

Comparative study on Software testing tools used for GUI testing

¹Abhishek V, ²Naveen A, ³Vidya G, ⁴Dr.Jasmine K.S

^{1, 2, 3} Students, Department of MCA, R.V. College of Engineering
Bangalore, India

⁴Associate Professor, Department of MCA, R.V. College of Engineering
Bangalore, India

Abstract

One of the most powerful movements in the information technology community today is the widespread adoption of several software testing tools for GUI testing. Using the right software testing tool is increasingly critical to project success, but the choices keep getting wider and more confusing.

This paper presents a straightforward Comparative study on software testing tools used for GUI testing. In this case, how different software testing tools of Open Source Software (OSS) and Proprietary software testing tools are used. GUI testing tools are used to test application's user interface and to detect if application is functionally correct. GUI testing involves carrying set of tasks and comparing the result of same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy.

Key Words— OSS, Proprietary, Testing Tools, Software Testing, GUI Testing

1. Introduction

GUI Testing includes how the application handles keyboard and mouse events, how different GUI components like menu bars, toolbars, dialogs, buttons, edit fields, list controls, images etc. reacts to user input and whether or not it performs in the desired manner[11]. GUI testing improves the overall look and feel of the application according to the requirements. GUI testing improves the overall look and feel of the application according to the requirements.

The various GUI components of these modules will be tested using different test cases. The test cases will contain the combinations of input which will give both correct and erroneous outputs. In case of erroneous outputs, an error message will be popped up and for correct input, the user will be allowed access to the system resources i.e. the documents available in the website. The important benefits of GUI Testing includes, higher test coverage levels, greater reliability, shorted test cycles, ability to do multi user testing at no extra

cost, all resulting in increased levels of confidence in the application and its successful deployment.

Comparative study on software testing tools like Open Source Software and Proprietary software testing tools used for GUI testing

2. Open source software

2.1. Why open source software?

Open source is a generic term for software that is intended to be distributed to anyone who wants it possibly under certain conditions determined by a licensing agreement. With the explosive growth of the Linux open source operating system over the last several years, the term has become increasingly commonplace.

Making money through traditional methods, such as sale of the use of individual copies and patent royalty payment, is more difficult and sometimes impractical with open source software. Some proprietary software advocates see open source software as damaging to the market of commercial software.

2.2 Why do people keep working on open source?

The desire to learn technical skills by joining an open project is strong. Typical reasons for staying in OSS are:

- improving skills: 32%
- ideology 31%
- improving software: 24%
- seeking recognition: 12%

Those who 'merely' deploy open source software are also part of the open source community

It is important to understand that software which is developed by companies such as HP, IBM, Novell or Sun using conventional methods is being released under an open source license just as effectively as that developed by a loose-knit community of students working at night to improve their programming skills.

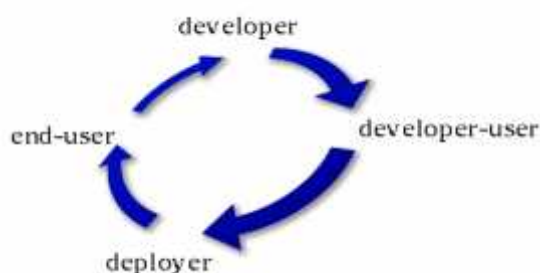


Figure 1. Open Source Software Communities

3. Proprietary software

3.1 Development process

Most traditional proprietary software projects use a variant of the waterfall model in their software development process.

3.2 Proprietary software infrastructure

Before the arrival of the Internet, proprietary development was the only economically viable way to create large, complex infrastructure (Figure 2).

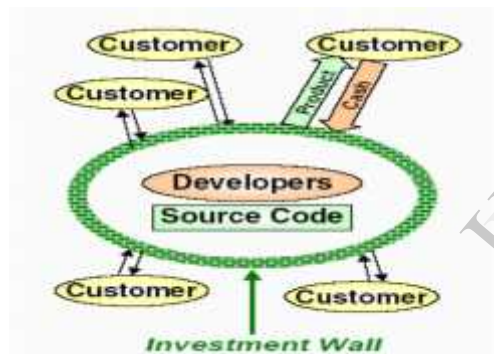


Figure 2. Proprietary software Infrastructure

Proprietary developers solved the stranding problem by using investment to cover early costs, with the condition that the early costs would later be recouped through sales of the resulting software. [8] To ensure the ability to sell the software later, proprietary projects required the use of investment walls [2] that behaved like one-way mirrors, allowing developers see out while preventing future customers from seeing in.

4. Distinction between open source and proprietary

The distinction between open source and proprietary is classified into four

- Design Architecture
- Organizational structure
- Communication and Control mechanism

- Testing processes (usually by developers). Testing, on the other hand, requires a running executable.

5. Testing tools

5.1. Open Source Software Testing Tools

5.1.1 Maverryx

Maverryx is an automated functional, graphical user interface (GUI), and regression test tool for Java applications. Unlike other test tools, Maverryx does not use a GUI Map to create and run its automated tests. GUI test objects are recognized at execution time, by a GUI Objects Finder. This search engine supports exact and fuzzy matching algorithms to identify the test objects in the application's user interface. Avoiding GUI Maps allows starting automation early—long before the application is available for testing, while approximate matching gives the possibility to derive tests even from partial or lacking requirements, and to automate scripts resilient to frequent application changes. Maverryx is primarily used by Software Quality Assurance teams to perform automated testing in traditional and agile environments.

5.1.2 QAliber

QAliber is a GUI test automation set of tools for testing web and desktop applications under Microsoft Windows operating system.

5.1.3 Selenium

Selenium is a portable software testing framework for web applications. Selenium provides a record/playback tool for authoring tests without learning a test scripting language (Selenium IDE). It also provides a test domain-specific language (Selenese) to write tests in a number of popular programming languages, including Java, C#, Groovy, Perl, PHP, Python and Ruby. The tests can then be run against most modern web browsers. Selenium deploys on Windows, Linux, and Macintosh platforms.

5.2. Proprietary Software Testing Tools

5.2.1 HP QuickTest Professional

HP QuickTest Professional software provides functional and regression test automation for software applications and environments. Part of the HP Quality Center tool suite, HP QuickTest Professional can be used for enterprise quality assurance. HP QuickTest Professional supports keyword and scripting interfaces and features a graphical user interface. It uses the Visual Basic Scripting Edition (VBScript) scripting language to specify a test procedure, and to manipulate the objects and controls of the application under test.

5.2.2 HP WinRunner

HP WinRunner software was an automated functional GUI testing tool that allowed a user to record and play back UI interactions as test scripts.

5.2.3 IBM Rational Functional Tester

IBM Rational Functional Tester is a tool for automated testing of software applications from the Rational Software division of IBM. It allows users to create tests that mimic the actions and assessments of a human tester. It is primarily used by Software Quality Assurance teams to perform automated regression testing.

5.2.4 IcuTest GUI unit testing

IcuTest is a unit testing framework for GUIs. The current version supports Windows Presentation Foundation applications. GUI verification is done primarily using image comparisons. Test suites can run interactively or fully automated.

6. Observations and findings

6.1. Results

When testing all these applications including open source software (PHP, Perl) using Selenium testing tool and proprietary software (ASP, JSP, JavaScript) using HP QuickTest Professional testing tool, the elapsed time for the open source software is less when compared to the proprietary software. The OSS test used in this dissertation itself is an open source test and it supports all the application software regardless of whether that software is installed in the system or not. It automatically generates test results in the form of a chart. From that chart the user can easily conclude which software is best when comparing to the other. The properties of test used in this dissertation for all applications are, Scheduled task group is set to 10 sec. The stop task group after fixed time is set to 5 seconds. The number of virtual user for the elapsed time with respect to timer values is set to 2.

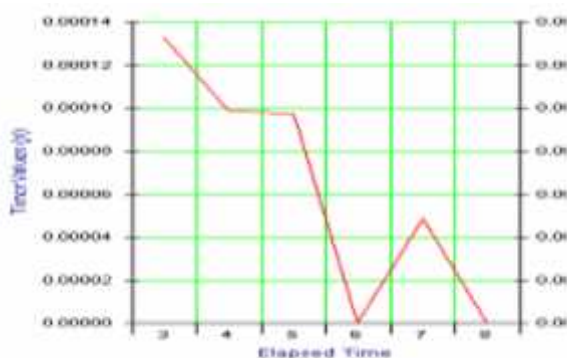


Figure 3. Chart showing Elapsed time & timer values for ASP

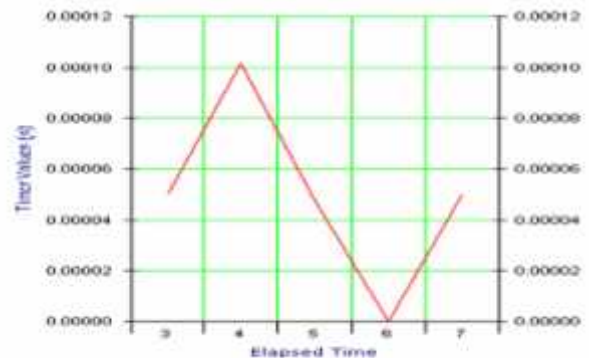


Figure 4. Chart showing Elapsed time & timer values for PHP

TABLE 1. Starting timer value for all applications

Si. No	Name of Application	Starting Timer values (sec)
1	JSP	10
2	ASP	14
3	Java Script	16
4	Perl	5
5	PHP	5

From the above charts, it was very clear that the Open Source Software application's Elapsed time is very less with respect to timer values whereas the elapsed time for the Proprietary software application is somewhat high. The above table showing the starting timer values for the open source as well as proprietary software.

7. Conclusions

Open Source Software, is the most reliable software, and in many cases has the best performance in GUI Testing. In this paper, when tested Open Source application (PHP, Perl) and Proprietary software application (ASP, JavaScript, JSP) are tested with Testing tools like Selenium and HP QuickTest Professional respectively, the performance measures like starting timer values of the Open Source Software testing tools is very less when compared with the Proprietary software testing tools. After running open source software testing tools using scripts and tests, the test report is generated very quickly for the same number of virtual user when compared to proprietary software testing tools.

Open source software testing tools has several strengths such as minimal cost, reusability, producing reliable source code, stability and security, the proprietary software testing tools also has some own strengths and weakness.

8. References

- [1] Success of Open Source – By Steven Weber, Harvard University, Published 2004.
- [2] R.C. Pavlicek, Embracing Insanity: Open Source Software Development, Sams, Indianapolis, Sept. 2000.
- [3] Learning from Open Source – By Evon Hippel - MIT Sloan Management Review, 2001
- [4] Open Source Software versus Proprietary Software – By JM Dalle, N Jullien, 2001.
- [5] Open Source Software Adoption – By H Wang, C Wang - Software, IEEE, 2001 - ieeexplore.ieee.org
- [6] B. Gates, N. Myhrvold, W.H. Gates, P.M. Rinearson, the Road Ahead: Completely Revised and Up-To-Date, Penguin, New York, Nov 1996.
- [7] Understanding Open Source Development – By Joseph Feller, Published 2001, Addison Wesley Professional.
- [8] For Opensta Testing [Http://Www.Opensta.Org](http://Www.Opensta.Org)
- [9] For Open Source Testing. [Http://Www.Opensourcetesting.Org/Resources.php](http://Www.Opensourcetesting.Org/Resources.php)
- [10] For Proprietary Software [Http://En.Wikipedia.Org/Wiki/Proprietary_Software](http://En.Wikipedia.Org/Wiki/Proprietary_Software)
- [11] For GUI testing. http://en.wikipedia.org/wiki/GUI_Testing.