Traditional V/S Optimize Techniques of Software Testing

Kavish Aggarwal
Student, Amity Institute of Information Technology (AIIT)
Amity University
Noida, India
Kavish.aggarwal@yahoo.com

Richa Singh
Assistant Professor, Amity Institute of Information Technology (AIIT)
Amity University
Noida, India
Rsingh10@amity.edu

***Abstract*-In today's world testing is one of the most important thing that is needed to be performed before delivering a product to a user or a customer. Earlier the testing was done through manual and traditional methods where a coder or a tester was required to sit and test software or applications for any bugs or drawbacks in software. But nowadays with the help of automation testing tools and methods, it has become easy and fast to detect bugs and to overcome it.**

This paper throws light on the new advancement that has occurred over a period of time, from manual software testing to automation software testing. It also tells about the different tools used in automation testing.

***Keywords*- Testing; Manual; Automation; Tools**

I. Introduction

This paper tells us about the new technical developments that have taken place in the field of software testing. The newly emerged progress from the phase of manual testing to automation testing has been discussed in this paper.

In this paper some of the most popular and widely used tools for automation software testing has been selected and are gone through a detailed analysis. This analysis gives an overview of what features one tool posses which the other tool lacks. With the help of this paper the readers would be able to make a proper judgment about which tool should they use to get a optimize result.

II. Literature Review

In [1] the study tell us a comparison of 17 software testing tools which focused on features like generating the test cases automatically, test report customization, debugging assistance etc.

In another paper i.e. [2] the study suggests the manner of software testing of twelve different software organizations. It also tells what percentage of resources was used in the testing phase and also about how the process of testing is regulated.

Software testing is a crucial part of software engineering but earlier when software testing was introduced; some of the software organization does not considered it as much important. In [3] there is a case from 2002 where software organizations in the province of Alberta dose not paid much attention on the testing of their software as compared to other organization in the United States. This resulted in two major setbacks which were: firstly the product which suffered lack of testing also had reduction in the quality; and the second problem was that these organizations suffered problem in adopting the upcoming methodology.

One of the way's to test a software is through unit testing, in which independent units or similar kind of units are selected and tested together. [4]

From traditional software programming the world is now more focused on web based software and applications and therefore it is also the responsibility of the organizations to test these web based software before delivering it to the end users. In [5] the author tells us about some of the distinctive technology for building modern web based software applications and many more about the web based advancement.

In testing software we generally create test cases but in [6] the author introduces a new technology where the test data which we generate during testing are automatically created. This technology is based on mutation analysis.

In [7] the author tells us how he was able to bring together Tabu Search technique and Korel's chaining approach and was able to come up with a productive testing technique. This technique helps to generate test data to obtain branch coverage in testing.

Internet of Things or IOT is the latest trend and since it is new emerged technology in market, it is the most likely to catch bugs. For this in [8] the authors have created Rulede, a tool to find bugs for IOT. It is built on LLVM compiler and is been tested on Contiki, TinyOS and RIOT Os (open source operating software for IOT).

The ultimate goal of software testing is to provide the end user a fully functional software with no or as minimum as possible defects or bugs. This is the primary responsibility of the tester to ensure that user should not have any issues regarding any defect or failure. In [9] the author tells what is testing and how important it is to deliver a quality product to the user. For this there are various methods, which can help a tester to explore for various defects and bugs. Some of the necessary methods are also discussed in this paper. As this a vast field to explore with fast changing trends, it is crucial for a tester to remain fully up to date.

A software tester always put its level best to deliver fully integrated and bug free software. To deliver this he uses testing technique and attributes. But to do so the tester must identify what type of technique can be applied for checking a particular quality attributes [10].

III. Manual Testing

In manual testing a software testing team is appointed apart from the development team (in some cases are same), who are responsible for the execution of test cases for the software developed with manual testing tools.

This type of testing is very time consuming as a tester has to analyze each and every step of testing. It is feasible for small scale projects where the test cases are run once or twice but in large scale projects it is not advisable. It is been observed that in certain cases bugs and defects which occur at the user end and are failed to be sported by automation testing, such case are to be best handled through manual testing.

But at times manual testers do get bored specially in regression test cases where the same procedure is to be tested again and again. For such cases the testers prefer automation testing.

IV. Automation Testing

In automation testing the testing is done with the help of specially developed software testing tools which executes the test cases for the newly developed or existing software. Some of the most commonly automation testing tools are:-

1. Selenium
2. Testing Whiz
3. Unified Function Testing (UFT)
4. Sahi
5. Ranorex
6. Watir
7. Cucumber
8. Test Studio
9. Silk Test
10. Robotium

This type of testing consumes less time as compared to other methods and is more efficient. Now a day's most of the organizations prefer automation software testing since it is more accurate as compared to other traditional methods.

The automation testing is best for regression test case as it can execute the same cases repeatedly, much faster than manual testers and also doesn't get bored.

The only drawback of automation testing is that it cannot determine whether the software developed is user friendly nor not. This is important because ultimately the end user is a human and it is essential for the product to be user friendly, so that it can gain more popularity which could eventually generate more profitability.

V. Difference between Manual Testing and Automation Testing

S.no.	Manual Testing	Automation Testing
1	In manual testing the presence of a human is required	In automation testing all the execution is done by automates tools
2	It is more time consuming	It is less time consuming
3	It is less reliable	It is more reliable
4	It is best suited if the testing is to be done once or twice	It is best suited if the testing is to be done multiple times
5	We have to invest on human intelligence again and again	We have to invest on the testing tools once and then only on the maintenance
6	It lack the facility of real time data	It does not lack the facility of real time data
7	Only one software can be tested by one tester at one time	Multiple software can be tested at one time
8	The response that will be generated will also suggest weather it will be user-friendly	The response that will be generated will only tell about the errors in the program

VI. Different Automation tools

S.no.	Tools	Supported Scripting Language	Programming / Scripting Language used to created	Software/Web/ Mobile Testing	Supported Framework	Open source or not
1	Selenium	C#, Groovy, Java, Perl, PHP, Python, Ruby and Scala.	Java	Web Testing	TestNG, JUnit, NUnit, Behat + Mink, unittest, pyunit, py.test, robot	Open Source
2	Testing Whiz	Java Scripting	Java	All	Android Application	Not Open Source
3	Unified Functional Testing (UFT)	VB Script, java script and Windows shell script.	VB Script	All	Data-Driven, Hybrid, Linear, BPT, Keywords-Driven	Not Open Source
4	Sahi	Java	Java and Java Script	All	Excel Framework	Open Source
5	Ranorex Studio	VB. Net and C#	C# and VB.Net	All	Keyword Driven	Open Source
6	Watir	Ruby Programming Languagr	Ruby Programming Language	Web Testing	BDD	Open Source
7	Cucumber	Ruby Programming Language	Ruby Programming Language	Software Testing	BDD	Open Source

8	Test Studio	ASP. Net	ASP. Net	All	HTML, AJAX, Silverlight,ASP.NET MVC, JavaScript and WPF	Not Open Source
9	Silk Test	VB.Net	VB. Net	Software /Mobile Testing	Java,DOM,IE,Firefox,Chrome,Edge ,SAP	Not Open Source
10	Robotium	Java	Java	Software /Mobile Testing	Android Application	Open Source

VII. Analysis of software testing tools

Many tools have been analyzed in the field of software testing tool and have been categorized in many different categorizes. In [11] this segregation is done on the basis of what type of software the tool is capable of test. To conduct the study, 135 different testing tools were selected at random from internet. There study exhibited in figure 1 concluded that the use of web application testing tools is much more as compared to other software testing tools which were selected. There are number of reasons that why more focus is on web based application, but the most significant reason out of all is that maximum users are using web based applications nowadays as it is accessible from any platform regardless of any operating system.

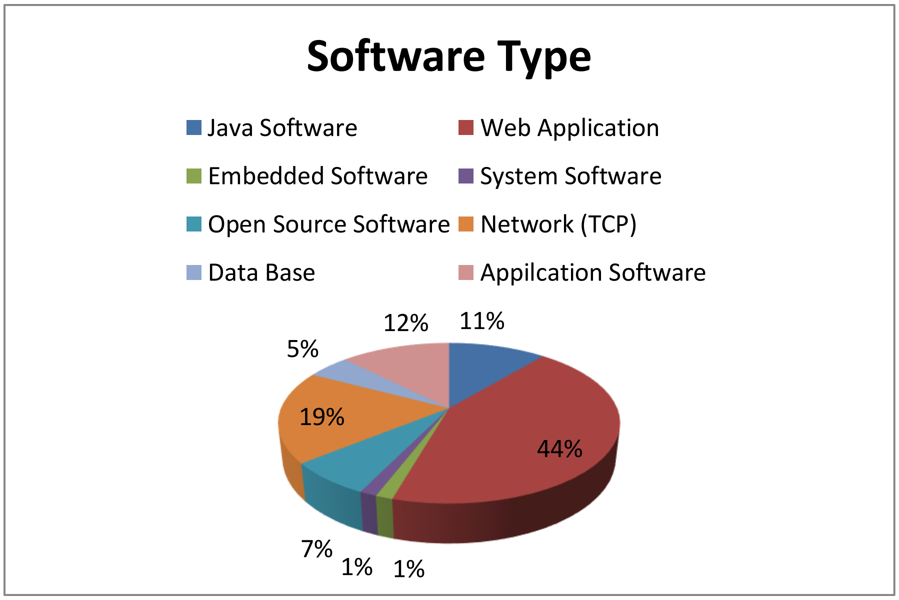

Fig. 1) Testing tools over the software type. [11]

To carry the research forward the main focus would be on the majorly used software type i.e. Web Application. A set of software testing techniques have been selected to experiment with the above mention software type. These set of testing technique consists of open source, security, acceptance, performance, unit, functional, regression, load and stress. The study exhibited in figure 2 suggested that most of the testing tools which were selected were developed to majorly focus on functional testing. Open source, unit and security testing were least on the focus.

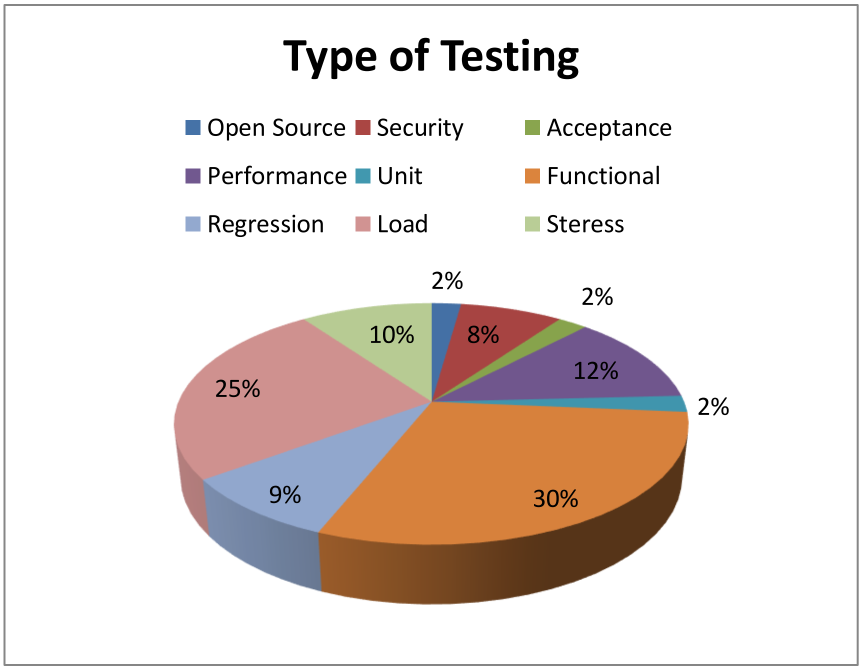

Fig. 2) Testing tools for web application over the type of testing. [11]

VII. Conclusion and future Scope

With the above conducted research we can conclude that firstly after going through this paper one would be able to make a better judgment of which tools should be selected to test software to obtain an optimized result. Second there would be a better understanding about some of the automation testing tools available.

The analysis conducted also suggests that out of all of the selected software type, web application is widely used. There could be many reasons to it but the most likely is discussed above.

Since testing is a crucial part of software engineering therefore we cannot stop at what we have achieved and there would always be scope for future development. One of the example would be, where we can see advance in software testing is Augmented Reality (AR). Since AR is a trending technology therefore it likely to catch more bugs in comparison to any other technology.

VIII. References

1) Yang, Qian, J. Jenny Li, and David M. Weiss. "A survey of coverage-based testing tools." *The Computer Journal* 52.5 (2009): 589-597.
2) Grindal, Mats, Jeff Offutt, and Jonas Mellin. "On the testing maturity of software producing organizations." *Testing: Academic and Industrial Conference-Practice And Research Techniques, 2006. TAIC PART 2006. Proceedings*. IEEE, 2006.
3) Geras, Adam M., M. R. Smith, and J. Miller. "A survey of software testing practices in Alberta." *Canadian Journal of Electrical and Computer Engineering* 29.3 (2004): 183-191.
4) Runeson, Per. "A survey of unit testing practices." *IEEE software* 23.4 (2006): 22-29.
5) Offutt, Jeff. "Quality attributes of web software applications." *IEEE software* 2 (2002): 25-32.
6) DeMilli, R. A., and A. Jefferson Offutt. "Constraint-based automatic test data generation." *IEEE Transactions on Software Engineering* 17.9 (1991): 900-910.
7) Díaz, Eugenia, Javier Tuya, and Raquel Blanco. "Automated software testing using a metaheuristic technique based on tabu search." Automated Software Engineering, 2003. Proceedings. 18th IEEE International Conference on. IEEE, 2003.
8) Liang, Hongliang, et al. "Understanding and detecting performance and security bugs in IOT OSes." Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2016 17th IEEE/ACIS International Conference on. IEEE, 2016.
9) Bhatt, D. (2017). A Survey of Effective and Efficient Software Testing Technique and Analysis.
10) Nam, Jaechang, et al. "Designing bug detection rules for fewer false alarms." Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings. ACM, 2018.
11) Mustafa, Khaled M., Rafa E. Al-Qutaish, and Mohammad I. Muhairat. "Classification of software testing tools based on the software testing methods." Computer and Electrical Engineering, 2009. ICCEE'09. Second International Conference on. Vol. 1. IEEE, 2009.