

# A Performance Based Comparison of Two automated Software Testing Tools: Review

Ashish Kumar Sharma<sup>1</sup> Surendra Kumar<sup>2</sup>

<sup>1</sup>M. Tech Student <sup>2</sup>Assistant Professor

<sup>1,2</sup>Dr. K. N. Modi University, Newai, India

**Abstract**— Software testing is a very important part of any software development life cycle. Testing is an important part of delivering a quality software product. Software testing are basically two types manual testing and automated testing. Manual testing requires many effort and hard work, so there was always a need in software testing to decrease the testing time. For reducing the testing time and find the more effective result automated testing is used. Now a day the use of the computer is increased. To improve the quality of the software product is very essential.

**Keywords:** SDF, Software, Manual Testing, Automated Testing

## I. INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate process to ensure the correctness of the product. The quality product always enhances the customer confidence in using the product thereby increases the business economics. In other words, a good quality product means zero defects, which is derived from a better quality process in testing. Testing is the process of executing a program. William Howden, a professor from University of California at San Diego, wrote that “testing is the unavoidable part of any responsible effort to develop a software system”. The software testing are basically two types (1) Black Box Testing (2) White Box Testing. Black box Testing is also known as behavioral Testing, is a software testing method in which the internal coding of the item being tested is not known by the tester. These tests can be functional or nonfunctional.

This method is named so because the software program is in the eyes of the tester is like a black box. So tester is not able to see inside.

### A. White Box Testing

White box (also called, clear box) testing is completely opposite of the black box testing. In this testing, tester requires having some level of programming skills. The internal structure of the system, as well as details of code, should be clearly understood by the tester. As most of the software defects or bugs are caught and also resolved during white box testing (during the unit test), it's a very effective way of testing. Here, the tester does not need to implement the program but should have programming knowledge to find the bug or anomaly within the code. A good example for white box testing is “unit testing”, where the code developed for a particular module, that needs to be tested prior to integrating it with rest of the modules. Thus, unit testing is very efficient in minimizing overall bugs in a system. Test-driven development (TDD) utilizes the white box testing technique.

### B. Manual Testing

In manual testing, the basic level of testing, the tests are performed following the test cases and directly interacting with the application. In this testing, the tester prepares the test cases. Test cases are written in simple English language, which illustrates the features or functionalities to be tested and the expected result. As the tester needs to perform all these activities manually, the whole process of testing can become too lengthy. However, for some particular complex system or application manual testing is preferable and effective since some critical and rare bugs are discovered during manual testing. The tester for manual testing plays a similar role to an end user of that software application and explores the different parts of the application by testing and making sure about the correctness of application. Some of the drawbacks for manual testing are

- Time inefficient and labor intensive.
- Very flat learning curve.
- Lacking the advantage of reusability.
- Not an iterative process, or multiple iterations do not necessarily provide better accuracy.
- Manual tests provide limited visibility, as the tester does not have much knowledge of how the code works.
- Tests have to be repeated by the software developer, tester and finally by the business analyst (to verify that the User Interface is working properly).
- No validation process to verify that the test is actually performed correctly; test cases are manually written whether the test passes or fails.

### C. Automated Testing

With the boom of test automation within the industries, testing has become very efficient. Test automation eliminates the burden of user initiation and difficulty of performing various types of testing such as regression and load/performance testing. With the advancement of automated testing, complex testing tasks became much easier than before, as it allows performing the test with numerous sets of data and multiple times without intervention of human.

Automation testing requires some initial investment for the software and compatible hardware resources but could potentially be more economical since it reduces the human efforts in testing. The process of automated testing can be conducted in different stages. But in general, can be divided into the four basic ways:

- 1) Preparing the test plan or creating the test cases
- 2) Preliminary selection of testing tool
- 3) Writing/generating the test script
- 4) Performing the test using automated test scripts

### D. Objective

The goal of this research paper is to introduce the various features and presentations of software testing tools (Ranorax

and UFT/QTP) as well as assess and compare these tools to determine their usefulness. This study also allows drawing the basic comparison among automated and manual testing to represent the significance of automated testing in software industries. To accomplish the goal of this research, the following steps are to be performed.

- Identifying a set of tools to be evaluated.
- Selecting the target application to be tested by all these tools.
- Testing the target application using the selected automated testing tools and gather resulting data.
- Developing a set of comparisons to be used to assess the tools.
- Performing an analysis of each tool and comparing each other based on an ideal feature set and depending on the result of test execution.
- Outlining inferences and making recommendations based on the outcome of research.

## II. LITERATURE REVIEW

Research conducted by Nisha Gogna [12], presented the basic features of the automation tool: WATIR and Selenium. Gogna mentioned that Frames and pop-ups are accessible using WATIR. However, Selenium requires the user to have advanced language skills in order to test for frames and pop-ups. This is indeed a good paper to learn about Selenium and WATIR, but lacks in the area of comparison between these two tools.

Harpreet Kaur and Gagan Gupta [13] showed a comparative study among Selenium, Test Complete, and QTP tools. That study includes different aspects but does not provide the automation features, such as record and playback, cross platform or browsers support features.

Rigzin Angmo and Monika Sharma [14] conducted a performance evaluation of Selenium suite with WATIR Web driver. Although that research covers most of the comparison criteria, the one thing that's absent in the research is comparisons with any commercial testing tool. There would be more cases to compare while bringing both open source and commercial tools into consideration.

T. J. Naidu and N. A. Basri conducted a research, [15] where both the tools are open source (Selenium and SAHI). Finally, they mentioned, Selenium is aimed to be best for programmers while SAHI is aimed at non-programmer testers. In the research conducted by Abha Jain and Manish Jain, the different features of QTP and Ranorex have been analyzed and compared. It's a good research to follow but it lacked comparison with any open source tools.[16] Another research was conducted by Meenu and Yogesh Kumar, where the comparison was made among UFT/QTP, Selenium, Test Complete and SoapUI. All these tools were compared to a number of features and based on the comparison it declared SoapUI is the best tool [17]. However, SoapUI is a commercial tool, does not provide support for multiple languages, operating systems, and windows applications. The web application to be tested for this research is the students' E-services website for St Cloud State University. The main focus of the testing will be the course registration process.

## III. METHODOLOGY

The use of automated software tools has increased over the years. The continued success of software or website is highly dependent on the reliability of the automated software tool. Several tools are available to a developer when performing automated testing. This chapter discusses an explorative case study on a Quick Tech Professional and Ranorex automated software tools

### A. Selected Tool

The testing tools chosen in the comparison for stand-alone based testing were QTP, and Ranorex. Initially, many other test tools may be selected but were choosing them because of many reasons. For example, Quick Test Professional (QTP) known to be the industries best solution for functional testing and regression test automation.

QTP uses keyword driven approach providing the test automation engineer full access to underlying test and object properties via an integrated scripting and debugging interface. Ranorex is the fastest growing automation tool which has imbibed the features of the best automation tools available in market and providing an edge over them.

Being a banking project, the client is more interested in speedy and reliable testing. Usually, banking clients avoid changes as reliability is a prime concern for them. If Ranorex is to be implemented in a banking project, following are the roadblocks to be taken care of:-

#### 1) POC

As per expectation the client to be reluctant to switch to a new tool or for the first time automation projects, a tool which is not yet proved in market, so it is required a very elaborate POC report and critical cost saving data.

#### 2) Low License Cost

The licensing cost for Ranorex is considerably lower than QTP. This is a primary sell point if the client is budget centric as a lot of cost would be saved.

### B. Evolution Methodology

There are many reasons, to compare testing tools like QTP & Ranorex. Tables are important because it shows us a way to compare different tools to efficiently select appropriate tools to use, based on what testing needs are at the given time. The criteria for comparison in each tool studied are as follows: Basic features

- debugging facilitate,
- automated progress,
- support for the testing process,
- support for the usability,
- and required time & Software cost.

## IV. RESULTS

### A. Result of Testing Tool

The results were extensive and outlined in this section. Both Ranorex and QTP tools have been tested, evaluated and compared to each other. Following are the three main criteria for selection of tool for automation of the manual test cases:

**B. Project Testing Budget**

Being a banking project, this may expect a good share of budget on testing which would allow scope for POC's and tool selection.

**C. Project Criticality**

Being a critical domain, testing needs to be trustworthy. This calls for a tool which is well proven for similar projects.

**D. Time Allotted for Testing**

Each day delay in rolling out of a banking application means a huge loss for the client. This adds to the requirement of a testing tool which can be scripted faster and has lower execution time per cycle.

All these factors will affect the selection of tool for automation. Going ahead, we will do Proof of Concept for two automation testing tools i.e. Ranorex and QTP taking these points into consideration. The first table, features of testing tools, has results that are detailed below.

| S. No. | Feature   | QTP  | Ranorex  |
|--------|---|--|--|
| 1      | Language Support                                | VB Script  | Java, C#, VB.net, Python   |
| 2      | Windows (Non browser) based Application support | Yes, It also supports add-ons, but user needs to purchase license for them.                    | Yes it also support windows based Application  |
| 3      | Browser support                                 | Google Chrome Internet Explorer , Firefox ( ver. 21)   | Opera, Firefox, Netscape, IE, Chrome   |
| 4      | Operating system support                        | Only Windows   | Windows 7, Windows Vista, Windows Server   |
| 5      | Environments Supported                          | ActiveX, Visual Basic, Web (Ships with basic license-Supports other environments with add ins) | Win32, WPF, WinForms and QT Desktop applications, HTML, AJAX, Flash, Flex and Silverlight web applications |
| 6      | Mobile (Phones & Tablets) support               | Different commercial product i.e. HP UFT Mobile (formerly known as MobileCloud for QTP)        | Android , iPhone & iPad , Blackberry , Headless WebKit   |

Table 4.1: Result of Basic Features Metric

| S. No | Features               | QTP  | Ranorex  |
|-------|------------------------|--|--|
| 1     | Frame work             | Easily integrated with HP Quality Center or HP ALM (separate commercial products)  | Easily integrated  |
| 2     | Continuous Integration | Possible through Quality Center / ALM or Jenkins   | Ranorex can easily be integrated into continuous integration systems as Ranorex Studio and others.   |
| 3     | Object Recognition     | Inbuilt Object Repository (storing Element Id, multiple attributes) along with weightage that gives flexibility on deviation acceptance in control recognition | It has a highly defined object repository which properly stores and displays the objects in the hierarchy of their presence on the screen. |
| 4     | Image based test       | Easily possible  | Very developed image based object recognition; it can support any graphic based environment.   |

Table 4.2: Result of Tools Usability Metric

The above Metric shows the Tools Usability Metric. The most used QA tool in the market is QC which again is a product of HP, the vendor for QTP. Thus, QTP provides exceptional integration with QC. Ranorex does have integration with QC and other QA tools but to limited extent. The next metric shows Debugging help metric. Because of lesser world presence, Ranorex does not has teams working round the clock which affects their turnaround time for people working at shifts other than the Austrian time. On the other hand, QTP has a world presence and teams are available round the clock for help.

Next metric shows testing process metric. It is known to be the industries best solution for functional testing and regression test automation. It uses keyword driven approach proving the test automation engineer full access to underlying test and object properties via an integrated scripting and debugging interface.

| S. No | Features                              | QTP            | Ranorex |
|-------|---------------------------------------|----------------|---------|
| 1     | Compare test result with database     | Yes            | Yes     |
| 2     | Document test cases                   | Yes            | Yes     |
| 3     | Ability to perform regression testing | Easily Perform | Yes     |

Table 4.5: Result of Testing Process Metric

The next metric considering various factors, this is how the timeline and cost of project would be affected by using QTP and Ranorex. Final Outcome is the time required for training resources on QTP is much lower than training them in Ranorex.

The initial license cost for QTP is a lot higher than Ranorex which makes the project initiate at a higher cost as compared to Ranorex. Knowing that QTP is the world leader in automation testing, the price difference could be justified.

But with similar features being provided by Ranorex, the wide gap in cost does not justify.

| S. No. | Features                         | QTP  | Ranorex  |
|--------|----------------------------------|--|--|
| 1      | Software cost                    | Licensed and very Expensive, i.e. 8000 USD   | Licensed but not very much Expensive, i.e. 1855 USD  |
| 2      | Script creation time             | More time  | Lower time due to ease of scripting, better interface and more options to ease the work.         |
| 3      | Execution time & Running Cost    | Same for both the tools.   | Same for both the tools.   |
| 4      | Training time & cost             | Lower Training time and trainer is to be paid separately as the vendor does not provide free training. | Higher training time and excellent training services provided by the company along with license. |
| 5      | Experience of coding engineering | Not Much required  | Not Much required  |
| 6      | Setup cost                       | QTP is a lot higher  | Lower  |

Table 4.6: Result of Time and Cost Metric

| S. No. | Criteria                   | QTP | Ranorex |
|--------|----------------------------|-----|---------|
| 1      | Cost                       | 3   | 8       |
| 2      | Environments               | 6   | 9       |
| 3      | Browser Supported          | 3   | 10      |
| 4      | Online Support             | 10  | 6       |
| 5      | Coding Languages Supported | 4   | 9       |
| 6      | Ease Of Learning           | 8   | 8       |
| 7      | Strongly Typed             | 3   | 9       |
| 8      | Training cost              | 9   | 5       |
| 9      | Evolution With Age         | 10  | 3       |
| 10     | Integration With QA Tools  | 8   | 6       |

Table 4.7: Tool Comparison

## V. CONCLUSIONS

- 1) After simulation of the topic, one point is indubitable that automation testing is much more suitable than manual testing.
- 2) Coming to the tool selection, Ranorex could provide serious cost benefit. Though, it loses its edge due to a very strong market hold of QTP, so clients do not prefer to experiment on other technologies. Specific to banking projects, due to its criticality, not enough consideration is given to new POC's and tools.
- 3) Though, initially switching is in both tools is costly and time consuming, the switch to Ranorex has a lot of long term benefits. If a project is already implementing and has no plan to add new modules, which is a least possible scenario, then the switch to Ranorex would be unfruitful.
- 4) With the help of QTP it is easy to generate most comprehensive reports due to the availability of an efficient online help and Stores test results in an open XML format, enabling you to easily customize the reports according to your own requirements
- 5) The test report generated is complimented with graphs for faster and better comparison of defects in every run.

## REFERENCES

[1] David Crowther, Peter Clarke, Examining Software Testing Tools, and Dr. Dobb's Journal: Software Tools for the Professional Programmer, ISSN# 1044789X Academic Search Premier, June 2005, Vol. 30, and Issue 6.

[2] Pak-Lok Poon, Sau-Fun Tang, GH Tse, TY Chen, Choc"late A Framework for Specification-based Testing, ACM, Vol 53, No. 4, April 2010.

[3] Roger Pressman, Software Engineering, A Practitioner's Approach, Sixth Edition, 2005.

[4] Sukhjinder Singh1, Er.Sukhpreet Singh2, Dr.Madhuchanda Rakshit , "A Review of Various Software Testing Techniques", IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 4, Aug-Sept, 2013 ISSN: 2320 – 8791.

[5] Prof. (Dr.) V. N. Maurya, Er. Rajender Kumar "Analytical Study on Manual vs. Automated Testing Using with Simplistic Cost Model", International Journal of Electronics and Electrical Engineering ISSN: 2277-7040 Volume 2 Issue 1 (January 2012)

[6] John Satzinger, Robert Jackson, Stephen Burd, Object-Oriented Analysis & Design with the Unified Process, Thomson Course Technology, 2005.

[7] A Askarunisa, N Ramraj, Shanmuga Priya, Selecting Effective Coverage Testing Tool Based on Metrics, The Icfai University Journal of Computer Sciences, Vol III, No. 3, 2009.

[8] Linda Hayes, The Truth about Automated Test Tools, Datamation, Vol 43, Issue 4, April 1997, EBSCO host database, Academic Search Premier.

[9] Alex Cerv antes, "Exploring the Use of a Test Automation Framework", IEEEAC p ap er #1477, version 2, up dated January 9, 2009.

[10] Ieshin, M. Gerenko, and V. Dmitriev, "Test Automation-Flexible Way", IEEE, 978-1-4244-5665-9, 2009.

[11] A Askarunisa, N Ramraj, Shanmuga Priya, Selecting Effective Coverage Testing Tool Based on Metrics, The Icfai University Journal of Computer Sciences, Vol III, No. 3, 2009.

[12] T. K. Abdel-Hamid, The Economics of Software Quality Assurance: A Simulation-Based Case Study, (Management Information Systems Research Center, University of Minnesota), URL: <http://www.jstor.org/pss/249206>, 1988.

[13] H. Liu and H. B. Kuan Tan, "Covering code behaviour on input validation in functional testing," Information

- and Software Technology, vol. 51, no. 2, pp. 546–553, Feb. 2009.
- [14] P. Mitra, S. Chatterjee, and N. Ali, “Graphical analysis of MC/DC using automated software testing,” in Electronics Computer Technology (ICECT), 2011 3rd International Conference on, 2011, vol. 3, pp. 145–149.
- [15] Shivkumar Hasmukhrai Trivedi Software Testing Techniques, Volume 2, Issue 10, October 2012 ISSN: 2277 128X
- [16] Srinivas Nidhra1 and Jagruthi Dondeti ,BLACK BOX AND WHITE BOX TESTING TECHNIQUES –A LITERATURE REVIEW, International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012
- [17] Jovanović, Irena, Software Testing Methods and Techniques, , May 26, 2008, <http://www.internetjournals.net/journals/tir/2009/January/Paper%2006.pdf>
- [18] <http://www.internetjournals.net/journals/tir/2009/January/Paper%2006.pdf>
- [19] T. Murnane, K. Reed, and R. Hall, “On the Learnability of Two Representations of Equivalence Partitioning and Boundary Value Analysis,” in Software Engineering Conference, 2007. ASWEC 2007. 18th Australian, 2007, pp. 274–283
- [20] T. Murnane and K. Reed, “On the effectiveness of mutation analysis as a black box testing technique,” in Software Engineering Conference, 2001. Proceedings. 2001 Australian, 2001, pp. 12–20.
- [21] Beizer, Software Testing Techniques. Dreamtech Press, 2002.
- [22] S. Noikajana and T. Suwannasart, “Web Service Test Case Generation Based on Decision Table (Short Paper),” in Quality Software, 2008. QSIC '08. The Eighth International Conference on, 2008, pp. 321–326.
- [23] M. Sharma and B. S. Chandra, “Automatic Generation of Test Suites from Decision Table – Theory and Implementation,” in Software Engineering Advances (ICSEA), 2010 Fifth International Conference on, 2010, pp. 459–464.
- [24] Software testing glossary available at [http://www.aptest.com/glossary.html#performance testing](http://www.aptest.com/glossary.html#performance_testing).
- [25] White Box Testing available at [http://www.testingbrain.com/WHITEBOX/WHITE\\_BOX\\_Testing.html](http://www.testingbrain.com/WHITEBOX/WHITE_BOX_Testing.html).