Gyancity Journal of Electronics and Computer Science,
Vol.5, No.1, pp. 47-55 March 2020
ISSN:2446-2918DOI: 10.21058/gjecs.2020.51006

Automated VS. Manual Testing: A Scenario Based Approach Towards Application Development

MEHWASH RAFIQ[1],DR. REHAN ASHRAF[2], HARIS ABID[3]

[1]*Department of Computer Science, University of Agriculture, Faisalabad, Pakistan*

[2]*Department of Computer Science and IT, National Textile University, Faisalabad, Pakistan*

[3]*Department of Computer Science, University of Agriculture, Faisalabad, Pakistan*

mehwashrafiq@hotmail.com, rehan@ntu.edu.com, harisabid23@yahoo.com

Abstract

In order to produce the quality product, assessment is made during and at the end of Software development process to check whether software is error free and to ensure whether specified requirements of the Software are met. This practice is called Software Testing. In order to cope time and resource constraints of this modern era, now a day new way of testing called automated testing is working against manual testing. In automated testing, pre-scripted tests are executed by software tool. A software tool is used to test or check software execution. Actions are pre-recorded and predefined then playback is performed through an automated testing tool, comparison of the results with the expected behavior is made and the success or failure is reported. There exists different mode of testing like unit testing, integration testing and functional testing and each one works in different perspective. Not all types of testing can be automated but a few can be. In this paper various types of testing that can be automated are discussed and how they work in different scenario. The way in which different automated testing tools perform these testing are discussed in this paper.

Key Words: Automated Testing, Types Of Automated Testing, How Automated Testing

I. INTRODUCTION

Testing is an investigation activity most done to check the quality of a product. It makes sure that the product is providing intended services with quality. It is mostly considered the last activity in the development life cycle but this perception is wrong. If we take it at the last of and remove the fault. So it is a preferable and adopted approach to take testing as a continuous development, consequences of product fault and failure may be severe and take more cost and time to detect activity in the whole development life cycle. There may exist various types of testing that work in different ways and find different types of faults. (Importance of testing, types of

ISSN:2446-2918DOI: 10.21058/gjecs.2020.51006

testing). Testing is a tedious task and it may take a large amount of resources in terms of time and cost. It is not wrong to say that for a successful software project, planning and testing may take half of the whole development life cycle time. After utilizing a large amount of resources on testing, project managers or customers cannot rely on its results or coverage because all testing processes are manual and done by human experts and human beings can make errors. (Why automated testing) Therefore need for automated testing may exist. In automated testing, pre-scripted tests are executed by software tool. A software tool is used to test or check software execution. Actions are pre-recorded and predefined then playback is performed through an automated testing tool, comparison of the results with the expected behavior is made and the success or failure is reported.(benefits of automated testing). Due to large and difficult processes software testing looks for to automate because testers want to make testing process faster and cheaper and in automation as much of the test process as practical and advantageous, and more reliable. To this point, a procedure is required through which the correct and incorrect behaviors of the System Under Test (SUT) can be differentiated. That procedure is named as test oracle [4]. Things are made easier using automated testing as testing effort is reduced as much as possible because a minimum set of screenplay is used as a basis for testing. Automation of the testing process is the best option where unit testing consumes a huge proportion of a quality assurance (QA) team's resources. The capability of automated testing tools is to execute tests, outcomes are reported and results are compared with earlier test runs. Repetition is the key feature of automated testing as automated testing tools can run tests repeatedly, as and when needed. Two roughly divisions can be made of testing work which are automated and manual testing. In manual testing, the role of an end user is played by human tester and the attribute of given software under test (SUT) is executed and it is made assure that its behavior is according to expectations. Completeness of testing is granted by following a written test plan through which a set of test cases is executed by tester. The automation of software testing activities is the actual purpose of automated software testing. In more precise expressions, special software is used in test automation (not a software which is being tested) the execution of tests are controlled and the comparison between actual outcomes and predicted outcome is made [8]. As the software industry is being developed with time, accepted approach is to adopt automated testing gradually. A lot of advantages can be gained by using automated testing. It opens ways of using less time and fewer resources, more can be got; only by making a slight revision, test components of automated test cases can be used repeatedly which makes process easier and to find bugs earlier is possible and that is why less cost is required to fix them; more reliable test results can be gained. In many scenarios, for manual testing to complete the testing process perfectly is very difficult or even impossible, but it can be completed easily through automated testing [5]. There are various types of testing like unit testing, integration testing, system testing, sanity testing, smoke testing, interface testing, regression testing, acceptance testing, coverage based testing, stress testing and load testing, white box testing and black box testing. All these types of testing

work in different ways and consider different objects like unit testing focus on a single unit of software project.

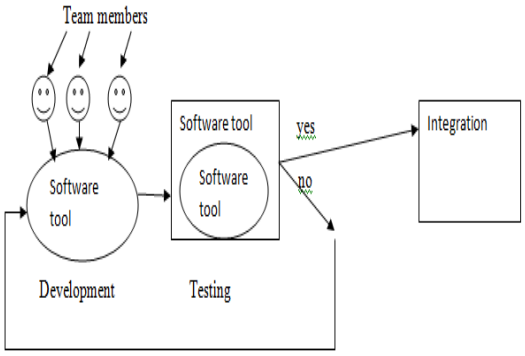

Fig. 1 Automated Software testing process

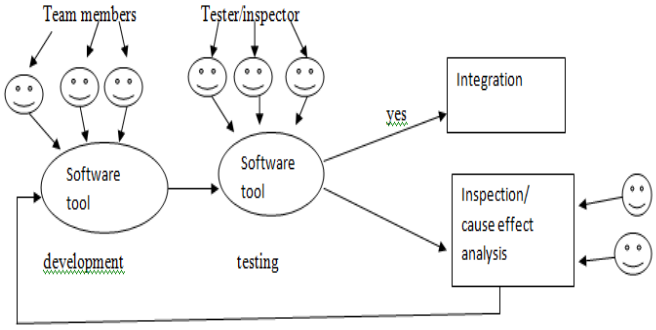

Fig. 2 Manual Testing process

In figure 1 automated testing process is opened in which different team members with different skills develop software tools and a software tool tests the developed software. Result of the automated testing process is in binary nature whether software tools are accepted or goes for other activities after rejection. In figure 2 manual testing process is shown in which team with specialized skill develop the software and people with testing skills tests the software tool. Each test follows a different procedure to evaluate an object to be tested. Similarly, the automation process for these tests works differently. Different types of testing can be automated but not all types can be automated because you cannot automate people reaction or emotions in your software and you cannot automate things you do not think of. In this paper, different types of testing that can be automated are presented and difference between their manual process and automated process is also presented. Different types of testing that cannot be automated is also discussed and reason why they cannot be automated is discussed here. First we discuss the types of testing that can be automated and compare their manual procedure to automated procedure. Second testing types that cannot be

automated are discussed and the reason behind their non automation behavior is also discussed.

II. TESTING THAT CAN BE AUTOMATED

2.1 Unit testing

A smallest part of an application by using which a single operation can be performed independently and individually is known as a unit. Investigation is made to check the performance of each individual unit in unit testing. It is the initial level of testing in the development process as individual error free unit may lead to successful software projects. Unit testing can be manual or automated. In manual unit testing is quite subjective and done by human experts. Experts according to their expertise evaluate a unit according to its requirements. Sometimes it can be costly in term of time and finance when you make a single change in source code a unit you need to execute unit test again which may take time and cost. So unit testing is mostly automated. A piece of code is written to test a piece of code. A common practice with the aim to improve software quality in industry is called unit testing. However, a challenging and tedious task is to write an effective unit test. Techniques such as dynamic symbolic execution (DSE), random testing, search-based software testing (SBST) or hybrid approaches are commonly used automated test generation techniques. [2]

2.2 Integration testing

After unit testing another testing usually takes place commonly known as integration testing. As its name shows it integrates or combines different units (which are tested in unit testing phase) and on this combination, a testing is performed to determine whether these units coordinate or work well with others units or in a combination with others. Support to continuous integration development environments inherent several challenges. The expectation must be conform by developers that changes will be committed by them frequently, typically once per day at a minimum, but more often usually. Atomic commits must be supported by version control system, in which a single commit operation holds a set of related changes; builds from being attempted on partial commits are prevented in this way. Automation in testing must takes place and to continue to operate in the existence of significant levels of code churn, robustness in the infrastructure must be enough.[7]

2.3 Functional testing

The working system as a whole is checked in functional testing. The functionality, accessibility and usability of the whole system is examined. In order to test it against functional requirements, software is executed. Easy approach is to adopt manual testing. Functional tests can be run by using both manual and automated tools. The simplest method of doing functional testing is manual test execution, and it tested the

software in the same way that a user would like to use it. That is why the involvement of customers or users is highly appreciated in functional testing. The primary challenge of manually execution functional tests can be very time-intensive and it is the biggest challenge in manual functional testing, and other activities may get disturb because huge amount on resources is utilized in testing in the form of manpower and time, which reduce the development of new features and effects the improvement of product quality. UXFunctional testing is a form of automated testing which investigates applications functions, or, in other words, it checks the relation of user to product. Selenium is a Popular tool which is an Open Source Functional Testing Tool. Watir for functional testing of web applications, Watir is used. Tests executed at the web browser are supported by it and ruby scripting language is used by it. Functional testing of the GUI is supported by TestComplete. its repetitive aspects are automated and flexible, filtered results are produced. During the life of a project, the simplest functional test should be appropriate and without human intervention measuring results against an already-validated standard output should be the capability of this test. To congregate these criteria, TestComplete's function Test Log is designed. Creating and running all of your tests from a central platform is possible through Spiratest: manual and automated both are applicable in this sense. You need to be careful for some points when you design manual test script for functional testing: reusable test cases should be written so that these can be used repeatedly. High attention is paid to test data because tester relies on test data in manual functional testing. A thoroughly intensive testing is required for a business critical product like internet banking websites to locate any enduring bugs lasting in the code as a huge business loss may be possible due to small failure. Functional testing here is used to locate and remove some of the possible defects which may cause failure. The development of scripts is included in automated testing that not only saves resources in the form of manpower or time when applications are updated, but also makes the testing process fasten. The quality of test cases highly influences the quality of testing. To automate the regression testing scenarios is usually a good decision but it is not possible to automate all test cases [7].

2.4 Regression testing

Regression testing takes place where software is updated in response to change in requirement or to enhance performance or to detect failure. Regression testing evaluates the consequences of change and how change affects the whole system. Because the test is executed repeatedly after making a change the best choice is to adopt automated regression testing in which only affecting part of software would be re executed or tested. In order to exploit the velocity of fault detection, test cases are reordered using test case prioritization techniques. Reordering test cases is the best approach in the area of regression testing. The speed of faults detection is evaluated using different measures. To search out such ordered test cases that shall quickly imitate the failures is the target of prioritization. In automated program repair framework, Invalid patch detection rate can be maximized if prioritization of test

ISSN:2446-2918DOI: 10.21058/gjecs.2020.51006

cases can be redefined in such a manner of scheduling test cases. Cost can be saved in terms of time and money and efficient automated program repair can be obtained by increasing the failure detection rate. [9] Following concepts are usually automated in automated regression testing

- Testing processes, when a software is updated that testing process would be responsible for correctly recompiling it.
- To control the workflow a testing process should be automated which can find the core logic of the software and check the functionality of software against that logic.
- To test all other supporting services that balance the core software Tools, an automated testing process should be developed.

2.5 Black box testing

Black box test checks the behavior of software projects, which is why it is called behavioral testing. It does not check the internal behavior of the system rather focus on input and output relation to check whether input is providing corresponding required output. Output in response to an output is usually known or is expected. Software is executed to check if given input providing the expected output otherwise data analysis is made to know dependencies and data flow analysis. Cause and effect graph is also made to determine what events made what types of consequences. Difference between the actual output and expected output is measured through sensitivity analysis. Predictive analysis is made to get expected output based on the historical available data. It is proven that black box testing involves a large number of other activities like predictive analysis to determine expected outcome after that software is executed and sensitivity analysis measure a difference between expected output and execution output. If the difference is high cause and effect analysis determine relation between events and its corresponding results. The best choice is to automate all these procedures. It may save resources and produce better, reliable and error free results.

III. TESTING THAT CANNOT BE AUTOMATED

Testing activities of all testing types cannot be automated. Testing that is performed on an object that is not performing a clear function cannot be automated. These tests are usually subjective in nature. Similarly human behavior or thinking, perception cannot be automated. Where automated testing cannot work, manual testing can be used. Even automated testing cannot replace human experience or analytical skills in many scenarios. Creativity cannot be embedded in automated testing. Different types of testing that are usually performed manually. Some of them can be automated but not fully but partially:

- UX tests
- UI tests

ISSN:2446-2918DOI: 10.21058/gjecs.2020.51006

- Exploratory tests
- API tests

3.1 Exploratory Testing

When exploratory testing is approached, creativity and analytic software tester are used throughout the process. In accumulation, learning about the application is done by tester. After that tester would be able to comprehend all the unforeseen aspects of the product using his experience and analytical skills. Automation cannot be done for performance testing. Because the automation process cannot detect logical flaws. Testing can be automated only for those aspects which can be predicted. Automation is usually related to supervised learning in this sense. To check an object automated testing is the best choice. Only need to design and automate the testing process, it means it takes initial level efforts after that results in the form of pass or fail by testing process giving input. New information cannot be obtained besides pass or fail Outside of pass or fail, in this sense it can be called binary automation testing. For this reason it is not sufficient because it provides a relatively narrow scope of risk. The confines of automated tests can be crossed by using the abilities of exploratory testing. To discover new defects and testing usability substantial expertise and business knowledge of testers are used. Program is explored in a deeper and insightful manner. After discovering a new defect by tester using the course of exploratory testing. Automation can be performed on some action to handle the discovery of defects and for future concerns. In contrast, exploratory testing examines an application in a structured way relying on the tester's skill, experience, and perceptions. Weaknesses or limitations in the application are tried to be captured by the tester. A continuous learning process is involved in exploratory testing. Learning is all about test design, and execution of test cases for a software product. Due to high use of agile development, exploratory testing is becoming a more desirable testing scheme. However, software companies often overlook exploratory testing is often ignored by tester due to its high resource demand. Moreover, exploratory limited test coverage is provided by exploratory techniques compared to other testing methods. [3]

3.2 Usability Testing

A human-driven evaluation about the usability of a product is known as user experience (UX), it is about the usability of application, or feature. It means that automation cannot be used to evaluate the UX very accurately because human observation aspects about products are taken out. Internal usability testing is worthy in the sense, that the role of the end-users role is played by researchers and they make an attempt to use the designed system as the way the user would like to use. Another form of usability testing vivo usability testing exists using this the speed of rapid iterations is scarified and the holistic system is explored in a accepted scenery over time.[6]

3.3 UI Testing

It is another form of testing closely related to user interface. User interface is the initial and most important thing to which users first interact. Each user interacts with the interface according to his knowledge and requirements. User knowledge and usability cannot be automated. Users with different background knowledge are required to check UI testing. This aspect of UI testing is difficult or impossible to automate

3.4 API Testing

Back-end endpoints and integration should be checked and API testing is the best approach for this testing. Some front-end testing needs can be reduced by having a solid API testing framework. For mobile applications a solid cross platform is created.

3.5 Performance Testing

As its name suggests Performance testing preliminary check the performance of computer, network, software or device. Under workload speed, responsiveness and stability is determined. It checks whether increasing workload, how much performance of product is affected. It can be automated and can be done manually or both approaches can be combined to get maximum results. Functional and Performance testing are basically two classification tests. Parameters like turnaround time, reliability, and load capability are analyzed in performance testing. It is tested that these parameters are according to the client potential. Different types of checking are aimed at functional testing, it is tested whether provided functions are correct and according to intended requirements of business [1].

IV. METHODOLOGY

A critical review of a survey is conducted to obtain the knowledge about automated testing. Total 30 papers about automated testing are collected and in which 17 papers are used for the literature survey. Other papers are extensions of these 17 papers and some of them are repetitive. Two professors of high experience of academia and industry are also concerned.

V. CONCLUSION

There exist various test types which investigate different aspects of a software and work in different manners. New approach of testing automated testing is taking a huge place in the software industry due to its automatic behavior and resource saving nature. Automated testing reduces time and saves manpower and provides reliable results. Automated testing comprises many benefits but it is not feasible in every situation where you want to subjectively evaluate a product. Automated testing can also be called because it gives results only in the form of pass or fail. Automated

testing is feasible in a situation where outcome is predictable that is why it is closely related to supervised learning. Similarly the importance of human skill, experience and analytical abilities cannot be denied. Both testing approaches are good but no one can replace others.

VI. References

1. Abha *et al*.,."A Comparison of RANOREX and QTP Automated Testing Tools and their impact on Software Testing". *International Journal of Engineering, Management, Sciences,* Vol. 1, No. 1, 2014.
2. Andrea *et al*.,. "Generating TCP/UDP network data for automated unti test generation". *ESEC/FSE 2015 proceedings of the 2015 10th joint meeting on foundations,* ISBN:978-1-4503-3675-8, 2015, pp 155-165.
3. Chirstopher and Hyunsook "Model-Based Exploratory Testing: A Controlled Experiment ". 2014 *IEEE 7th international conference on software testing,* ISBN 978-1-4799-5790-3., Acc No. 14363377, 2014.
4. Earl *et al*.,. "The oracle problems in Software testing". *IEEE transaction on software engineering,* Vol. 41, No. 5, 2015, pp 507-525.
5. Kan *et al*.,. "A method of minimum reusability estimation for automated software testing". *Journal of shanghai jiaotong University science,* Vol. 18, No. 3, 2013, pp 360-365.
6. Rod *et al*.,. "The Writing Pal Intelligent Tutoring System: Usability Testing and Development", *Computers and composition,* Vol. 34, 2014, pp 39-59.
7. Sebastian *et al*.,. "Techniques for improving regression testing in continuous integration development environment". *FSE 2014proceedings of the 22nd ACM SIGSOFT international symposium on foundations of software engineering,* ISBN:978-1-4503-3056-5, pp 235-245.
8. Vahid and Mika, "When and what to automate in software testing". *Information and software technology,* Vol. 76, 2016, pp 92-117.
9. Yuhua *et al*.,."The Efficient Automated Program Repair through Fault-Recorded Testing Prioritization". *2013 IEEE*
10. *international conference on software maintenance,* ISBN:1063-6773, pp 180-189. *s*