Comparative Analysis of Testing Tools

Kritika Rana¹, Mir Mohammad Yousuf²

¹Research Scholar, Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India.
²Assistant Professor, Department of Computer Science and Engineering, Lovely Professional University, Phagwara, India.

Abstract—Amongst Software Engineering’s various modules, software testing has all been one of the essential components. Thus, there are various kinds of software testing tools both paid and open source. Software testing is a method of evaluating whether or not the real outputs meet the intended outputs, and ensuring that the software system is free of all sorts of defects. Testing of software also helps to identify bugs in the software, discrepancies or missing specifications in relation to the actual needs. It can be done manually or by using automated testing tools. In this paper, we are going to perform the comparative analysis of various Software Testing tools. Software testing is done by some strategies or methods, some of the major methods are Black Box Testing and White Box Testing.

Keywords: Software engineering, Blackbox testing, Whitebox testing.

I. INTRODUCTION

Software testing is a method where the software is run with an intention to find flaws in the software. Specifically, we could say that it is a process to verify and validate that the software satisfies the market and technological specifications. There are various testing tools available in market both licensed and open source testing tools. Testing tools assure software works well even under extreme pressure and peak traffic conditions. The comparative analysis done in this paper will help you to find out which tool will work best for you. A good quality software will last longer and will perform efficiently even under extreme pressure. The total security of the software is improved with testing, although it’s not a smooth task to test software. Testing is a crucial step during the process of development of a software as every element of the concerned software needs to be evaluated so as to guarantee precision and validity and to make sure it is ready to be deployed. Many explanations exist that illustrate precisely why testing of a software is critical and the essential factors to consider when testing any kind of software. Past work in testing of software has focused mostly on more methodological problems in software testing. But focusing on methodological issues alone is not enough if somebody wants to do software engineering research that is relevant for industry. We should then give more importance to the study of our trade tools. In testing of software, it is essential to figure out the faults and bugs made throughout the phases of development. The product quality needs to be maintained. Delivering Quality products to the customers helps in gaining trust. It is important as it guarantees that the client considers the company trustworthy, and that their trust is preserved in the transaction. Testing is necessary for delivering a great-quality product that requires low maintenance cost and results into much more precise, compatible and dependable performance. To have better and smooth performance of the software from the initial deployment itself, testing is essential. Fixing bugs in future or in the upcoming development levels can turn out to be costly, and that is why testing is required to guarantee that the program does not lead to any sort of errors.
Fig. 1 [4] Software Testing Lifecycle

Some major methods used in various Software Testing types during various Software Testing levels are:

<table>
<thead>
<tr>
<th>Table 1. Software testing methods</th>
</tr>
</thead>
</table>
| **Black Box Testing** | The tester is unaware of the internal composition / architecture / implementation of the element being checked in this type of testing process. Black box testing can be either functional or non-functional. This approach tries to find: 
  • Features which are wrong or incomplete. 
  • Problems in the GUI. 
  • Data structure defects. 
  • Mistakes in conduct or execution. 
  • Initialization and termination errors. |
| **White Box Testing** | The tester is aware of the internal composition / architecture / implementation of the element being checked in this type of testing process. Testing is more thorough with this method. |
| **Gray Box Testing** | This approach is a mixture of both Black Box Testing as well as White Box Testing. |
| **Agile Testing** | This method is not sequential (means it is executed only after coding phase) but is continues. |
| **Ad Hoc Testing** | In this method, the tests are conducted informally and randomly without formal procedure. |

**Main strategies of Software Testing:** Application is manually checked in manual testing. Neither any sort of script nor any type of tool is used for testing purpose. It requires skilled labour, long time and will costs high. Manual testing can be repetitive and boring. Whereas, in automated testing involves tools to execute test cases. This testing way is time, cost and manpower saving. Automated testing is recommended only for stable systems and mostly used for Regression Testing. [5] James Bach (creator of Rapid Software Testing) said a wonderful quote: “Tools don’t test. Only people test. Tools only perform actions that help people test”. Out of the number of Software Testing Tools available in market, selection of tools is based in the project requirements.
The objectives of this paper are:

a) To compare the seven automated testing tools and seven proprietary software testing tools so that the tester can use the tools according to their needs.

b) To get the contradictory analysis of tools on the basis of various factors like: Application support, Scripting language, Programming skills, Interface, License type, Cost, Benefits, Drawbacks.

c) To identify the factor of dimensions on tools.

II. LITERATURE REVIEW

Rina and Sanjay Tyagi [6] [2013] analysed variety of tools used for performance testing such as WAPT, Loadster and NeoLoad in terms of their different performance parameters. The behaviour of various tools used for testing purpose for benchmarking is understood by evaluation. Under these performance testing methods, the same web site has been checked for consistency, variations obtained in outcomes of different parameters of performance such as latency, response time, number of hit pages, error rate, memory, CPU usage and so on. For a number of virtual users, the same website was put under load test and results were analysed.

Muhammad Abid Jamil, Muhammad Arif et. al. [7] [2016] address both the current and improved research methods for the purposes of better-quality control. They have explained that the use of simulation tools can be helpful for testers to create an environment in which the product is meant to run, some exception testing and methods for handling exception can be determined.

Dr. M. Kannan and K. Lokeshwari [8] [2017] evaluate the automated software testing tools QTP, Selenium and Load Runner to find their functionality, efficiency, and readability. This analysis helps the both the developer of the software and its tester to choose the appropriate method to suit their needs. This study analysed three tools: Selenium, QTP, LoadRunner. Authors have concluded that LoadRunner and Selenium are effective tools for automation testing. The best among these three tools is Selenium.

Rabiya Abbas et. al [9] [2017] compared 4 different automated software testing tools, Apache JMeter, HP LoadRunner, Microsoft Visual Studio (TFS) and Siege, based on certain requirements such as the generation of test scripts, plug-in support, tests reports, support for applications & costs. The emphasis is on evaluating and analyzing these load testing tools and deciding which testing tool is better or most efficient. The authors presented a systematic and detailed analysis using diverse tools used for testing. Based on this review, authors have stated that on the grounds of expenditure, time, and complexity of the software application, anybody can choose the tools for testing.
Dr. S. M. Afroz et al. [10] [2011] addressed the issues surrounding the web application, then analyzed both types of tools i.e. static and dynamic and found that DART shows good results. Study has presented report on analysis of static as well as dynamic studies, as well as contrasting the DART and Apollo Software Web Tools dynamic test generation. This study has described the Apollo is successful in comparison with the current instruments.

Mohammad Imran et al. [11] [2016] have compared the features supported by these testing tools which decide their usability and efficacy. Authors have concluded that LoadRunner will be best suitable for application that require minimum safety and when safety is required QTP is a good choice. The objective of this article is the study and comparison of automated software testing tools such as QuickTest Professional and LoadRunner to determine their usability and efficiency.

Rifa Nizam Khan et al. [12] [2015] conducted a study of automated tools available in market such as IBM Rational Functional Tester (RFT), LoadRunner, Silk Test and HP Quick Test Professional (QTP), to measure their usefulness and effectiveness. They concluded that QTP is good among the four tools.

Harpreet Kaur, Dr. Gagan Gupta [13] [2013] aims at analysing and comparing Selenium, QTP, Testcomplete to assess their usability and efficacy. The industry offers a wide variety of software testing tools. Authors have concluded that QTP is best software testing tool among the three tools.

Inderjeet Singh and Bindia Tarika [14] [2014] have evaluated the three open source testing tools and have concluded that Selenium’s rating is ideal, whereas Watir stands at second & Sikuli stands at last rank but Sikuli has fast execution speed than both the tools. The research addressed the statistical study of the various tools in terms of their capturing capability, Data Driven Testing, Quality, Supported Languages, Testing and Code Reusability which inferences the tool's feasibility under these parameters.

Manjit Kaur, Raj Kumari [15] [2011] evaluated the functionalities that are supported by QuickTest Professional & also by the Automated QA TestComplete functional testing tools which help to reduce script maintenance resources and increase script reuse performance. Comparative analysis of automated tools like Mercury QuickTest Professional & Automated QA TestComplete has been conducted based on factors like efforts involved in making test scripts, the capacity to repeat tests, record performance, pace and expense.

II. COMPARATIVE ANALYSIS

Seven open source software testing tools have been taken as proprietary software testing tools on the basis of scripting languages, license type, benefits and drawbacks. Below is the comparison table that can help testers to select tools according to their needs. The table does a comparative analysis that can be helpful for the researchers, academicians as well as the testers.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Features</th>
<th>Apache J meter</th>
<th>MS Visual Studio</th>
<th>Watir</th>
<th>Selenium</th>
<th>Test Complete</th>
<th>Siege</th>
<th>Sikuli</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Scripting language</td>
<td>JavaScript, BeanShell</td>
<td>PowerShell, Perl</td>
<td>Port to .Net in WatiN (support for all 20+ .Net languages like C# and VB.NET), Port to Java in Watij.</td>
<td>JAVA, C#, Peri, Python, JavaScript, Ruby, PHP</td>
<td>VBScript, JSScript, DelphiScript, C++Script, C# Script</td>
<td>Skrit</td>
<td>Python language level 2.7, running RobotFramework text-scripts is supported, Ruby language</td>
</tr>
<tr>
<td>S. No.</td>
<td>Features</td>
<td>Quicktest professional</td>
<td>HP Load runner</td>
<td>Silk test</td>
<td>Rational func. tester</td>
<td>Test complete</td>
<td>Ranorex</td>
<td>SOAtest</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------</td>
<td>------------------------</td>
<td>---------------------------------</td>
<td>-------------------------</td>
<td>-----------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>1</td>
<td>Scripting language</td>
<td>VB Script</td>
<td>Citrix, ANSI C, .Net and Java</td>
<td>Embedded VB, .NET scripting, Eclipse plug-in for Java developers</td>
<td>Java or Visual Basic.net</td>
<td>JavaScript, Python, VBScript, DelphiScript, C#Script and C++Script</td>
<td>C#, VB .NET</td>
<td>Java, JavaScript, Jython, Groovy, Scripting Languages</td>
</tr>
<tr>
<td>2</td>
<td>License type</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
<td>Proprietary</td>
</tr>
</tbody>
</table>
### IV. CONCLUSION

The testing of software is an important component of the application development cycle. It isn’t a single operation after application creation, but is part of every stage of the lifecycle. During the specification of the specifications a successful test strategy must start with consideration. Technical specifics will be fleshed out by device designs at high and low rates, and technical will be carried out after completion of the application developers and separate test classes. As with other software lifecycle tasks, testing has unique challenges of their own. The value of successful, well-planned testing activities will only increase as software systems become increasingly complex. After comparing all these open source and proprietary software testing tools, I would like to conclude that Open source software testing tools have strengths like minimum expenditure needs to be done to use them, they can be reused, they produce source code which could be relied upon, also they are safe and smooth. Proprietary software testing tools have strengths and weaknesses of their own. Depending on the plan, time, and complexity of the software application, everyone can choose the testing tools.

### V. REFERENCES:


