these error questions. In other words, there is basically a document to reference. A good bug report should include the title, steps to fix the bug, expected and actual results, and related attachments such as screenshots and recordings to help the development team understand the bug.

Step5: You need to create a detailed test report. After running the test, it is good to know the test at a very high level. For example, the number of tests that passed, the number of tests that failed, the number of tests that passed, the number of test that were skipped this way, and so on. This procedure is very simple.

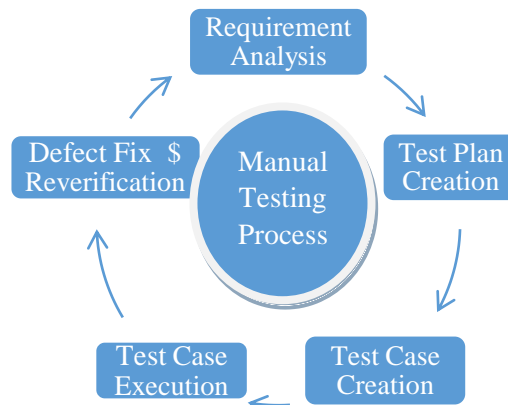


Fig. 2 Manual Testing

B. Automation Testing (Dynamic)

Automated testing uses automated tools. The goal of automated testing is to simplify as much of the testing effort as possible with a small number of scripts. For example, regression testing consumes a significant portion of the resources of a quality assurance (QA) team, this process may be a good candidate for automation. Automated testing should aim to provide better accuracy, efficiency, and quality assurance at the end [10]. Automation entails the use of third-party software to generate the necessary script to control the execution of automated tests. The test suits is run by an automated tool. Manual testing requires the tester to sit and run the test suits one at a time. The tester needs to create a test script. Automated tests are faster than manual tests. Random testing is not allowed for automated testing. Automation tests are performed using tools. This field requires highly

qualified testers with extensive knowledge of the field. Automated tests are called validation tests. For the purpose of modification, the tester needs to modify the script. Tests can be run on different operating platforms, saving time. Testers should have a good knowledge of programming languages to perform automated tests. Automated testing is cheap. Automated testing is more powerful than manual testing. Tools and automation engineers need more investment. Automated testing requires less complex test setup or environment setup to test the script.

2.Steps of Automated Testing

Step1: Testers should be proficient to have expertise in automation tools.

Step2: You need to select the appropriate tool to test.

Step3: Analyze different applications and make decisions about the best tools for automation.
 Step4: Training should be provided to other team mates.
 Step5: Evaluate suitable framework for automation.

Step6: Building the proof of Concept.
 Step7: Develop the Automation framework.
 Step8: Develop Test Script.
 Step9: Execute and Analyze the script.
 Step10: Report a bug

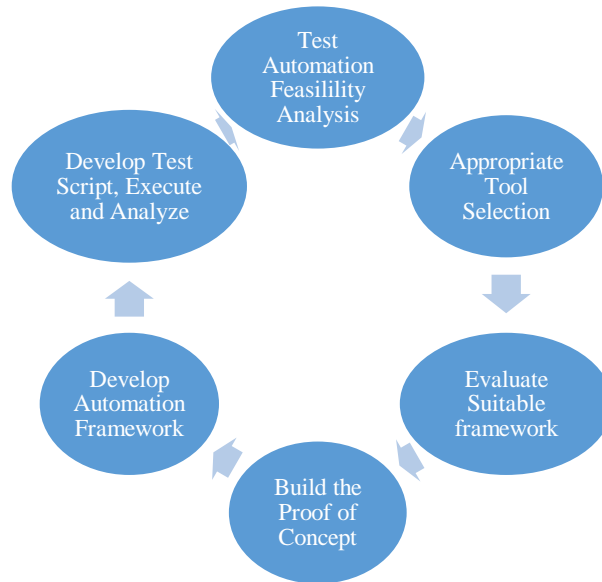


Fig. 3 Automated Testing

C. Comparative between Manual and Automated Testing

Speed: The primary distinction between manual and computerized systems is speed. Accounting software processes data and generates reports much faster than manual systems. Calculations are performed automatically in software programs, reducing errors and increasing efficiency. Once data has been entered, reports can be generated by simply pressing a button in a computerized system. Manually testing the software for flaws is done by testers. It necessitates that a tester assume the role of an end user and use the majority of the system's features, application to ensure that it behaves correctly They adhere to a written plan. A test plan that guides them through a series of critical test cases [12].

Cost: Another distinction between manual and computerized systems is cost. Manual accounting on paper and pencil is far less expensive than a computerized system, which necessitates the purchase of a machine and software. Training and program maintenance are two additional costs associated with accounting software. Costs for printers, paper, ink, and other supplies can quickly add up.

Backup: A third distinction between manual and computerized systems is the ease with which a computerized system can be backed up. In the event of a fire or other disaster, all transactions can be saved and backed up. You can't do this with paper records unless you make copies of every page, which is a time-consuming and inefficient process.

D. Benefits of Automated and Manual Testing:

In order to compare two alternatives based on opportunity costs, we must assess the benefit of each option, i.e., automated test case versus manual test execution. The benefit of executing a test case is usually determined by the information provided by the test case. The typical data is an indication of a flaw. There are, however, additional information objectives for a test case (e.g., to assess the conformance to the specification). All information objectives are important in order to support informed decision-making and risk mitigation. It provides a comprehensive discussion of what factors constitute a good test case. We simply assume that the advantage of a test case is probably at par with its potential to contribute to risk mitigation, whereas automated and manual testing are both beneficial. We simply assume that the

benefit of a test case is proportional to its potential to contribute to risk mitigation, with automated and manual testing addressing different types of risk. One common distinction is based on the discovery that automated testing best addresses the regression risk, i.e. defects in modified but previously working functionality are quickly detected, whereas manual testing is appropriate for exploring new ways in which functionality can be broken. As a result, the goals of automated and manual testing are typically distinct [15]. Automation testing has major advantages over manual testing: efficiency and accuracy. According to the research, as efficiency increases, costs decrease, and a higher return on investment is unavoidable. The research, on the other hand, shows that automated testing is more efficient, accurate, and cost effective than manual testing. The difference in accuracy between the best manual and the automation test time is 70%. This demonstrates how manual testing is more error prone than automated testing. In fact, as academic skill sets improve, the number of errors committed during the manual testing process decreases, but it never reaches precision [14]. The proposed solution for this problem is automation techniques; this study presents a testing roadmap and argues that software quality and cost are intertwined, reliant on effective testing. This study explains why less error in software is always better improved quality as a result of automated testing. Automated testing provides coverage (branch, code, statement, and path) to quality. It focuses on testing time and contends that automated techniques reduce testing time particularly in regression testing. Automated testing allows businesses to reduce testing costs, "The reusability of the testing process saves time and increases the productivity of the testing process. Tests automation can reduce the amount of time spent on testing. Automation testing can improve the detection of flaws in the testing [16].

Enhancement in Testing processes Test Automation

The most important improvement in the testing process is to direct the testing process to test automation. That is, run the test process using specific software and compare the actual results with the expected results. Test automation techniques can save your time because they can save you a lot of tedious manual testing time. With SDLC, test automation is done both during implementation and during testing. Test automation is practiced around the world instead of manual testing because it saves a lot of time by running the testing process in less time. Test automation has

taken over the manual testing process by reducing its need and revealing the amount of bugs and defects that the manual testing process cannot recognize. Regression testing is one of the most important types of tests and can take a lot of time to run manually. Usually, after fixing errors and bugs, we test whether the software or application is working properly. This is because the code or application error or error rate can be even higher after debugging. Therefore, to avoid the time required for regression testing, a set of automated test suites is created and a regression test suite is formed for this purpose. Test automation also helps find problems much earlier, saving a lot of change costs and energy later.

IV. CONCLUSION :

We would like to discover that software testing is a critical activity in the Software Development Life Cycle (SDLC). We can never portray a product as "Perfect." Testing is a continuing process. Because it is a time-consuming and intensive process, upgraded techniques and innovative methodologies are required to maintain the software quality. This necessitates the use of Automated Testing both before and during the testing process. This paper aims to describe both manual and automated software testing techniques in detail, recent trends used in the testing especially in the IT sector to achieve software testing goals and objectives. Software testing is frequently less formal and thorough than it should be. Those involved in software testing must work together to perform it effectively and efficiently.

REFERENCE

- [1]. Arumugam, A. K., "Software Testing Techniques & New Trends." International Journal of Engineering Research & Technology (IJERT) 8.12. 2019.
- [2]. Anwar, N. and Kar, S., "Review paper on various software testing techniques & strategies." Global Journal of Computer Science and Technology. 2019.
- [3]. Babbar, H., "Software testing: techniques and test cases." Int. J. Res. Comput. Appl. Robot 5.3 : 44-53. 2017.
- [4]. Berner, S., Weber, R., and Keller, R. K. "Observations and lessons learned from automated testing." Proceedings of the 27th international conference on Software engineering. 2005.
- [5]. Leitner, A., Ciupa, I., Meyer, B., and Howard, M. "Reconciling manual and automated testing: The autotest experience." 2007 40th Annual Hawaii

- International Conference on System Sciences (HICSS'07). IEEE, 2007.
- [6]. Enoiu, E., Sundmark, D., Čaušević, A., and Pettersson, P. "A comparative study of manual and automated testing for industrial control software." 2017 IEEE International Conference on Software Testing, Verification and Validation (ICST). IEEE, 2017.
- [7]. Kumbhar, Manisha. "Performance Testing Tools: A Comparative Study of QTP, Load Runner, Win Runner and JUnit.", 2020.
- [8]. Chauhan, R. K., and Singh, I. "Latest research and development on software testing techniques and tools." International Journal of Current Engineering and Technology 4.4: 2368-2372. 2014.
- [9]. Bansal, A. (2014). A Comparative study of software testing techniques. International Journal of Computer Science and Mobile Computing, 3(6), 579-584. 2014.
- [10]. Jamil, M. A., Arif, M., Abubakar, N. S. A., & Ahmad, A. Software testing techniques: A literature review. In 2016 6th international conference on information and communication technology for the Muslim world (ICT4M) (pp. 177-182). IEEE. 2016.
- [11]. Korel, B. Automated software test data generation. IEEE Transactions on software engineering, 16(8), 870-879. 1990.
- [12]. Kaur, M., and Kumari, R. Comparative study of automated testing tools: Testcomplete and quicktest pro. International Journal of Computer Applications, 24(1), 1-7. 2011.
- [13]. Sharma, R. M. "Quantitative analysis of automation and manual testing." International journal of engineering and innovative technology 4.1. 2014.
- [14]. Asfaw, D. Benefits of automated testing over manual testing. International Journal of Innovative Research in Information Security, 2(1), 5-13. 2015.
- [15]. Ramler, R., and Wolfmaier, K. Economic perspectives in test automation: balancing automated and manual testing with opportunity cost. In Proceedings of the 2006 international workshop on Automation of software test (pp. 85-91). 2006.
- [16]. Taipale, O., Kasurinen, J., Karhu, K., and Smolander, K. Trade-off between automated and manual software testing. International Journal of System Assurance Engineering and Management, 2(2), 114-125. 2011.