EVALUATIONS OF SOFTWARE TESTING TOOLS

F. Okezie
Department of Computer and Information Sciences
Covenant University,
Ogun State
Nigeria

Odun-Ayo
Department of Computer and Information Sciences
Covenant University,
Ogun State
Nigeria

S. Bogle
Department of Computer Science
Humboldt State University
USA

Päivi Raulamo-Jurvanen
M3S, University of Oulu
Oulu, Finland
paivi.raulamo-jurvanen@oulu.fi

Simo Hosio
UBICOMP, University of Oulu
Oulu, Finland
simo.hosio@oulu.fi

Mika V. Mäntylä
M3S, University of Oulu
Oulu, Finland
mika.mantyla@oulu.fi

Abstract

In this paper, we will understand the various concepts involving software testing and these are taken by combining several research papers.

Software testing is a crucial phase in the process of software development as it identifies defects and error if present any. This will ensure that the developed software is of good quality. So, the most challenging task is choice of testing tool that suits the project along with considering the two important factors i.e., time and cost.

There are two types of testing, manual and automatic. Manual testing involves informal review, inspection, walk through and technical review by a tester which takes lot of time and effort. So, most of the testers have switched from manual to automatic to reduce the time and cost of testing process. Automation testing is categorized into four: performance testing, safety testing, accuracy testing, and testing of reliability. Testing should be done before deploying the software to ensure that it is defect free and all the functionalities are working as per the requirement. If there are no errors in testing phase, then does that mean the software quality is good? Not really, there are various other factors like performance, response time, reliability, reusability, maintainability, correctness etc., which determines the quality of software, it is also achieved with testing.

Further, different other research papers have been studied to understand the most widely used automation tools in software testing, compared their characteristics based on certain criteria to evaluate their performance in different environments so that it can beneficial for both researchers and practitioners. Also, practitioner evaluations have gained much interest in selecting testing tool. A survey is conducted of 89 members and analyzed the reliability of opinions using Krippendorff's alpha, intra class correlation coefficient and coefficient of variation.

In order to recommend the best testing tools, many studies have been conducted and found that "Appium" is appropriate tool for android platform and "Ranorex" is best for website application. Appium is the most popular open source framework for mobile app automation testing. Ranorex offers a wide range of testing tools to integrate with most popular continuous integration tools, test management tools and task scheduling tools. The results of the survey indicated that on average, opinions from seven experts gives a moderate level of reliability.

Keywords—Krippendorff's alpha, intra class correlation coefficient and coefficient of variation, software testing, testing tools

I. INTRODUCTION

Software testing is a crucial phase in the process of software development as it identifies defects and error if present any. This will ensure that the developed software is of good quality. So, the most challenging task is choice of testing tool that suits the project along with considering the two important factors i.e., time and cost.

There are two types of testing, manual and automatic. Manual testing involves informal review, inspection, walk through and technical review by a tester which takes lot of time and effort. So, most of the testers have switched from manual to automatic to reduce the time and cost of testing process. Automation testing is categorized into four: performance testing, safety testing, accuracy testing, and testing of reliability. Testing should be done before deploying the software to ensure that it is defect free and all the functionalities are working as per the requirement. If there are no errors in testing phase, then does that mean the software quality is good? Not really, there are various other factors like performance, response time, reliability, reusability, maintainability, correctness etc., which determines the quality of software, it is also achieved with testing.

II. MATERIALS AND METHODS

There are various factors that should be considered before choosing a tool for testing and it is considered to be the basic and important step in the achievement of test automation. A testing tool can be used for different application testing's like web, desktop, mobile, or a combination of these. Also, it involves any testing functionality like unit test, regression test, integration testing etc. The tools discussed in this paper

are the ones most widely used by industry practitioners, their strengths, weakness based on certain factors like reusability, reliability , cost etc.,

The other way that has gained interest in recent times is that practitioner evaluations are credible than empirical evidence. The authors of "practitioner evaluation of software testing" research paper have conducted an online survey of 89 members and analyzed the reliability of opinions using Krippendorff's alpha, intra class coefficient and coefficients of variation. In context of software process improvement, it is accepted that practitioners prefer the opinions of their equals over empirical evidence.

A. Software Test tool Selection

For tool selection, there are different more or less comparison tools available, such resources might be useful for tool selection, but major challenge is that contents are neither generalizable nor validated for tools selection. Related academic studies rely on surveys as the key methodology but only a few report test tools by name. In literature, test tool evaluations tend to propose and include tasks like live trails, proof of concepts and demos. Such task requires resources and competence hence considered to give wrong results. So, investigating solutions and methodologies to help making sense of testing tools is important.

B. Developers Beliefs and Opinion Surveys

It is general that people get influenced by strong opinions and beliefs obtained from peers from their personal experience rather than from empirical research. Similarly, author Rainer says that even software practitioners find local opinions more credible than empirical evidence.

Opinion surveys is a common way of describing public's collective sentiment for some defined need. They act as the promising complementary way for understanding the collective public opinion. The author Hosio have developed a lightweight decision support tool for surveying large pools of users for subjective opinions on how a given solution fairs in light of various criteria. The data further can be modeled for results that suit a desired criteria configuration. This is called concept of the wisdom of crowds.

C. Assessment Of Responses

There are also studies which exhibits low value for expert agreement/reliability using Krippendorff's alpha or ICC by authors Borg, Anvaari and Kitchenham. Evaluations depend on the interpretation of construct under study which means it includes some degree of subjectivity. Sometimes, even a small group of experts can provide as accurate evaluations as large group.

We further see explanations of Krippendorff's alpha as a measure for the agreement among observers (respondents), intra class correlation as a measure of reliability of evaluations and coefficient variation to evaluate agreement. Then, we look at the how the number of respondent's effect on the accuracy of evaluations and the effect of demographics on tool evaluations.

D. Opinion Surveys

Authors have conducted survey and the criteria to be evaluated were Applicability, Compatibility, Configurability, cost effectiveness, costs, cross platform support, easy to deploy, easy to use, expandability, further development, Maintenance of Test cases and data, performance, popularity, programming skills and reporting features.

The respondents had chance to select one or more tools and evaluated the criteria of choice for each tool, one tool at a time. Tool list of 100 were created which are identified by practitioners for software testing. The respondents could indicate the basis of their evaluations that is whether they are from personal experience, or on a general opinion. Evaluations were on a scale of 0-10 at a interval of 0.5 where the default value is being 5 and using slider as the UI input element. Both the questionnaire and survey tool were validated by authors and industry partner.

The details are survey are survey 1 was published on Aug 29th 2016 in Finland to Finnish software testing professionals and then posted a link to selected groups to survey in various social platforms like Twitter, LinkedIn, Reddit and sent a link to public email list of a testing association in Finland. Received 21 responses with useful data and decided to send survey 2.

Second survey got published on Mar 1st, 2018, they used the same online tool but with precise focus to make sure fair amount of valid responses. The authors contacted a number of practitioners from a set of Finnish collaborating companies in Eureka Itea3 Testomat research project. The selected practitioners were known to be either familiar Robot framework, an open source, generic test automation framework for acceptance testing and acceptance test driven development.

It is promoted by sending email to seven professionals from six companies asking them to distribute the link to their colleagues considered relevant for answering the questions. Similar approach is also known as "snowball" or "chain sampling" has been used by Agerfalk and Fitzgerald. To reach a wider audience, the survey was promoted in robot framework slack and in Twitter with hashtag robot framework. They received 68 responses with useful data.

III. METHODS FOR ANALYZING THE DATA

A. Outliers in data

It is important to identify outliers in the data. For this, Tukey has proposed a rule of thumb for detecting outliers that is on the basis of quartiles. He defined an outlier as a value more than 1.5times the IQR(Q3-Q1) from the quartiles which is either below Q1-1.5*IQR or above Q3+1.5*IQR. It is important to study and identify outliers in the data as these are nothing nut noise, error or legitimate data but also can be "inspiration or inquiry"

B. Krippendorff's alpha

It is a statistical measure for determining the inter-rater reliability. It's value ranges from 0- perfect disagreement to 1- perfect agreement. The value alpha >=0.800 suggest drawing reliable conclusions while 0.667<=alpha<0.800 are said to be tentative conclusions only. Rfunction kripp alpha is used to measure the level of agreement among respondents on the criteria of the top 6 most evaluated tools.

C. *Intra Class Correlation Coefficient*

It is a statistic measure using for measuring inter rater reliability for ratio type of data. As for alpha, the value of ICC varies between 0 and 1, higher values indicating greater reliability. The commonly referenced ICC values are >=0.90 for excellent, 0.75<= and <0.90 for good, 0.50<= and <0.75 for moderate and <0.5 for poor agreement.

Rfunction ICC is used to estimate the association among the respondents for the top 6 tools. The function provides results for six different forms, presented as two numbers, i.e., ICC(x,y) where x indicates the model and the y indicates the type of measurement protocol. As the results may differ and lead to different interpretations, it is recommended to report both the results and the computational variant. To choose the correct form the prerequisites are analyzed suggested by Koo and Li.

1) Do we have the same set of respondents for all criteria?

Yes, same set of respondents evaluated all criteria.

2) Is the sample of respondents randomly selected from a larger population or is it a specific sample of respondents?

Authors had specific sample of respondents. They evaluated the same criteria, but the underlying contexts and constructs may vary for samples. Hence, there is no thought about generalizing the tool related results regarding the values as such, but to analyze reliability of responses.

3) Are we interested in the reliability of a single respondent or the mean value of multiple respondents?

Authors were interested in the reliability of the mean value of many respondents.

4) Are we concerned about consistency or agreement?

They wanted to check consistency and not absolute agreement.

First two questions – to guide the selection of model

Third question – type

Last question – difference of purpose

Authors have measured ICC using two-way mixed effects, average measures for consistency i.e., ICC(3,k) with the purpose to estimate the degree the respondents provided consistency in the evaluations. In reporting the results authors have followed the guidelines suggested by Hallgren and Koo and Li. Cases where single measured ICC's (ICC2) and average measured ICC's (ICC3) are high, suggested to report that both cases to show discrepancy.

D. *Coefficient Of Variation*

Measured the CV for the criteria evaluations for top 6 tools to analyze the extent of variability in evaluations in relation to the mean of population. Variation and CV are directly proportional meaning lower the CV the less the variation. Since our criteria is different i.e., some are more human oriented than other like programming, costs, CV allows to compare the difference across different criteria having different means.

As data is considered to be ratio type, and the scale is 0 to 10 at interval 0.5 i.e., 21 levels, authors calculated CV for both ratio and ordinal type of data. Formulas are:

(1) CV for ratio type of data

$CV = \sigma/x = \text{sqrt}(\text{var}(x))/\text{mean}(x)$

(2) CV for ordinal type of data.

$$\Delta = \sum_{i<j}\sum |i-j| P_i P_j$$

$$\Delta^{\star} = [4/(k-1)]\Delta$$

$$CV = 1 - (1-\Delta^{\star})^{1/2}$$

E. *Number of Respondents for ICC*

To analyze the effect of number of respondents to the incremental accuracy, authors applied the example modeled by Libby and Blashfield. They tested the effects of group size on decision making and concluded that having three accurate judges could improve the average performance.

Authors have generated random sets of respondents from 2 to n where n is the total number of respondents for a tool as in below table for each top 6 tools. For each size of sets from 2 to n, ran 100 iterations of ICC with an intention to compare the medians of the groups to the common ICC reference values. Hence, total number of ICC values for the tools ((n-1)*100) were 400 for Appium (n-5), 900 for Jenkins (n=10), 300 for Jira (n=4), 400 for JMeter (n=5), 7600 for Robot Framework (n=77) and 400 for selenium (n=5).

TABLE I. TOP SIX TOOLS SURVEY DATA

Tool	Survey#1[1]		Survey#2[1]		Total[1]	
	Resp	Eval	Resp	Eval	Resp	Eval
Appium	2	23	3	45	5	68
Jenkins	5	75	5	75	10	150
Jira	3	45	1	15	4	60
JMeter	5	68	–	–	5	68
RFW	9	119	68	998	77	1117
Selenium	1	15	4	47	5	62

[1] Number of respondents and evaluations for a tool

F. *Effects Of Demographics*

Authors have conducted negative binomial regression analysis with R function glm.nb6. They used an automatic method, R function stepAIC to analyze proposed variable selection. For baseline model, seven variables were included i.e., familiarity with Robot Framework, experience regarding the use of the tool, years in the current role and in work area, type of role and work area and business domain.

IV. DISCUSSION OF TOOLS AND ITS APPLICATION

A. *Selenium*

It is a free and open source testing tool that is used for web application testing. It supports different browsers and platforms like windows, Mac, Linux and languages like to compose tests like Java, PHP, C#, python, Groovy, Ruby and Perl. It includes 4 essential segments i.e., Selenium IDE, Selenium RC, WebDriver, Selenium Grid, helps in automation testing of functional parts of online applications. It can be utilized to perform black box testing on web applications.

Advantages include ease of use, flexibility, enabling users to debug, set breakpoints in test cases, can be changed into different programming languages so testing can be done on dynamic web applications.

Challenges include it does not support conditional and iteration statement, database testing and no error handling capacity, a level of programming skills is required to write test cases.

B. *TestComplete*

It is an automated testing tool used for functional testing. With this, testers can enable tests for windows , web, android and iOS. Automated scripts can be composed in different programming languages like python, C++ script, VB script, Jscript. Supports different testing types and methods like functional, unit and GUI testing. TestComplete is not an opensource and requires a license after free trail.

C. *Ranorex*

It tests applications from the user viewpoint. Supports testing in desktop, web and mobile applications, and a variety of platforms which includes Android, iOS and windows. Test scripts are written in number of languages like C#, python, C++, VB.net and XML. It has many good features like reusable test codes, integration with various tools, GUI recognition, record and playback, bug detection etc., Also performs platform compatibility testing to ensure excellent quality software. A significant advantage is that it is user friendly, inexpensive and can be used by any testing team and organization.

D. *Appium*

It is a new revolution in automate testing that provides effective, bug free and high-quality apps, which saves a project time, cost and labor. It is an opensource mobile testing tool which allows developers to write tests on various platforms like iOS and android. It supports different languages like Java, python, ruby, JavaScript, etc., It consists of three parts, Appium server, Inspector and Doctor. Appium tests user interaction with mobile apps content and the results are used to find accuracy.

E. *Quick Test Professional*

It is used to test applications on desktop or web, it offers testing for regression and functional testing. It uses VB script to run the scripts and to list the test forms and it also controls testing of applications. Automating regression testing and functional testing is possible with QTP. It is mainly used for UI based test cases automation, but some non-UI based cases like file system activities and database testing can also be automated.

Advantages of QTP is that it comes with a user friendly and simple IDE which makes it easy to understand by non-programmers enabling them to create and add test cases with ease. It further supports various add-ins like Java, Oracle, SAP.

Disadvantage is that its licensed and maintenance costs are high and multiple instances cannot be created. It runs slowly when compared with other open sources testing tools like selenium.

F. *OpenScript*

It is an IDE for Oracle application Testing and is based on Eclipse as a module to help creation and upkeep of functional and load testing automated scripts. Tool has the ability to record the applications in various browsers like Internet Explorer, Mozilla Firefox, and Google chrome. It has easy to use UI features which lets non software engineers to use it easily.

G. *Janova*

It is web based automated software testing which runs securely in the cloud; hence it is faster than traditional web testing tools and simple setup does not have installation. This tool does not require any scripts to be written hence it is easy to use by programmers and non-programmers. Even though it is not open source, its license cost is not expensive.

H. *Rational Functional Tester*

It is an OOP based automated testing tool which includes regression, functional, GUI and data driven testing. Supports with Java, web based and Microsoft Visual Studio, .Net applications. It ensures that test cases are maintained and executed properly by Quality Assurance departments of organizations.

The choice of specific testing tool is based on the application type, complexity, cost related to tool and the budget of testing stage in the organization.

V. RESULTS

Overall summary of the different tools for various categories is presented in below table. It helps industry practitioners in selection of tool for testing; be it small scale or large-scale project.

TABLE II. TOOLS FOR DIFFERENT CATEGORIES

Tool/ Criteria	Open Source	License	Supporting Platform (Mobile)	Supporting Platform (web)	Supporting Platform (Desktop)	Learning Ease/ Ease of Use	Coding/ Programming Skills	Code Reusability	Test Results Report	Record & Playback
Selenium [12, 19, 20]	✓			✓		✓	✓	✓	Plug In	✓
TestComplete [21]		✓	✓	✓	✓	✓	x		✓	✓
Ranorex [19, 20, 22, 23]		✓	✓	✓	✓			✓		✓
Appium [23, 24]	✓		✓					✓		✓
Quick Test Professional [12, 19,20]		✓		✓	✓		x	✓	✓	
OpenScript [25]		✓		✓	✓	x		✓	✓	✓
Janova [22]		✓		Cloud based			x		✓	✓
Rational Functional Tester [22]	✓			✓		✓				

From the above table we see that out of 8 testing tools, 5 of them require license and remaining 3 are open source. The applications and platforms each tool supports are given. We see that some tools are easy to learn while other needs programming skills.

With this, we know what tools to select for testing based on the requirement so that it would save cost and time. If the project to be tested is a small one, then open source tools are better rather than incurring huge cost for licensing ones.

From the criteria selected, we say that the choices of selecting open source tools are less when compared with that of licensed ones. The authors state that 17% of tools support web platform testing, 10% tools support desktop platforms while 8% support mobile platforms. There are some tools which performs well across all tools like TestComplete and ranorex.

Programing knowledge is required to use Selenium testing tool effectively. 5 testing tools supported the ability to reuse the code for similar testing instead of rewriting the code from scratch. 10% of the tools had the capacity to document test results and generate reports after every test. 17% of tools supported record and playback feature which supports improve efficiency when using a testing tool.

Below are pie and bar chart which summarizes the above table and results from it.

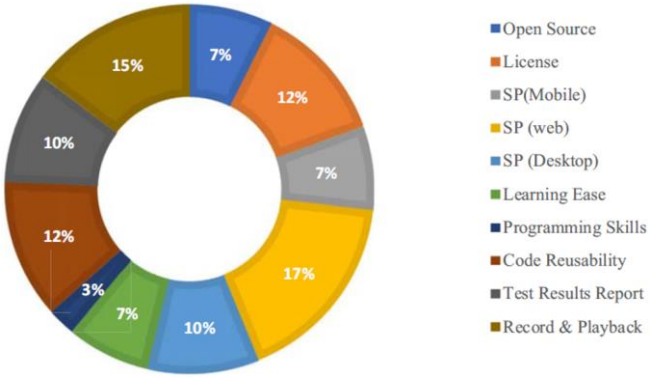


Fig. 1. Pie Chart Of Tools Composition

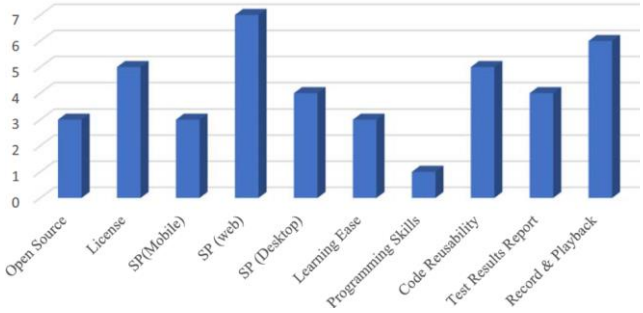


Fig. 2. Bar Chart Of Tools Composition

Below table represents the tools and its respective testing type.

TABLE III. TOOLS AND ITS TESTING TYPE

Tool	Testing Type
Selenium	Functional testing
TestComplete	Functional testing, Graphical User Interface testing, Unit testing
Ranorex	Graphical User Interface testing, Compatibility testing
Appium	Graphical User Interface testing, Functional testing
Quick Test Professional	Functional testing, Regression testing
OpenScript	Functional testing, Load testing, Database testing
Janova	Functional testing
Rational Functional Tester	Functional testing, Regression testing, Graphical User Interface testing

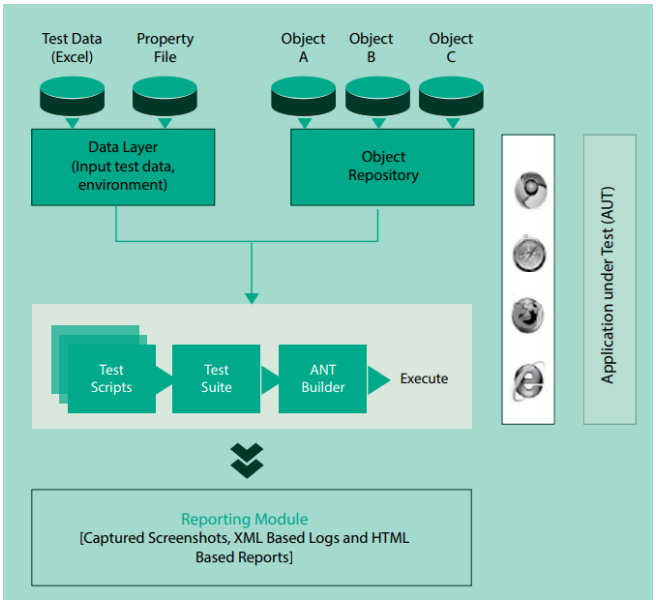


Fig. 3. Implementing hybrid Framework with Selenium

TABLE IV. COMPARISION OF SELENIUM, QUICK TEST PROFESSIONAL AND TESTCOMPLETE

FEATURES	SELENIUM	QUCIK TEST PROFESSIONAL	TESTCOMPLETE
Licensing Cost	It is open source. So, there's no licensing or renewal cost for this tool. It's free of cost.	Licensed and very Expensive, Ten user license costs approx. 60L.	$2K Enterprise Seat License
Application support	Web Applications only it supports addition of plug-ins to achieve desired results that are not provided by Selenium Core. Since, selenium is open source, plug-ins are also available free of cost.	A client server application Only. It also supports add-ons, but user needs to purchase license for them.	All of this included right out of the box there are no plug-ins or add-ons to buy. You can install Test Complete and immediately create any test against any application.
Object Oriented Language support and Scalability	Supports Java, .Net, Perl, PHP, Python and Ruby.	Scripts can be developed only in VBScript or JavaScript.	Test Complete supports scripting in VBScript, JSScript, DelphiScript, C++Script and C#Script, so you can create scripts in the language.
Support for operating system/platforms	Supports Windows PC/MAC/UNIXPlatforms.	QTP supports only Windows XP.	Windows 7, Windows Vista, Windows Server 2008 or later operating systems.
Programming skills	For using Selenium one needs to have programming skills.	QTP is quite easy to use. It is quite easy to edit the script, parameterize, navigate, playback and validate the results.	TC is good for both web based and desktop application.
Usage	Selenium needs quite a bit of expertise	QTP is quite easy to learn in a short time.	Support for all 32-bit and 64-bit window application.
Database applications	With Selenium one needs to exert hard to do the same job.	QTP works very well with database applications.	TC works very well database application.
Platform dependency	With Selenium these tasks can be easily accomplished.	It is difficult to deploy smoke tests for web applications using QTP especially with Windows7.	It is difficult to deploy application using.
Report Generation	Selenium users don't enjoy such luxury as enjoyed.	With QTP we can easily generate most comprehensive reports due to the availability of an efficient online help.	Report generation is an easy-to-use utility that is support along with TC and lets you generate dump files.

If our test automation requirements are getting fulfilled with TestComplete then no need to go for Quick test professional by incurring high cost. Both these tools serve the same purpose just that QTP is a versatile tool for critical and more risky applications under test.

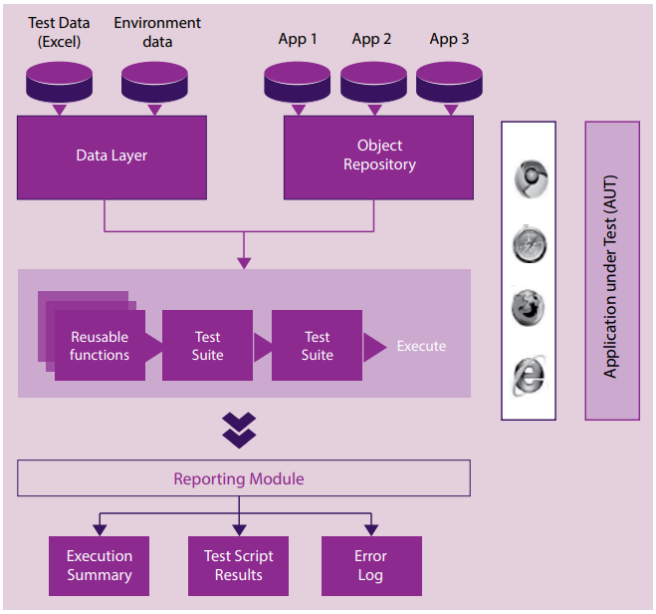


Fig. 4. Implementing Hybrid Framework with QTP/UFT

The survey for practitioner evaluations included evaluations for 2128 criteria, for 38 unique tools. All the test cases, duplicates or having default values have been removed.

Top 6 most evaluated tools in the surveys, namely Robot Framework, Jenkins, Appium, JMeter, Selenium and Jira, received 1525 evaluations in total.

The arithmetic mean of evaluations for the top 6 tools in below table.

TABLE V. # OF OUTLIERS & RANK OF CRITERIA

Rank	Criteria	Appium		Jenkins		Jira		JMeter		Robot Fw[a]		Selenium	
		Eval.[b]	O.[c]	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.
6	Applicability	66.0^{6}	0	84.0^{5}	1	83.8^{5}	0	79.0^{6}	0	83.0^{4}	5	62.0^{7}	0
8	Compatibility	52.0^{9}	0	84.5^{4}	0	77.5^{6}	0	71.7^{12}	0	81.8^{5}	4	56.0^{9}	0
7	Configurability	70.0^{5}	0	81.0^{8}	0	77.5^{6}	0	82.0^{5}	1	80.0^{7}	2	68.3^{5}	0
4	Cost-Effectiveness	61.3^{7}	0	86.5^{2}	0	61.3^{13}	0	83.8^{3}	0	89.4^{2}	5	76.3^{3}	0
1	Costs	84.0^{1}	0	86.0^{3}	1	47.5^{15}	0	93.0^{1}	0	92.5^{1}	8	72.5^{4}	0
5	Cross-Platform Support	77.5^{3}	0	83.5^{6}	1	85.0^{4}	0	73.0^{10}	0	83.7^{3}	1	66.3^{6}	0
9	Easy To Deploy	39.0^{13}	0	70.5^{12}	0	57.5^{14}	0	84.0^{2}	0	79.4^{9}	8	51.0^{12}	1
9	Easy To Use	45.0^{11}	0	60.0^{15}	0	73.8^{9}	0	77.5^{9}	0	78.9^{10}	6	56.3^{8}	0
13	Expandability	46.3^{10}	0	78.5^{9}	0	73.8^{9}	0	59.0^{14}	1	78.4^{11}	2	38.8^{15}	0
3	Further Development	78.8^{2}	0	82.0^{7}	0	86.3^{3}	0	78.8^{8}	0	80.1^{6}	2	82.5^{1}	0
12	Maintenance of TC&D	53.8^{8}	0	72.0^{11}	0	65.0^{11}	0	72.0^{11}	0	76.8^{13}	3	53.8^{11}	0
11	Performance	36.0^{14}	0	77.0^{10}	1	65.0^{11}	0	79.0^{6}	0	78.2^{12}	3	55.0^{10}	0
1	Popularity	72.0^{4}	0	89.0^{1}	0	87.5^{1}	0	83.3^{4}	0	70.4^{14}	1	82.5^{1}	0
15	Programming Skills	27.5^{15}	0	63.5^{13}	0	87.5^{1}	0	50.0^{15}	0	62.4^{15}	5	43.8^{13}	0
13	Reporting Features	42.5^{12}	0	63.0^{14}	0	75.0^{8}	0	66.0^{13}	0	79.8^{8}	0	43.8^{13}	0
	Total # of outliers		0		4		0		2		55		1

Practitioners tend to perceive Jira as a tool for testing seemed reliable because Bringing testing capabilities within Jira helps tightly integrate product management development and testing to streamline efficiency and productivity.

In both surveys, Robot Framework was most evaluated. It is a byproduct of two obvious facts.

1) *It is a local tool among the respondents*
2) *The utilization of convenience sampling*

TABLE VI. RESPONDETS BACKGROUND

Number of Respondents in Work Areas	Software Development	19	21.3%
	Software Testing	55	61.8%
	Requirements Mgmt	1	1.1%
	Project Mgmt	4	4.5%
	Not specified, NA	10	11.2%
Number of Respondents in Roles	Individual Contributor	37	41.6%
	Specialist	25	28.1%
	Lead	16	18.0%
	Executive	6	6.7%
	Not specified, NA	5	5.6%
Experience in Years	Max (Min)	45	(0)
	Avg (Median)	12.9	(11.0)
Years in Current Role	Max (Min)	24.0	(0)
	Avg (Median)	3.8	(3.0)
Basis of Evaluations	Experience	1979	93.0%
	Opinion	149	7.0%

TABLE VII. TOOLS-SURVEY EVALUATIONS, # OF OUTLIERS & RANK OF CRITERIA

Rank	Criteria	Appium		Jenkins		Jira		JMeter		Robot Fw[a]		Selenium	
		Eval.[b]	O.[c]	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.	Eval.	O.
6	Applicability	66.0^{6}	0	84.0^{5}	1	83.8^{5}	0	79.0^{6}	0	83.0^{4}	5	62.0^{7}	0
8	Compatibility	52.0^{9}	0	84.5^{4}	0	77.5^{6}	0	71.7^{12}	0	81.8^{5}	4	56.0^{9}	0
7	Configurability	70.0^{5}	0	81.0^{8}	0	77.5^{6}	0	82.0^{5}	1	80.0^{7}	2	68.3^{5}	0
4	Cost-Effectiveness	61.3^{7}	0	86.5^{2}	0	61.3^{13}	0	83.8^{3}	0	89.4^{2}	5	76.3^{3}	0
1	Costs	84.0^{1}	0	86.0^{3}	1	47.5^{15}	0	93.0^{1}	0	92.5^{1}	8	72.5^{4}	0
5	Cross-Platform Support	77.5^{3}	0	83.5^{6}	1	85.0^{4}	0	73.0^{10}	0	83.7^{3}	1	66.3^{6}	0
9	Easy To Deploy	39.0^{13}	0	70.5^{12}	0	57.5^{14}	0	84.0^{2}	0	79.4^{9}	8	51.0^{12}	1
9	Easy To Use	45.0^{11}	0	60.0^{15}	0	73.8^{9}	0	77.5^{9}	0	78.9^{10}	6	56.3^{8}	0
13	Expandability	46.3^{10}	0	78.5^{9}	0	73.8^{9}	0	59.0^{14}	1	78.4^{11}	2	38.8^{15}	0
3	Further Development	78.8^{2}	0	82.0^{7}	0	86.3^{3}	0	78.8^{8}	0	80.1^{6}	2	82.5^{1}	0
12	Maintenance of TC&D	53.8^{8}	0	72.0^{11}	0	65.0^{11}	0	72.0^{11}	0	76.8^{13}	3	53.8^{11}	0
11	Performance	36.0^{14}	0	77.0^{10}	1	65.0^{11}	0	79.0^{6}	0	78.2^{12}	3	55.0^{10}	0
1	Popularity	72.0^{4}	0	89.0^{1}	0	87.5^{1}	0	83.3^{4}	0	70.4^{14}	1	82.5^{1}	0
15	Programming Skills	27.5^{15}	0	63.5^{13}	0	87.5^{1}	0	50.0^{15}	0	62.4^{15}	5	43.8^{13}	0
13	Reporting Features	42.5^{12}	0	63.0^{14}	0	75.0^{8}	0	66.0^{13}	0	79.8^{8}	0	43.8^{13}	0
	Total # of outliers		0		4		0		2		55		1

[a] Robot Framework
[b] Arithmetic mean of the evaluations for a criterion. The superscript is the ranking of the criterion.
[c] Number of outlier values in the data for a criterion

TABLE VIII. TOOLS-KRIPPENDORFF'S ALPHA & COEFFICIENT OF VARIATION

		Appium		Jenkins		Jira		Jmeter		Robot Fw[a]		Selenium	
		Ordin.[b]	Ratio[c]	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio	Ordin.	Ratio
	Krippendorff's α	0.294	0.208	0.173	0.113	0.224	0.069	-0.07	0.076	0.127	0.086	0.15	0.044
Coefficients of Variation	Applicability	0.13	0.18	0.11	0.13	0.06	0.08	0.20	0.23	0.18	0.20	0.26	0.36
	Compatibility	0.20	0.34	0.11	0.12	0.12	0.16	0.35	0.46	0.18	0.19	0.14	0.24
	Configurability	0.13	0.18	0.12	0.13	0.04	0.06	0.18	0.20	0.20	0.21	0.17	0.30
	Cost-Effectiveness	0.08	0.14	0.13	0.14	0.20	0.32	0.10	0.13	0.15	0.15	0.10	0.15
	Costs	0.23	0.26	0.20	0.25	0.33	0.60	0.10	0.12	0.13	0.17	0.33	0.42
	Cross-Platform S.	0.23	0.27	0.19	0.22	0.06	0.08	0.31	0.37	0.14	0.14	0.30	0.41
	Easy To Deploy	0.24	0.55	0.21	0.26	0.35	0.57	0.17	0.19	0.22	0.24	0.18	0.32
	Easy To Use	0.22	0.43	0.27	0.38	0.17	0.26	0.24	0.30	0.19	0.22	0.13	0.23
	Expandability	0.28	0.58	0.17	0.19	0.13	0.18	0.29	0.44	0.22	0.23	0.25	0.62
	Further Devel.	0.22	0.27	0.12	0.14	0.07	0.09	0.25	0.30	0.21	0.22	0.15	0.18
	Maintenenance	0.20	0.35	0.20	0.25	0.26	0.42	0.25	0.31	0.22	0.24	0.17	0.32
	Performance	0.27	0.65	0.18	0.21	0.19	0.31	0.19	0.22	0.20	0.22	0.18	0.31
	Popularity	0.25	0.31	0.09	0.09	0.06	0.07	0.17	0.21	0.27	0.30	0.11	0.14
	Programming Skills	0.35	1.16	0.34	0.43	0.06	0.07	0.43	0.71	0.35	0.41	0.30	0.68
	Reporting Features	0.28	0.59	0.25	0.32	0.19	0.27	0.40	0.50	0.21	0.23	0.38	0.75
	Pearson's Corr.	$r(13) = 0.85$		$r(13) = 0.99$		$r(13) = 0.99$		$r(13) = 0.96$		$r(13) = 0.98$		$r(13) = 0.87$	
	P-value	$6.497e-05$		$3.33e-12$		$9.534e-14$		$6.358e-09$		$8.522e-11$		$2.314e-05$	

[a] Robot Framework
[b] *Ordinal* level of measurement, see calculation for CV in Section 3.2.4.
[c] *Ratio* level of measurement, see calculation for CV in Section 3.2.4.

TABLE IX. INTRACLASS CORRELATION COEFFICIENT

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.34	3.9	14	56	0.00015	0.13	0.62
ICC3	0.36	3.9	14	56	0.00015	0.14	0.65
ICC2k	0.72	3.9	14	56	0.00015	0.43	0.89
ICC3k	0.74	3.9	14	56	0.00015	0.45	0.90

Number of criteria = 15 Number of respondents = 5

(a) Appium

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.21	5.8	14	126	1.2e-08	0.083	0.44
ICC3	0.33	5.8	14	126	1.2e-08	0.163	0.58
ICC2k	0.72	5.8	14	126	1.2e-08	0.475	0.89
ICC3k	0.83	5.8	14	126	1.2e-08	0.661	0.93

Number of criteria = 15 & Number of respondents = 10

(b) Jenkins

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.23	3.0	14	42	0.0031	0.034	0.53
ICC3	0.33	3.0	14	42	0.0031	0.082	0.63
ICC2k	0.55	3.0	14	42	0.0031	0.123	0.82
ICC3k	0.66	3.0	14	42	0.0031	0.263	0.87

Number of criteria = 15 & Number of respondents = 4

(c) Jira

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.136	3.2	14	56	0.00097	0.012	0.37
ICC3	0.306	3.2	14	56	0.00097	0.094	0.60
ICC2k	0.440	3.2	14	56	0.00097	0.058	0.75
ICC3k	0.688	3.2	14	56	0.00097	0.341	0.88

Number of criteria = 15 & Number of respondents = 5

(d) JMeter

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.12	16	14	1064	1.6e-35	0.064	0.26
ICC3	0.16	16	14	1064	1.6e-35	0.088	0.33
ICC2k	0.91	16	14	1064	1.6e-35	0.840	0.96
ICC3k	0.94	16	14	1064	1.6e-35	0.881	0.97

Number of criteria = 15 & Number of respondents = 77

(e) Robot Framework

type	ICC	F	df1	df2	p	lower	upper
ICC2	0.20	2.5	14	56	0.0073	0.0300	0.48
ICC3	0.23	2.5	14	56	0.0073	0.0375	0.53
ICC2k	0.55	2.5	14	56	0.0073	0.1340	0.82
ICC3k	0.60	2.5	14	56	0.0073	0.1632	0.85

Number of criteria = 15 & Number of respondents = 5

(f) Selenium

VI. TEST AUTOMATION

Test automation framework is a combination of guidelines, practices, concepts, coding standards, report mechanism, etc.,
A tester should follow these guidelines to take advantage of various productive results.
There are different frameworks available in the market, the widely used ones are discussed. Each one of these has individual characteristics and features.

A. Keyword Driven Framework

In this, testers create various keywords and associate different functions with each one of them. Function library has the logic to read these keywords and call and perform the associated actions.

The driver script reads the scenario and performs test execution. It is used in situations where testers who create test scripts does not have programming expertise whereas framework creation is done by automation experts.

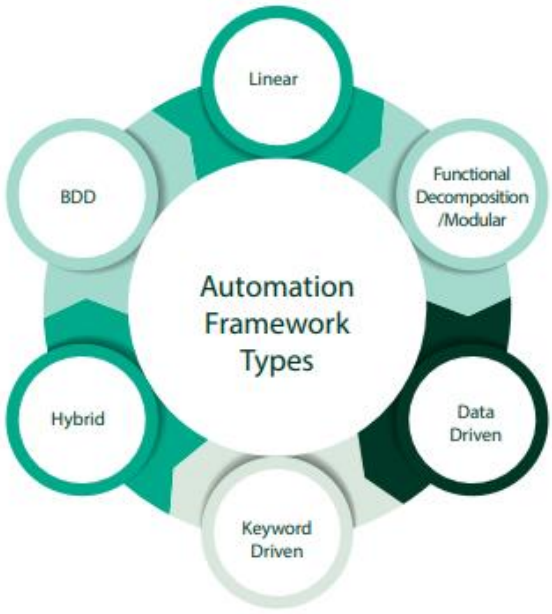
B. Hybrid Framework

This is created by combining different features of two or more frameworks. This enhances the strength of different frameworks and minimizes their weakness.
It is highly robust, flexible, and maintainable. However, this requires strong technical expertise to design and maintain.

C. Behaviour driven Development

It automates validations in an easily readable and understandable format to business analysts, developers, testers etc. Such framework does not necessarily require user to be acquainted with programming language. There are different tools available for BDD like cucumber, Jbehave and many more which works along with other test automation tools. This is more suitable for applications using agile methodology and where early automation is required. It focuses on the behavior of the system rather than implementation aspect.

The traceability between requirements and scripts is maintained throughout and test scripts are easy to understand for the business users.



D. Pillars of the right framework for the digital Era

Automation can improve quality and lead to higher testing efficiency. Therefore, it is important to plan it well and make the right choice of tools and frameworks. When test automation uses the right framework based on the context, it yields great benefits. Hence, it is worth understanding the key requirements of the framework, before choosing the right one.

Some of the key aspects of automation framework to look for digital assurance journey are as follows:

E. Extreme automation

Digital transformation programs, big data, cloud and mobility are changing the way testing is being done. Extreme automation is the key and automating every part of the testing process instead of just regression is crucial now. A framework which is more scalable and facilitates lifecycle automation as well as broader test coverage is needed for digital assurance programs.

F. Technology and tool agnostic approach:

There are many tools and frameworks which poses a lot of integration challenges; hence it is imperative to choose a framework which is technology and tool agnostic and support various tools and technologies.

G. Script less capabilities

Automate the automation and need to look for script less automation avenues. Most software testers and users find it challenging to learn programming languages like Java, Visual Basic to write the scripts that the test automation demands.

There are frameworks and accelerators available with user friendly GUI which helps to create automation scripts in a much easier way than having to know and write code in any specific programming language. Choosing a framework which helps to create a test script from the recorded script or based on the input from a spreadsheet will help in accelerating automation and reduce dependency on skilled resources.

H. True Shift Left Attitude

Gone are the days when automation team would wait until the application is built and start automation activities thereafter. The need o the hour is to shift and start automation during the requirement gathering phase of the systems development life cycle(SDLC) itself. Automation framework with exhaustive reusable library and support for BDD will help both business users and QA team to start automation activities early.

I. Omnichannel, mobility and Cloud features

Organizations are focusing more on digital assurance, but it is important to test real user behavior and to test on multiple devices like mobiles, tablets etc., Hence the chosen test automation needs to facilitate testing on multiple devices to ensure the uniform experience on all devices. If the framework supports the reuse of the scripts used for online or desktop, it will help in saving much effort.

When addressing the multifaceted needs of mobile testing conducting comprehensive testing across hundreds o different devices, brands, models, and operating system combinations is tedious. A framework that facilitates integration with cloud infrastructure is an added advantage.

J. Zero Touch Automation

As DevOps is slowly taking over IT, it is important to reduce the distance development and deployment. Test scripts needs to be executed in an unattended manner without requiring much manual intervention. Remote execution, parallel execution, zero touch execution and execution from continuous integration tools like Jenkins and Hudson when supported by automation framework, will help a lot in managing multiple sprints and shorter cycles better.

K. Seamless Integration

With lot of tools being used in application development and testing, it is important that automation and chosen framework facilitates integration with various tools. Hence, it is important to integrate with management tools, defect tracking tools, build tools analytics tools and continuous integration tools in landscape.

L. User friendly Reporting

Agile and DevOps has brought the business, development and QA teams to work together. The ability to run a high volume of tests is of little use if the results of the tests are not easy to understand by stakeholders.

The framework has to facilitate automatic generation of reports of the tests execution and show the results in easy-to-read format. Although most of the tools in the market give few reporting options, they are not self-explanatory and adequate. Hence the framework with good reporting capabilities such as HTML reports, live execution, and screen shots in case of failures and video reporting of the execution options will be very helpful. Automation framework facilitating detailed test result reporting reduces the overall effort to a greater extent.

Author, Indumathi Devi, concluded that no one size fits all. Since every project is unique, the challenges, duration and tools choices may vary. Organizations seeking agility in their business process need to undertake robust test automation solutions that ensure better software quality. Successful test automation frameworks for digital assurance are the ones that support extreme automation, omnichannel testing, zero touch execution of test scripts and have some or all the key aspects described above. Author recommended that organizations select an automation framework that can lead to smarter automation, better overall results, productivity benefits and cost efficiencies in the highly dynamic digital landscape.

VII. RECOMMENDATION

Quality is the main focus for any project in software, so it is recommended to consider size and cost budgeting in selecting any tool for testing and also the platform where project should be used should reflect in the criteria for choosing a tool. From the results obtained in the paper, authors recommend that TestComplete and Ranorex testing tools for testing across all platforms and also since they are licensed tools the budget should be taken into consideration. Appium is recommended for mobile testing, while selenium is for web testing with an advantage that it is an open source.

VIII. THREATS TO VALIDITY

For evaluating the validity of peer reviews, regarding internal validity, authors acknowledge the bias of the sampling techniques for the surveys. Small sample size of 89 is the threat to the external validity.

Tool evaluations are construct and context specific, bound to time and experiences, the results are not generalized. However, the results provide a basis for analyzing possible problematic perceptions. The questionnaire can be refined based on the results for further studies.

XI. CONCLUSIONS

Software testing guarantees that the software been deployed to the market is free from error and effects. To meet up with market demands and time factor, software testers has employed the use of automated testing tool to carry out testing, this is used over the manual approach to testing, as automated testing saves time and also minimize cost incurred in the organization during the testing phase.

This paper has evaluations of some automated testing tools which includes selenium, TestComplete, Ranorex, Open Script, Janova, etc., highlighting their basic features and characteristics. All testing tools are efficient to be used for testing but depending on the scenario, some may tend to be more efficient than others.

Tool evaluations are construct and context specific and bound to time and experiences. Thus, opinions on software testing tools can be diverging or conflicting. Recollection of personal experiences is error-prone, but beliefs should be given attention in research to help to provide and to disseminate verified evidence to the practitioners. Sometimes, these can be misleading so perceptions and beliefs of practitioners should be analyzed with caution.

According to author findings, experience-based evaluations seem to be more positive than those based on pure opinion and expert respondents tend to provide consistent evaluations for some criteria.

Findings suggest that more than just three expert respondents are required to gain reliable evidence for testing tool evaluations. On an average, opinions from seven experts can provide reliable evidence for moderate level of accuracy. There is a need for practical and efficient ways for conducting tool evaluations that provide reliable empirical evidence for software practitioners.

REFERENCES

[1] Kaur, H. & Gupta, G. (2013). Comparative Study of Automated Testing Tools: Selenium, Quick Test Professional and Testcomplete, Int. Journal of Engineering Research and Applications, 3(5), 1739–1743.

[2] Raulamo-Jurvanen, P., Mäntylä, M. & Garousi, V. (2017). Choosing the Right Test Automation Tool, in 21st International Conference on Evaluation and Assessment in Software Engineering (EASE 2017), 21–30.

[3] Raulamo, P., Hosio, S. & Mäntylä, M. V (2019). Practitioner Evaluations on Software Testing Tools. In Proceedings of the Evaluation and Assessment on Software Engineering, 57–66.

[4] Top 10 Testing Automation Tools for Software Testing, pp. 1–4.

[5] Babar, M. I., Jehanzeb, M., Khan, M., & Iqbal, S. (2017). A Comparative Analysis of Quality Assurance of Mobile Applications using Automated Testing Tools, International Journal of Advanced Computer Science and Applications, 8(7), 249-255

[6] Bajaj Harsh, Infosys. 2015. Choosing the right automation tool. https://www.infosys.com/it-services/validation-solutions/white-papers/documents/choosing-right-automation-tool.pdf.