www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 may, 2014 Page No. 5916-5923

Studying and Comparing Automated Testing Tools; Ranorex and TestComplete

Neha Dubey, Mrs. Savita Shiwani
Department of Computer Science and Information Technology
SURESH GYAN VIHAR UNIVERSITY, JAIPUR

Abstract

Testing automation tools enables developers and testers to effortlessly computerize the complete practice of difficult in software progress. The intention of this research paper is to carry out a comparing and studying the concepts, builds and features of automated tools such as the Ranorex and the Automated QA TestComplete based on criteria such as the hard work involved with generating test scripts, capacity to playback the scripts, end result reports, and expenditure. The elementary objective is to investigate the features and concepts supported by these two functional testing tools in order to access unconventionally what pros and cons of the tools and what could be the guidelines for its additional expansion.

Keywords: Ranorex, Test Complete, Testing tools.

1. INTRODUCTION

The aim of software testing practice is to recognize every defect obtainable in a software product. It is the process of exercising and evaluating a system or system components by means of manual automatic to validate that it satisfies particular necessities or to categorize differences involving predictable and definite consequences. There are two ways of testing that are manual or automation.

Manual testing carried out by the testers. Testers test the software manually for the defects. It requires a tester to play the role of an end user, and use most of all features of the application to ensure its correct behavior. They follow a written test plan that leads them through a set of important test cases .The problems with manual testing are, it is very time consuming process, not reusable, has no scripting facility, great effort required, and some errors remain uncovered .

Automation testing covers all the problems of manual testing .Automation testing automates the steps of manual testing using automation tools such as Ranorex and TestComplete (TC) .It increases the test execution speed, more reliable, repeatable, programmable, comprehensive, and reusable.

Testing automation tools enables developers and testers to effortlessly computerize the complete practice of difficult in software progress. The intention of this research is to carry out a comparing and studying the concepts, builds and features of automated tools such as the Ranorex and the Automated QA TestComplete based on criteria such as the hard work involved with generating test scripts, capacity to playback the scripts, end result reports, and expenditure. The elementary objective is to investigate the features supported by these two functional testing tools that aid in minimizing the resources in script maintenance and increasing efficiency for script reuse.

Table 1: Criteria for comparison

Parameters to be taken	Meaning
Recording Efficiency	For manipulation the application under test.
Capability of generate scripts	Generating the corresponding scripts.
Data Driven Testing	Reduces efforts.
Test Result Reports	Effective analysis of test scripts.
Reusability	Goal of the test automation.
Playback of the scripts	Replays the user actions.
Easy to learn	GUI
Cost	Cheaper

2. Brief Literature survey

In this segment, we planned the papers that reviewed during this project .In up to date years the significance of stylish User Interfaces has increased a lot. Nowadays User Interfaces have to deal much more than before with untrained people sitting in front of their computers. So it is no wonder that not only the “automation behind the scenes” (Unit Testing for example) gained in importance, but also the automation of User Interface Tests with all its boon and bane .The paper surveys a set of tools that support the testing process in a variety of ways. Some tools simulate the final execution environment as a way of expediting test execution, others automate the development of test plans, and still others collect performance data during execution .In these tough economic times, software- development managers are pushing to get more and better testing done faster. Most recognize the automated testing tools facilitate higher quality and more productive testing, but acquiring such tools is often complicated. The paper had given the evaluation criteria for selecting the testing tools.

The paper gives a survey which tries to give an account of what type of trends exist today in software reuse and testing. The focus was to try to find out how developers use different tools today and what are lacking, especially in the field of reuse and testing. The paper classifies and distributes a set of testing tools over the types of testing (testing methods) for three types of software products (web application, application software, and network protocol).The paper told us if we’ve got a reasonably well structured system implementation, it is very easy to add in a mechanism to capture interactions with operations that system provides and to generate playback artifacts that are meaningful. Performing operation-centric capture/replay avoids many of the pitfalls of traditional GUI centric capture/replay.

The following problems are identified in the literature:

- Identify the requirements of a project.
- Lack in domain knowledge.
- Lack reusability in testing.
- Evaluation of testing tools.
- Many pitfalls in capture and replay.
- More time & effort taken by testing tools.
- Costly and time consuming tools.
- Lack in recording efficiency.
- Problems in communications, coordination and control in testing.
- Execution speed and test results reports.

3. Software Automated Testing Tools

3.1 Ranorex

Ranorex is an economical and complete tool used for programmed testing. This is an improved substitute to conventional challenging tools for the reason that it tests applications from a user’s perception, exhausting regular programming procedures and common languages such as C# and VB.net. It does not necessitate to study a scripting language, since it is written in pure .net code. We can use any one of the three languages, C#, VB.net and Iron Python. It is used by hundreds of initiative and commercial software companies everywhere. The recreation tools such as this can have comparable difficulties to the record and playback approaches, as the assessments are frequently resolutely friendly to the code, and both approaches still trust profoundly on expertise to generate the precise examinations to guarantee full consideration.

Ranorex is centered on XPath, which is a very good technique to catch certain elements in a web based application. It is a pure .net API, which is very different from other tools which sit on an API. Future plans for this tool involve creating an open and documented interface for the users to write their own plug-ins, which provides the maximum of object acknowledgement for their own applications. Following are some of the features in the tool.

- Image-based detection
- Contains Ranorex Recorder for Record-Replay
- Provides unified incorporation for 32 and 64 bit operating systems
- Constructed on the .NET Framework
- Proposals a stretchy and standard test mechanization interface
- The test automation modules can be created as simple executable builds, with a standard .NET compiler.
- The Ranorex automation library (API) is built on .NET, consequently, letting you to incorporate it into current test surroundings and to association current automation jobs through Ranorex.
- Due to smart and easy to read automation code, the use of Ranorex repository, which separates GUI identification information from automation code.
- Offers the facility to do test automation in your own environment Uses standard and modern programming techniques.
- Permits testers with less programming awareness to generate professional test modules with Ranorex Recorder.
- Ranorex Recorder brings user code actions, which permits developers to provide special validation or automation methods for their testers with less experience in programming
- Objectives to acquire everything automated.
- Supports all the technologies through the Ranorex Plug-Ins
- User interface allows for managing test cases and configurations
- Supports procedure of data variables

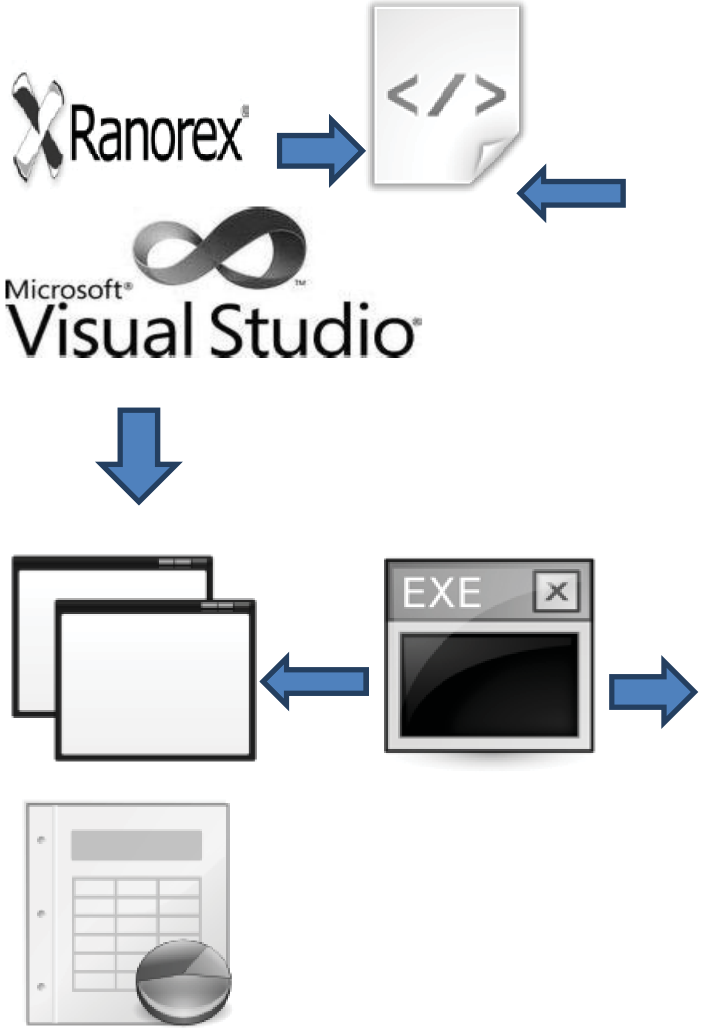


Figure:1 Ranorex Concept

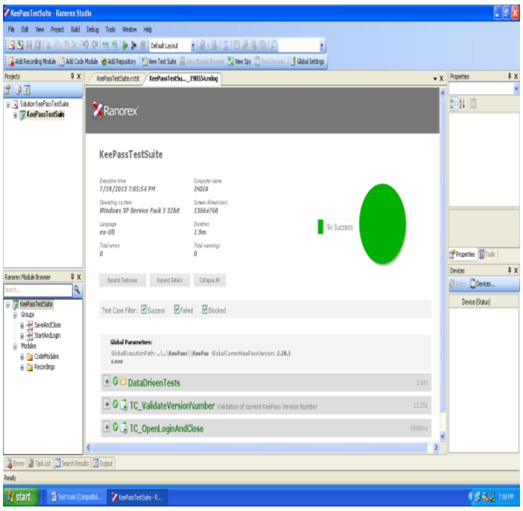


Figure 2: Ranorex Test Reports

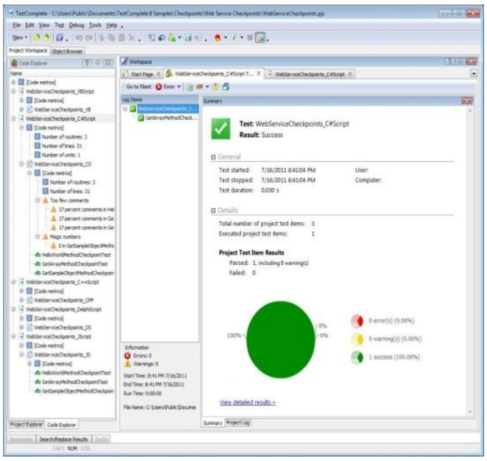
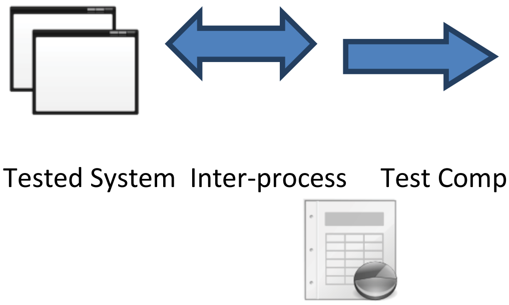


Figure 3: TestComplete Test Reports

3.2 TestComplete

TestComplete is an automatic self-testing tool industrialized through SmartBear; it makes available the testing of Windows and web applications and is one of the primary functional testing tools in the world. This is moreover confirmed by the fact that the tool has won the ATI Automation Honors award as the Best Commercial Automated Functional Testing Tool in 2010, and it is used in their projects by world's leading companies like Adobe, Corel, Falafel Software, ARUP Laboratories, QlikTech etc.

The TestComplete tool uses a keyword-driven testing framework to perform functional tests; in addition, with it it is possible to also develop tests with scripts. Its operation concept is comparatively simple. As shown in Figure 2, the tool, through inter-process communication and various built-in auxiliary tools, records the actions performed in the tested system and after that also execute them.



Test Reports

Figure4: Test Complete Concept

The following are the main features of TestComplete:

Table 2: Comparison of Testing Tool

Criteria	TestComplete	Ranorex
Test Recording	Yes	Yes
Desktop application Testing	Yes	Yes
Data driven Testing	Yes	Yes
Test Result reports	Yes	Yes
Playback	Yes	Yes
Identifying the tested object	Yes	No
Plug- ins and extensions	Yes	No

Table 3: Comparison of Testing Tool

Testing Tool	Developer	Price (EUR)	Client	Tool's Programming Language
TestComplete	SmartBear Software	~1 400	Adobe4,Corel,Falafel Software,ARUP Lab,QlikTech u.c.	VBScript,Jscript,C++ Scripts,C# Scripts,DelpghiScript

				s
Ranorex	Ranorex	~1 190	Bosch,General Electics,FujistsuSiemens,Yahoo,Real VNC u.c	C++,Python,C#,VB.NET

4. Comparison Results

To assess every comparison criteria determined, information obtained from the tool's official website and other trusted websites, specifications and help windows and

- TestComplete has an incorporated keyword-driven test editor that consists of keyword operations that communicate to automated testing actions.
- Built-in code editor that helps testers write scripts manually. It also includes a lay down of unusual plug-ins that helps.
- Proceedings the key measures required to replay the test and throw-outs all unwanted trial.
- TestComplete reads the names of the noticeable essentials and a lot of interior elements of Delphi C++Builder, .NET, WPF, Java and Visual Basic applications and allows test scripts to admittance these values for confirmation or employ in tests.
- TestComplete Unicode character set support allows testing of non ASCII applications that use Unicode character sets like Arabic, Greek, Katakana, Hebrew and others.
- Its engine is based on an open API, COM interface. It is source-language independent, and can read debugger information and use it at runtime through the TestComplete Debug Info Agent.
- Automatically captures screenshots during test recording and playback. This enables quick comparisons between expected and actual screens during test.
- Supports plugins so third-party vendors can connect TestComplete with their applications.

from practical use of the tools was used to make sure that they comply with the respective criteria.

The paper contains two comparison tables: Table 2 provides a summary on the criteria that the testing tools supports and Table 3shows the additional criteria for comparing tools.

For the purpose of rating the comparison parameters, we have used 5-pints scale i.e. 5 ,4 ,3,2,1 as Extremely Good, Average, Fairly Bad and Extremely Bad respectively.

4.1 Recording Efficiency

Sub Crietria	**Ranorex**	**TC**	**Comment**
Insert Commands	1	5	In Ranorex,we cannot insert commands while recording.
Recording Type	5	5	Both tools provide facility to record the mouse movements,screen co-ordinates,keystroks and objects and their properties.
Access to read controls	1	5	The recording toolbar of TC test always present at the application under test.This is not the case with Ranorex.
Pause	5	5	Both provide this facility.
Auto Documentation	5	5	Both tools provide this facility.
Validation	5	1	Only Ranorex

			provide the facility of validation of attributes.

For Ranorex, the value of Parameter is=1+5+1+5+5+5/6=3.6

So,Ranorex is average in this parameter.

For TC,the value of parameter is=5+5+5+5+5+1/6=4.3

TC is extremely good in recording efficiency.

4.2 Data Driven Testing

Both provide the access to different external sources and both tools can keep the scripts separated from data.

4.3 Test Result Reports

Sub Crietria	Ranorex	TC	Comment
Report Presentation	5	4	Ranorex gives executive summary of Results.TC gives results in single pane.
Info about applied chekpoints	5	5	Both provide the Info
Graphical info of previous runs	5	1	Only Ranorex gives info about previous and current runs in Pie Charts.

Ranorex is extremely good.

For TC, the value of parameter is=4+5+1/3=10/5=3.3

Tc is average.

4.4 Playback Capability

In our study, both tools play back the scripts efficiently. So we rank this parameter for both as fairly good (4pt).

4.5 Easy to Learn

Ranorex is little complex than TC due to more features,we rate the TC as extremely good (5pt) and Ranorex as fairly good(4pt).

4.6 Cost

TC is little costly than Ranorex tool. See Table 3.

5. Conclusion and Future Work

In final this research, I have learned that software testing tools are very dissimilar. It takes time and effort and having a software testing goal to know which tool is the best to use given the type of software testing needs. In this paper, we have tried to solve & add some new ideas to support testing approach by Ranorex and TestComplete. . The intention of this research is to carry out a comparing and studying the concepts, builds and features of automated tools such as the Ranorex and the Automated QA TestComplete based on criteria such as the hard work involved with generating test scripts, capacity to playback the scripts, end result reports and expenditure. The elementary objective is to investigate the features supported by these two functional testing tools that aid in minimizing the resources in script maintenance and increasing efficiency for script reuse. Ranorex is the finest tool for web based applications given the different test automation tools built into the software package. My visualization for the best tool is one that is cloud based with no install required and is easy to learn how to use. The ideal testing tool should be easy to navigate and also contain many tutorials on how to get started in with the tool. It be supposed to also have negligible bugs and economical. I wrap up that TestComplete may be exact for assured definite situation but Ranorex can be the superior choice in many more situations.

References:

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

2. J. Meek, N. Debnath, I. Lee, H. Lee. Algorithmic Design and Implementation of an Automated Testing tool, URL: http://www. computer. org. uncclc. coast. uncwil. edu/portal/web/csdl/abs/proceedings/itng/2011/4367/00/4367a054toc. htm. pp. 54-59, Eighth International Conference on Information Technology: New Generations, 2011.

3. N. Koochakzadeh, V. Garousi, A Tester-Assisted Methodology for Test Redundancy Detection, Advances in Software Engineering, Hindawi Publishing Corporation, V 2010, Article ID 932686, 2009, URL:http://www.hindawi.com/journals/ase/2010/932686/

4. Sara Sprenkle, Holly Esquivel, Barbara Hazelwood, Lori Pollock, WebVizor:A Visualization Tool for Applying Automated Oracles and Analyzing Test Results of Web Applications, IEEE Computer Society, August 2008.

5. Macario Polo, Sergio Tendero, Mario Piattini, Integrating techniques and tools for testing automation, EBSCO host database, ISSN# 09600833, Wiley Publishers, March 2007.

6. Darryl Taft, IBM Readies Rational Revamp, EBSCO host database, ISSN# 15306283, Academic Search Complete, June 2006.

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

8. IBM.com, IBM Rational Functional Tester, IBM Corporation, December 2008, URL: http://public.dhe.ibm.com/common/ssi/ecm/en/rad14072usen/RAD14072USEN.PDF.

9. Stuart Feldman, Quality Assurance: Much More than Testing, ACM Digital Library, Queue – Quality Assurance, February 2005, Vol 3, Issue 1.

10. Keith Stobie, Too Darned Big To Test, ACM Digital Library, Queue – Quality Assurance, February 2005, Vol 3, Issue 1.

11. William White, Sifting through the Software Sandbox SCM meets QA, ACM Digital Library, Queue – Quality Assurance, February 2005, Vol 3, Issue 1.

12. Antti Kervinen, Pablo Virolainen, Heurisitics for Faster Error Detection with Automated Black Box Testing, Science Direct, Electronic Notes in Theoretical Computer Science, Vol 111, January 2005, URL: http://www. sciencedirect. com/science/article/pii/S1571066104052326

13. Siegfried Goeschl, The JUnit ++ Testing Tool, Dr. Dobb‟s Journal: Software Tools for the Professional Programmer, February 2001, Academic Search Premier, February 2001, Vol. 26, Issue 2.

14. Sylvia Ilieva, Valentin Pavlov, IlinaManova, A Composable Framework for Test Automation of Service-based applications.

15. Yuetang Deng, Phyllis Frankl, Jiong Wang, Testing Web Based Applications, ACM, September 2004 SIGSOFT Software Engineering Notes, Volume 29 Issue 5.

16. Cem Kaner, Jack Falk, Hung Nguyen, Common Software Errors, Testing Computer Software, Second Edition, 1993.

17. Hamzeh Al Shaar, Ramzi Haraty, Modeling and Automated Blackbox Regression Testing of Web Applications, 2008, Journal of Theoretical & Applied Technology, 4 (12), 1182 – 1198. EBSCO host database, ISSN# 19928645.

18. Edward Hieatt, Robert Mee, Going Faster: Testing the Web Application, IEEE Computer Society, August 2002, Vol 19, Issue 2, Pg (60 – 65), URL: http://www. csse. monash. edu. au/courseware/cse4431/testingWebApps-IEEE-2002. Pdf

19. Geoff Dromey, Climbing over the „No Silver Bullet" Brick Wall, IEEE Computer Society, 2006.

20. Panagiotis Louridas, JUnit: Unit Testing and Coding in Tandem, IEEE Computer Society, 2006.

21. Antonia Bertolino, Software Testing Research: Achievement, Challenges, Dreams, IEEE Computer Society, 2007.

22. Bernard Stepien, Liam Peyton, Pulei Xiong, Framework testing of web applications using TTCN-3.