International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 1
***Article Received:** 25 November 2022 **Revised:** 12 December 2022 **Accepted:** 30 January 2023*

Evaluation of Software Testing Tools Based on Quality Assurance Parameters

Dr. Mrs. Ashwini Atul Renavikar[1*]
[1*]Associated with Symbiosis Skills Professional University and QA Professional at Shrisoft Pvt. Ltd. Pune
E-Mail : ashwiinirenavikar@gmail.com
***Corresponding Author:** Dr. Mrs. Ashwini Atul Renavikar
*Associated with Symbiosis Skills Professional University and QA Professional at Shrisoft Pvt. Ltd. Pune
E-Mail : ashwiinirenavikar@gmail.com

Abstract : Free and open-source software (FOSS) is brought up, with increasing frequency, in discussions about digital technologies and economic development, in particular with regard to strategies for capacity building for information and communication technologies (ICT), the Internet and e-commerce, in developing and transition economy countries. Free testing tools can immensely affect the overall cost of software products. Research attempts to evaluate open-source testing tools for their effectiveness and usability for quality software product development through rigorous testing.

Keywords : *communication technologies, Assurance Parameters*

1.0 Introduction

1.1 Origin of FOSS

Software is an important component in the digital technology equation. But it is much more than that. At a personal level, it is the interface between humans and hardware that speaks the binary language of ones and zeros. For technology to be useful, it needs to perform in a human-accessible way: this is achieved though software. **Success of FOSS** FLOSS applications are first, second or third-rung products in terms of market share in several markets, including web servers, server operating systems, desktop operating systems, web browsers, databases, e-mail and other ICT infrastructure systems. FLOSS market share higher in Europe than in the US for operating systems and PCs, followed by Asia. These market shares have seen considerable growth in the past five years.

- FLOSS market penetration is also high – a large share of private and public organizations report some use of FLOSS in most application domains. In the public sector, Europe has particularly high penetration, perhaps soon to be overtaken by Asia and Latin America. In the private sector, FLOSS adoption is driven by medium- and large-sized firms. Almost two-thirds of FLOSS software is still written by individuals; firms contribute about 15% and other institutions another 20%.
- Europe is the leading region in terms of globally collaborating FLOSS software developers, and leads in terms of global project leaders, followed closely by North America (interestingly, more in the East Coast than the West). Asia and Latin America face disadvantages at least partly due to language barriers, but may have an increasing share of developers active in local communities.
- Weighted by regional PC penetration, central Europe and Scandinavia provide disproportionately high numbers of developers; weighted by average income, India is the leading provider of FLOSS developers by far, followed by China.
- While the U.S. has the edge in terms of large FLOSS-related businesses, the greater individual contribution from Europe has led to an increasing number of globally successful European FLOSS small- and medium-sized enterprises (SMEs).

1.2 Change in software economics on account of FOSS

While giving users more rights and freedoms may be a worthwhile initiative, real-world considerations require that the basic economics of provision be examined in order to appraise the possible role for FOSS within the software and ICT services sector. The fact is that a large amount of FOSS programs are developed and used, and a substantial number of applications have become world-class standards.

Moving from a business perspective to mapping the motivations of individual developers, several studies attempt explanations using conventional economic theory. An open-source programmer’s code can be associated with the author and well recognized, providing a certain level of ego gratification. Commercial companies frequently review contributions to and participation in FOSS projects when

***Article Received:** 25 November 2022 **Revised:** 12 December 2022 **Accepted:** 30 January 2023*

assessing employability. Open source leaders may get access to financing and attract attention from venture capital.

Direct economic impact of FOSS

The existing base of quality FOSS applications with reasonable quality control and distribution would cost firms almost Euro 12 billion to reproduce internally. This code base has been doubling every 18-24 months over the past eight years, and this growth is projected to continue for several more years.

- Firms have invested an estimated Euro 1.2 billion in developing FLOSS software that is made freely available. Such firms represent in total at least 565 000 jobs and Euro 263 billion in annual revenue. Contributing firms are from several non-IT (but often ICT intensive) sectors, and tend to have much higher revenues than non-contributing firms.
- Defined broadly, FLOSS-related services could reach a 32% share of all IT services by 2010, and the FLOSS-related share of the economy could reach 4% of European GDP by 2010. FLOSS directly supports the 29% share of software that is developed in-house in the EU (43% in the U.S.), and provides the natural model for software development for the secondary software sector.
- Proprietary packaged software firms account for well below 10% of employment of software developers in the U.S., and “IT user” firms account for over 70% of software developers employed with a similar salary (and thus skill) level.

This suggests a relatively low potential for cannibalization of proprietary software jobs by FLOSS, and suggests a relatively high potential for software developer jobs to become increasingly FLOSS related. FLOSS and proprietary software show a ratio of 30:70 (overlapping) in recent job postings indicating significant demand for FLOSS-related skills.

- Increased FLOSS use may provide a way for Europe to compensate for a low GDP share of ICT investment relative to the US. A growth and innovation simulation model shows that increasing the FLOSS share of software investment from 20% to 40% would lead to a 0.1% increase in annual EU GDP growth excluding benefits within the ICT industry itself – i.e. over Euro 10 billion annually
- Avoid penalizing FLOSS in innovation and R&D incentives, public R&D funding and public software procurement that is currently often anti-competitive
- Support FLOSS in pre-competitive research and standardization
- Avoid lifelong vendor lock-in in educational systems by teaching students skills, not specific applications; encourage participation in FLOSS-like communities
- Encourage partnerships between large firms, SMEs and the FLOSS community Provide equitable tax treatment for FLOSS creators: FLOSS software contributions can be treated as charitable donations for tax purposes. Where this is already possible, spread awareness among firms, contributors and authorities.
- Explore how unbundling between hardware and software can lead to a more competitive market and ease forms of innovation that are not favoured by vertical integration.

1.3 Open source testing tools

Increased global competition in the field of software development and frequently changing technology is forcing IT vendors and consultants to concentrate more on quality aspects. QA (Quality Assurance) is defined as QC (Quality Control) over QC. Immense variety of testing tools available and it becomes crucial to find out the suitability and adequacy of them for quality control purpose. Open source tools is another aspect that needs to be evaluated from technical and commercial point of view. The research study under consideration is an attempt to identify the right set of criteria for the evaluation of testing tools and collect the data from industry practitioners to know their views and opinions. The main objective of the study is to suggest a set of guidelines for identifying right open source testing tool and exploring the same. The study takes into consideration open source testing tools for functional testing, performance testing and also compare the same against proprietary tools available in the market. Inputs required for identifying right set of evaluation criteria and its aptness for the organizational requirements has been collected in the form of questionnaire and detailed interviews. Testing professionals (QA leads and testers) from renowned organizations in Pune (which is considered to be IT hub of India) will be approached for the study.

With increasing complexity and variety of software development, it is needless to say that quality becomes prime concern for any software development activity. Organizations are becoming more conscious about the quality of software as well as the process that is followed for development of that end product.

Quality Assurance Departments (QA) have a major role to play, as far as quality goals of organizations are concerned. Though defect preventions and process improvement is handled by QA departments, it is QC (Quality Control / Testing department which has equally important role to play. Identifying the defects in the software and report them to developing team is an important role that is played by testing department.

It has been found that software testing takes 30-40% of resources as far as software development activity is concerned As the size of the software grows, nature of testing task becomes complex enough and repetitive to be automated.

Variety of automated testing tools are available in the market with their own special features, pros and cons. It asks for huge investment as far as purchasing of the tools is concerned. Also, training of new tools is equally important aspect the needs to be handled. Open source testing tools add another facet for choosing right tools for the organization and software development activity under consideration. Study involves different aspects of testing tool evaluation process as far as commercial off-the-shelf and open source tools are concerned. Identifying right set of parameters for evaluation of a tool and arriving at the profitable decision is a big challenge that is faced by most of the software organisations. The study may help in handling the above task and choosing better option from the available ones , as far as automated testing tools are concerned.

Software quality parameters used for assessment of testing tools are - Functionality- Reliability, Usability, Efficiency,
Maintainability, Portability

Any testing tool is expected to perform functions/ tasks as far as testing activity is concerned. Some tasks are explained below, with their conventional meaning:

A: Test planning and monitoring B: Designing Test Cases C: Constructing Test Cases. D: Executing Test Cases.

E: Capturing and comparing test results. F: Reporting test results.

G: Tracking Software problem reports/defects. H: Managing the testware

2.0 Review of Literature and Significance of the study

In the paper, authors present the results of empirical study of the effects of open source software (OSS) components reuse on software development economics. Specifically, authors examined three economic factors – cost, productivity, and quality. This study started with an extensive literature review followed by an exploratory study conducted through interviews with 18 senior project/quality managers, and senior software developers. Then, the result of the literature review and the exploratory study was used to formulate research model, hypotheses, and survey questionnaire.

Open Source Software (OSS) has already been adopted by a large number of organizations. An important – but sometimes neglected – group of OSS users are Independent Software Vendors (ISVs). ISVs often develop their applications on top of OSS platform software. Frequently, this requires making several extensions and modifications to these OSS components. Researchers identify a number of challenges that ISVs face in handling these extensions and modifications. Next, authors describe several strategies ISVs can follow in maintaining these modifications. Finally, authors suggest an opportunity for a closer collaboration between OSS projects and ISVs which could be mutually beneficial. _ 2007 Elsevier Challenges and strategies in the use of Open Source Software by Independent Software Vendors Kris Ven *, Herwig Mannaert 0950-5849 Information and Technology Software Information and Software Technology 50 (2008) 991–1002

The emergence and market success of Linux in recent years has been impressive. The paper investigates the question of why some producers of proprietary software support the development of open source software (OSS) while others refuse any support. As an analytical framework, a simple Launhardt– Hotelling model is used to show that the emerging price pressure on the former monopolists depends on the extent of the current heterogeneity between OSS and the proprietary software of the incumbents. The paper argues that the product heterogeneity can explain the differing realworld behavior of commercial software producers. Commercial versus open source software: the role of product heterogeneity in competition Ju¨rgen Bitzer 0939-3625 Economic Systems Economic Systems 28 (2004) 369–381

Collaboration, peer review and open source software Justin P. Johnson Information Economics and Policy 18 (2006) 477–497 0167-6245

Open Source Software (OSS) has hit the mainstream in recent years and its scope is set to increase. Best seen as a range of associated licensing techniques, there are many different types of OSS licences. Coupled with a lack of settled case law and rapidly developing market practice, legal interpretation of the OSS world presents challenges to lawyers. Of the 'top 20' OSS licences, the GPL is the most commonly used and among the most radical in legal effect. The GPL's legal radicalism centres on its Article 2(b) concept of 'copyleft'.

This paper seeks to close an empirical gap regarding the motivations, personal attributes and behavioral patterns among free/ libre and open-source (FLOSS) developers, especially those involved in community-based production, and considers the bearing of its findings on the existing literature and the future directions for research. Respondents to an extensive web-survey's (FLOSS-US 2003) questions about their reasons for beginning to work FLOSS are classified according to their distinct ''motivational profiles" by hierarchical cluster analysis.

Commercial software companies face many challenges when competing in today's fast moving and competitive industry environment. Recently, the use of open source software (OSS) has been proposed as a possible way to address those challenges. OSS provides many benefits, including high-quality software and substantial profits. A strategic analysis for successful open source software utilization based on a structural equation model So Young Sohn , Min Seok Mok The Journal of Systems and Software 81 (2008) 1014–1024 ISSN- 0164-1212

Gaps for further research

After review of existing literature in the area of use of open source software in software development, specifically, efficient use of open source testing tools and current technology trends, the researcher has identified following areas which can be further explored for banks in the Indian context.

- Literature available is very scanty as far as use and evaluation of open source testing tools is concerned.
- No empirical work-based literature available for evaluation of open source testing tools
- Clear guidelines are not available for top management to use open source software as a strategy

3.0 Research Methodology

According to Cooper and Schindler (2003a), a useful way to design research study is as a two-stage design. a. Clearly defining research question b. developing research design.

3.1 Research Questions

- What are the factors that affect the quality and degree of adoption of open source testing tools?
- What is the status of use of open source testing tools as far as IT industry is concerned?

3.2 Objectives of the study:

a. To study the process for acquisition / evaluation of software testing tools
b. To identify the gaps in Software testing tool acquisition process
c. Suggest a model for evaluation of software testing tool evaluation / acquisition process

3.3 Hypotheses for the study

H1: Successful adoption of Open source testing tools is positively related to tool evaluation process.
H2: Degree of adoption of open source testing tools is an effect of software engineering processes adopted.

3.4 Research Methodology:

To achieve the objectives of the study, the researcher has explored a mix of quantitative and qualitative (triangulated) approaches.
The study explored different sampling techniques for choosing right sample representing different respondents and IT organizations under study.

3.5 Sampling Techniques

For sample selection, researcher will be referring to a study conducted by **Carey Schwaber,**

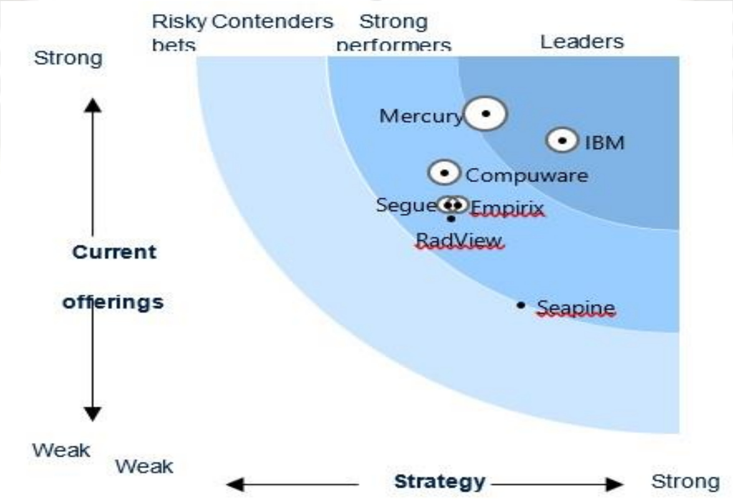


Figure : 1Source : Study conducted by Carey Schwaber, Analyst, Forrester Research – Feb 2004

With the above mentioned study as guideline, following tools are identified for the study

Table : 1 – COTS Tools

Sr No	Name of the tool	Vendor Name
1	TestPartner	Compuware
2	e-Tester	Empirix
3	Functional Tester	IBM Rational
4	QuickTest Professional	Mercury Interactive
5	WebFT	RadView
6	QA Wizard	Seapine
7	SilkTest	Segue
8	QALoad	Compuware
9	Testdirector	Mercury Interactive
10	Silkperformer	Borland
11	Purify	Rational

As far as open source testing tools are concerned, list of tools registered on www. sourceforge.net will be referred. This is the largest portal with highest number of open source tools registered on it. There are 6487 open source testing tools are registered on this portal. For selection of tools for study purpose, some aspects of these tools such as number of downloads, team size, defect tracking process etc. were studied. Following open source tools are selected for the study :

Table : 2 – Open Source Tools

Sr No	Tool Name
1	Bugzilla
2	Bugtrack
3	Aegis
4	Software Testing Automation Framework (STAF)
5	soapUI
6	Captura
7	ScreenHunter
8	ImageMagick import
9	IeUnit
10	WATIR
11	crashme
12	Open DTE

3.6 Reference period for the study

The data was collected for 2 months period, snapshot way.

3.7 Validity and Reliability of the data

Chi square and face validity techniques were used for checking validity and reliability of the data collection tool.

3.8 Statistical measurement

Content validity for the questionnaire designed for customer satisfaction survey, the responses were tested using face validity. Opinions of the research guide and other experts working in IT companies in the field of Quality assurance were sought during the process of testing the face validity. For fine-tuning the length of the questionnaire, number, wording and sequence of the questions, the researcher sought the opinion of an expert in the area of psychology and behavioral science.

3.9 Data Collection :

i) Survey The study uses the survey technique considering its advantages such as versatility, generalization, flexibility, sensitization to unknown problems and helps in verifying theories.

3.10 Questionnaire

The present study has deployed questionnaires in the following areas:

ii) Structured interviews The aim is for all interviewees to be given exactly the same context of questioning. This means that each respondent receives exactly the same interview stimulus as any other. The goal of this style of interview is to ensure that interviewees' replies can be aggregated … Questions are usually very specific and very often the respondents have a fixed range of answers (this type of question is often called closed, closed ended, pre-coded, or fixed choice).

iii)Unstructured Interviews rent from the other. The respondents are encouraged to speak openly, frankly and give as much detail as possible. The strengths of unstructured interviews lie in the fact that no restrictions are placed on questions. This method is useful when little or no knowledge exists about a topic and help to collect background data. Unstructured interviews are flexible and the researcher is able to investigate underlying motives.

Based on the sample selected, data was collected form 67 testing professionals in 12 software organizations in Pune working at different levels such as test engineers, test leads and test managers will be collected. Out of 12, two organizations have testing as their core competency. Software testing professionals with more than 5 years of experience in manual testing and at-least 2 years of experience in automated testing will be targeted for the study.

3.11 Data Analysis :

Licensed version of SPSS (Statistical Package for Social Sciences) ,required for data analysis. The same was used for the analysis of the data collected in the research work.

4.0 Data Collection and Analysis

Analyzing survey data is an important and exciting step in the survey process. As mentioned on the website of Research Methods Knowledge Base group, in most social science research the data analysis involves three major steps, which are performed in the following order:

- Cleaning and organizing the data for analysis (Data Preparation)
- Describing the data (Descriptive Statistics)
- Testing Hypotheses and Models (Inferential Statistics)

Data Preparation involves checking or logging the data in; checking the data for accuracy; entering the data into the computer; transforming the data; and developing and documenting a database structure that integrates the various measures. Descriptive Statistics are used to describe the basic features of the data in a study. They provide simple summaries measures., with simple graphics analysis. Inferential Statistics investigate questions, models and hypotheses

4.0 Data Collection Primary Data

i) Survey

The study used the survey technique considering its advantages such as versatility, generalization, flexibility, sensitization to unknown problems and helps in verifying theories along with quantitative or numeric description of trends, attitude or opinions of population by studying a sample of that population.

ii) Structured interviews

This technique of structured interviews was used by the researcher to explore the insights about the process for evaluation of open source testing tools activity in the IT companies.

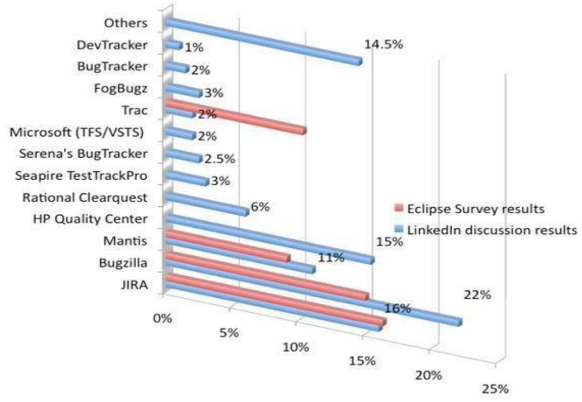
iii) Unstructured Interviews

This method is useful when little or no knowledge exists about a topic and help to collect background data. Unstructured interviews are flexible and the researcher is able to investigate underlying motives. **Secondary data** The researcher has followed above mentioned three steps for the analysis and interpretation of the data.

4.1 Analysis of Primary Data

Then researchers would like to look at the software recommendations from Domonic‘s discussion, which included nonopen source as well. The reason for this is that most people evaluating bug tracking and test management software are going to be looking for feature sets and functionality that are available in commercial tools. Hence researchers will also discuss what the most popular commercial software is, what they offer, and why they are considered for test management and bug tracking. If we can consider the discussions as a survey, figure shows the popularity of the different commercial tools available, including commercial and open source software. a. Details about The figure shows the popularity of the different open source bug tracking software tools discussed in the group discussion by the members.

Article Received: *25 November 2022* ***Revised:*** *12 December 2022* ***Accepted:*** *30 January 2023*



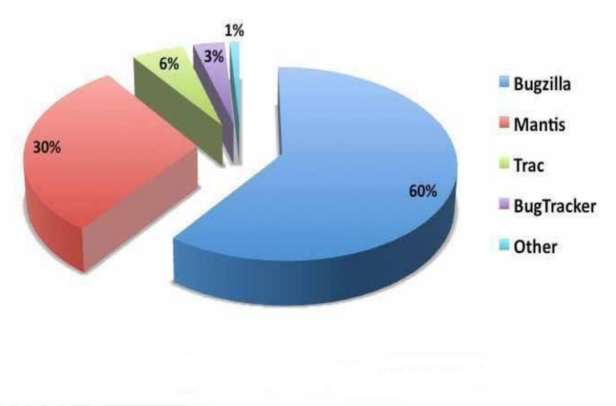


Figure 1

Figure 2

The figure 2 shows the popularity of the different commercial software tools discussed in the group discussion by the members of the business-oriented social networks.

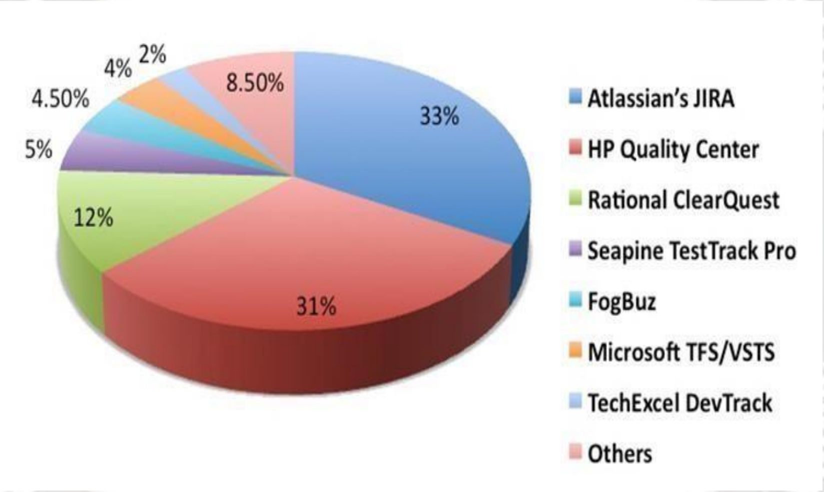


Figure 3

Tool	Current Version	Public Forums	Mailing Lists	CV Tracker	Documentation
Abbot	0.13.0	Very Active	2	Very Active	Excellent
DejaGNU	1.4.3	Fair	3	Fair	Good
JavaTest	0.00.08	Not Active	3	Not Active	None
Jfunc	1.1	Active	0	Active	Limited
Marathon	0.83	Very Active	3	Very Active	Excellent
Pounder	0.95	Not Active	0	Not Active	Good

QAT	2.7.2	Active	1	Active	Very Good

Tool	**Current Version**	**Public Forums**	**Mailing Lists**	**CV Tracker**	**Documentation**
Anteater	0.9.16	Good	3	Very Active	Excellent
HttpUnit	1.5.4	Good	2	Active	Good
JwebUnit	1.2	Not Active	2	Fair	Good
Bugkilla	0-1-0	Active	2	Active	Very Good
Grinder	3.0-beta 22	Not Active	3	Not Active	Excellent
Jameleon	2.0.2	Very Active	3	Active	Excellent
LogiTest	0.8.1	Not Active	2	Not Active	Good
Solex	0.5.0	Active	1	Active	Very Good
Tclwebtest	1	Active	1	Active	Good
TagUnit	1.0.1	Not Active	2	Active	Very Good
Web Form Flooder	0.2.8	Active	0	Active	Good
XmlTestSuite	1.2	Not Active	0	Not Active	Good

Based on ratings given to quality assurance parameters by the end user and its overall rating on effectiveness, multiple regression was applied and it was found that there is a strong connection between QA parameters and overall effectiveness of the tools.

4.2 Analysis of secondary data - Tools Overview Silk Test – Segue o Is an automated tool for testing the functionality of enterprise applications in any environment.

o Designed for ease of use, Silk Test includes a host of productivity-boosting features that let both novice and expert users create functional tests quickly, execute them automatically and analyze results accurately. **Watir (Ruby)**

o WATIR stands for "Web Application testing in Ruby". Watir is a free, open source functional testing tool for automating browser-based tests of web applications.

o Watir is a Ruby library that works with Internet Explorer on Windows. Like other powerful programming languages, Ruby gives you the power to connect to databases, read data files, export XML and structure your code into reusable libraries.

Selenium

o Selenium is a test tool for web applications. Selenium tests run directly in a browser, just as real users do. And they run in Internet Explorer, Mozilla and Firefox on Windows, Linux, and Macintosh. No other test tool covers such a wide array of platforms.

Pure Test

o PureTest is an application which is primarily used to setup scenarios of tasks, execute and debug them. Even though it supports testing a variety of applications it is especially useful for debugging and snooping of web applications. PureTest includes a HTTP Recorder and Web Crawler which makes it useful for generic verification of HTTP requests and web content checking.

MaxQ

o MaxQ is a web functional testing tool

o MaxQ records using a web site. It turns the links clicked on and any other input into a Python script that can be played back at any time. The tool can be used to:

o Check if the web site still works (regression test).

o Check if the web site is producing valid HTML.

o Automatically extract information from, or take some action on other web sites

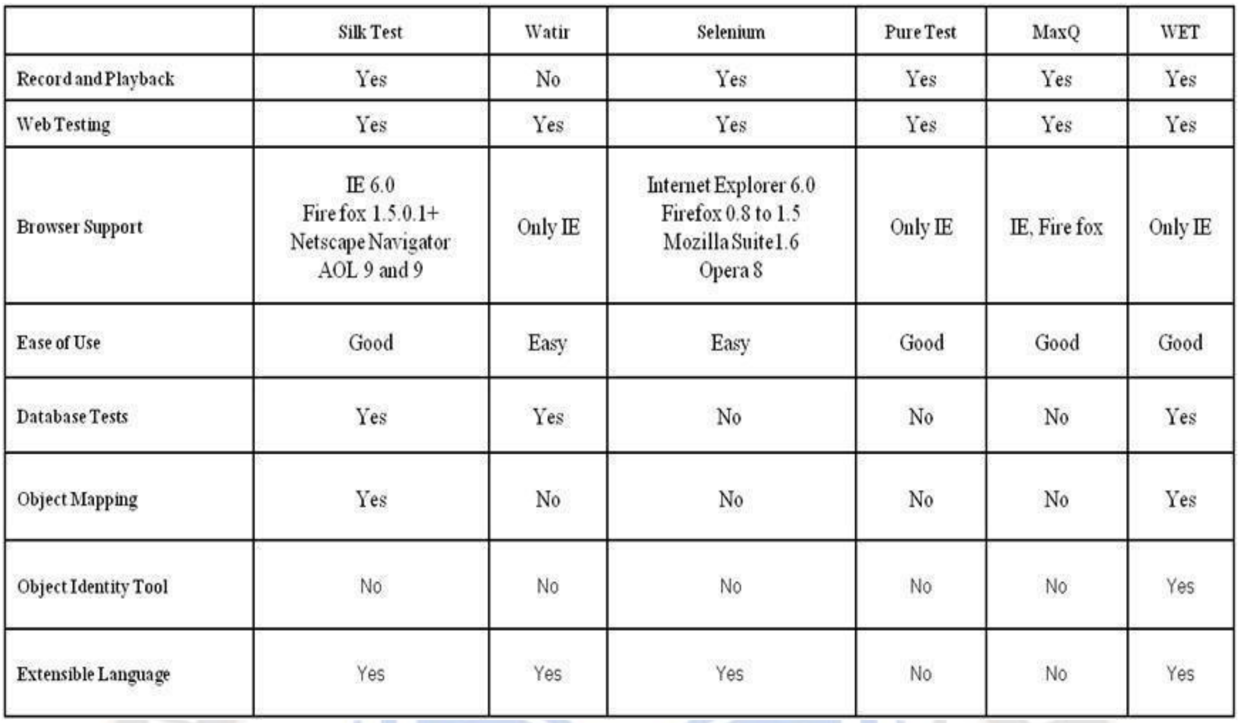
WET

o WET is a open source web automation testing tool which uses Watir as the library to drive web pages.

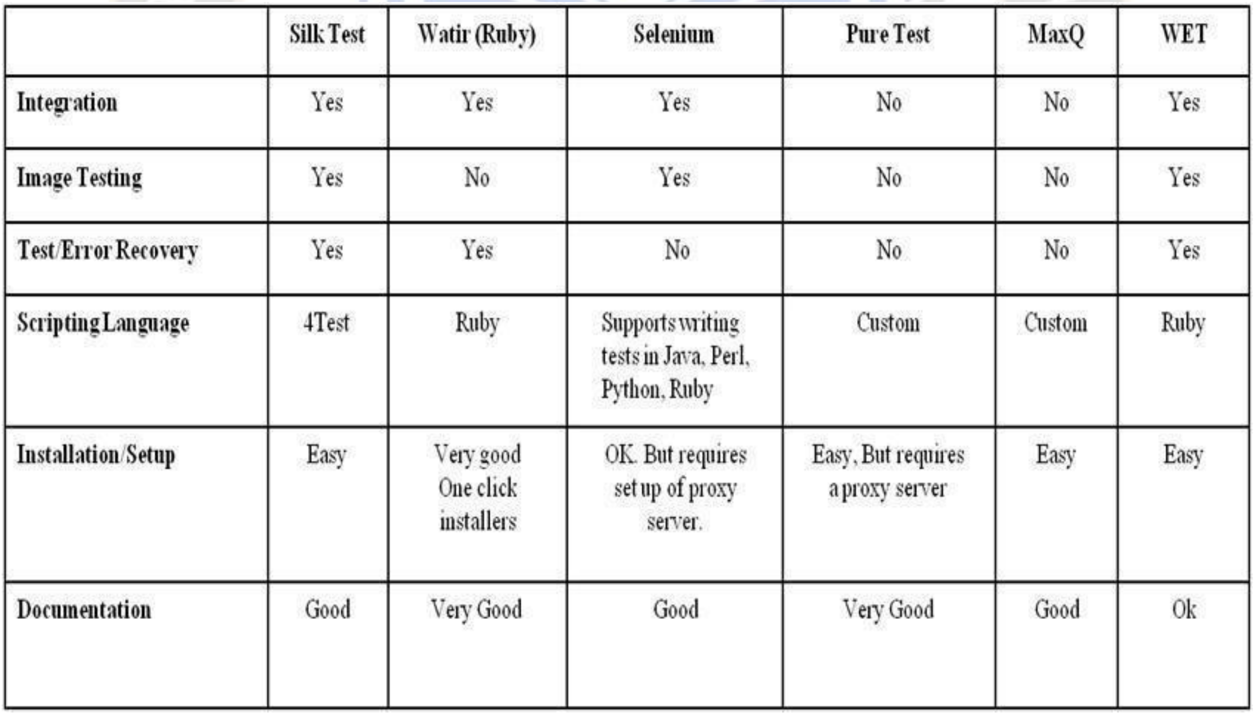
o WET drives an IE Browser directly and so the automated testing done using WET is equivalent to how a user would drive the web pages. Using WET, you can perform all the operations required for testing web applications - like automatically clicking a link, entering text in a text field, clicking a button etc.

***Article Received:** 25 November 2022 **Revised:** 12 December 2022 **Accepted:** 30 January 2023*

4.3 Comparative analysis of select open source testing tools



	Silk Test	Watir	Selenium	Pure Test	MaxQ	WET
Record and Playback	Yes	No	Yes	Yes	Yes	Yes
Web Testing	Yes	Yes	Yes	Yes	Yes	Yes
Browser Support	IE 6.0 Fire fox 1.5.0.1+ Netscape Navigator AOL 9 and 9	Only IE	Internet Explorer 6.0 Firefox 0.8 to 1.5 Mozilla Suite1.6 Opera 8	Only IE	IE, Fire fox	Only IE
Ease of Use	Good	Easy	Easy	Good	Good	Good
Database Tests	Yes	Yes	No	No	No	Yes
Object Mapping	Yes	No	No	No	No	Yes
Object Identity Tool	No	No	No	No	No	Yes
Extensible Language	Yes	Yes	Yes	No	No	Yes



	Silk Test	Watir (Ruby)	Selenium	Pure Test	MaxQ	WET
Integration	Yes	Yes	Yes	No	No	Yes
Image Testing	Yes	No	Yes	No	No	Yes
Test/Error Recovery	Yes	Yes	No	No	No	Yes
Scripting Language	4Test	Ruby	Supports writing tests in Java, Perl, Python, Ruby	Custom	Custom	Ruby
Installation/Setup	Easy	Very good One click installers	OK. But requires set up of proxy server.	Easy, But requires a proxy server	Easy	Easy
Documentation	Good	Very Good	Good	Very Good	Good	Ok

5.0 Findings and Conclusions

In the study under consideration, the researcher has tried to make value addition albeit a small one, to the body of knowledge through

A. Contribution by addressing research gaps
B. Contribution by tracking research questions
C. Contribution by fulfilling objectives

Major findings and conclusions

1. Some of the advantages found about open source testing tools are

a. First, enterprises find that they consistently get great value -- and the desired ROI -- from open source software. The quality of open source software met or exceeded the expectations of 92 percent of respondents in the survey.

b. Acquisition cycles and associated entry costs are minimal for open source software -- at least for
pilot projects and initial rollouts. Enterprises can use free versions of software
to start a project without having to endure the protracted sales and acquisition cycles that often accompany pilot projects with commercial software.

c. Open source applications can be even more secure than their commercial equivalents.
Open source communities fixed security vulnerabilities twice as quickly as commercial software vendors did.

2. IT industries in Pune are using open source testing tools but with lot of apprehension and caution.
3. User friendliness has been an issue about most of the open source testing tools.
4. Industries are open to accepting free software after strict evaluation by QA teams.
5. The recent recession has tightened the budgets of organizations the world over and most software and services companies have really felt the pinch. However, open-source companies have bucked this trend by exhibiting strong growth throughout. In the midst of the recession, an Economist article caught the zeitgeist with the headline. The opensource testing tools market is also
6. young and has not seen significant commercial investment yet.

Questionnaire
Dear Sir / Madam,
The researcher is an academician and presently collecting the data about evaluation of software testing tools. The research is purely academic oriented and strict confidentiality will be maintained about the data collected.
Please rank the following testing tool on the rank of 0-5 where 0 indicates that feature not available or not applicable. 1 indicates poor/ not suitable and 5 indicated most suitable / excellent. You can keep the column blank if you haven't used a particular tool. If your organization is using any other tool than mentioned in the list, please mention the same at the end of list.
Request you to spare some time from your busy schedule to fill up the questionnaire. Researcher will share the outcome of the research, if you are interested.

Personal Information:
Your rating about the Open source testing tool (0-5): 0-Not available, 1-Poor/not suitable, 5- Excellent/Most suitable

Tool	**Type***	**Cost**	**Record & play**	**Database**	**Object mapping**	**Usability**	**Test/error recovery**	**Technical support**	**Script Enhanceme**	**Any other (Pl. mention)**
Open Source testing tools										
Bugtrack										
Aegis										
(STAF)										
soapUI										
Captura										
ScreenHunter										
ImageMagick import										
IeUnit										
WATIR										
crashme										
Open DTE										
CppUnit										
JUnit										
Ruby Test::Unit										
NUnit										
OPENsta										

Any other tool you have used (Please mention name)										

***Type :**

Black box (Functional testing tool) 5.Performance testing tool

1. Bug tracking tool 6. Screen Capture tools
2. Browser based testing tools 7. Stress testing tools
3. Unit testing tools

Your rating about the Commercial testing tool (0-5): 0-Not available, 1-Poor/not suitable, 5-Excellent/Most suitable

Tool	**Type***	**Cost**	**Record & play**	**Database test**	**Object mapping**	**Usability**	**Test/error recovery**	**Technical support**	**Script Enhancemen**	**Any other (Pl. mention)**
Commercial testing tools										
TestPartner										
e-Tester										

***Type :**

1. Black box (Functional testing tool) 5.Performance testing tool
2. Bug tracking tool 6. Screen Capture tools
3. Browser based testing tools 7. Stress testing tools
4. Unit testing tools

Suggestions and recommendations :

Bibliography and References

[1] Empirical study of the effects of open source adoption on software development economics by Samuel A. Ajila, Di Wu, The Journal of Systems and Software 80 (2007) 1517–1529 2. Impact of user skills and network effects on the competition between open source and proprietary software by

[2] Lihui Lin, Electronic Commerce Research and Applications 7 (2008) 68–81 Challenges and strategies in the use of Open Source Software by Independent Software Vendors by Kris Ven, Herwig Mannaert, Information and Software Technology 50 (2008) 991–1002

[3] Commercial versus open source software: the role of product heterogeneity in competition by Jurgen Bitzer, Economic Systems 28 (2004) 369–381

[4] Self-organization of teams for free/libre open source software development by Kevin Crowston , Qing Li, Kangning Wei, U. Yeliz Eseryel, James Howison, Information and Software Technology 49 (2007) 564–575

[5] Bug-fixing and code-writing: The private provision of open source software by Jurgen Bitzer, Philipp J.H.

[6] Schroder, Information Economics and Policy 17 (2005) 389–406

[7] When does community participation enhance the performance of open source software companies? by Wouter Stama, Research Policy 38 (2009) 1288–1299

[8] Collaboration, peer review and open source software by Justin P. Johnson, Information Economics and Policy 18 (2006) 477–497

[9] Current developments in Open Source Software Richard Kemp computer law & security review 25 (2009) 569–582

[10] Why Open Source software can succeed by Andrea Bonaccorsi, Cristina Rossi, Research Policy 32 (2003) 1243–1258

[11] Community-based production of open-source software: What do we know about the developers who participate? By Paul A. David ,Joseph S. Shapir, Information Economics and Policy 20 (2008) 364–398

[12] A strategic analysis for successful open source software utilization based on a structural equation model by So Young Sohn , Min Seok Mok ,The Journal of Systems and Software 81 (2008) 1014–1024

[13] Open standards, open formats, and open source quality by Davide Cerri, Alfonso Fuggetta, The Journal of Systems and Software 80 (2007) 1930–1937

[14] Quality assurance under the open source development model Luyin Zhao , Sebastian Elbaum , The Journal of Systems and Software 66 (2003) 65–75

[15] The allocation of collaborative efforts in open-source software quality by Matthijs den Besten, Jean-Michel Dalle, Fabrice Galia, Information Economics and Policy 20 (2008) 316–322 16. Determinants of open source software project success: A longitudinal study by Chandrasekar Subramaniam ,Ravi Sen , Matthew L. Nelson , Decision Support Systems 46 (2009) 576–585

[16] The Open Source approach,– opportunities and, limitations with respect to security and privacy Computers & Security, Vol 21, No 5, pp 461-471, 2002

[17] How open source software works: “free” user-to-user assistance by Karim R. Lakhani, Eric von Hippel Research, Policy 32 (2003) 923–943

[18] Quality Factors and Coding Standards – a Comparison Between Open Source by Forges Andrea Capiluppi, Cornelia Boldyreff, Karl Beecher, Electronic Notes in Theoretical Computer Science 233 (2009) 89–103

[19] Open Source Software: An introduction by Mark Henley and Richard Kemp, computer law & security report 24 (2008) 77–85

[20] Measuring open source software success by Sang-Yong Tom Leea, Hee-Woong Kimb, Sumeet Guptab , Omega 37 (2009) 426 – 438

[21] From planning to mature: On the success of open source projects by Stefano Cominoa, Fabio M. Manenti , Maria Laura Parisi , Research Policy 36 (2007) 1575–1586

[22] An economic perspective on software licenses—open source, maintainers and user-developers by Kasper Edwards , Telematics and Informatics 22 (2005) 111–133 24. Something for nothing: management rejection of open source software in Australia’s top firms by Sigi Goode, Information & Management 42 (2005) 669–681 25. IPR indemnities in the open source and proprietary software worlds by Richard Kemp, Caspar Gibbons, Computer Law & Security Report (2005) 21, 420 -422

[23] An empirical analysis of open source software developers’ motivations and continuance intentions ChorngGuang Wu , James H. Gerlach ,

[24] Clifford E. Young , Information & Management 44 (2007) 253–262

[25] User acceptance model of open source software by M. Dolores Gallego, Paula Luna , Salvador Bueno, Computers in Human Behavior 24 (2008) 2199–2216