Why Paid tools are better than Selenium in Agile environment

Prashant Xavier Lopes

Abstract— Many organizations prefer Selenium webdriver over any Paid tool. In this paper I tried to showcase why any paid tool is better than Selenium.

Index Terms— Automation, Automation research, Automation tool, Automation tool research, Best Automation tool, Comparison between different automation tool, Conclude which automation tool, Research on automation tool, Shortlisting automation tool,Tolos better than Selenium, Tools other than Selenium, Which automation tool is better, Why to choose Selenium, Why to go for Paid tools

—————————— ◆ ——————————

1 BACKGROUND

In software industry, testing a software is as important as developing it. And as new features are added to the product, it is equally important that our old features are working as expected.

This testing of existing features (or doing regression testing) consumes huge amount of cost, time & resources. So the best way to save all these is to have a good automation tool in place.

Most of the organizations prefer Selenium over any other tool for automation as this saves a good amount of cost in terms of licensing fees.

In this paper, I will try to showcase why paid tools can be considered over Selenium, especially in Agile environment.

2 INTRODUCTION

Suggesting any paid tool over Selenium can be tough as there can be many reasons, but few of the common reasons could be –

- ➢ Selenium saves a lot of tool cost. Why to invest and increase cost of project?
- ➢ Resources are rigid and want to use Selenium, as they are familiar with it and also it opens doors for them in the market.

Considering above reasons, in an Agile organization we will try to discuss few points showcasing some comparison between Selenium and paid tools.

3 WITH RESPECT TO AGILE

When we talk of Agile, for our purpose let's focus on two points –

- ➢ *In Agile, both experience & experiments goes hand-in-hand.*

Like in any Agile project, if individuals suggest different approaches to solve a problem, one based on experience and one based on experiment, then we ideally do a small pilot and check which approach is better.

Here too, if individuals prefer Selenium over paid tool, then the best approach would be to do a small pilot with both the tools and check the results based on efficiency and delivered values.

- ➢ *Agile focuses on VALUE.*

Agile focuses on retrieving value as soon as possible and anything that does not give value should not be done.

The term VALUE can be both Return on Investments (ROI) or Value on Investments (VOI) or any one of them.

Now, since organizations use Selenium Webdriver; scripting and debugging of scripts consumes huge amount of time, as scripting in Selenium webdriver can be considered as creating a parallel system (as coding level is very high) .

But with any paid tool, scripting and debugging is much faster and also most of the paid tools provide many other features better than Selenium.

We can easily say that paid tools are having much more VALUE than Selenium.

In below points we will add more value points to each category and then conclude which is better.

4 WITH RESPECT TO BUSINESS

Focus of any sponsor is to get maximum returns (ROI or VOI) on their investments. Or in simple words we can say that they want maximum output by investing only the required amount of time and money.

Paid tools may ask for investment but will provide huge returns in very less time as –

- ➢ Faster scripts creation as compared to Selenium
- ➢ Faster debugging as most tools are based on Test case/steps creation using record & playback or keyword driven rather than writing complete lengthy codes
- ➢ Cost of resources & licenses will reduce overtime as per project scope; increasing the profitability

5 COST FACTOR

It is true that any paid tool will increase the cost of any project but let's check a project management concept which says

Time is inversely proportional to Cost

Time=K*Cost (K is any constant)

Or in simple terms we can say that we have to invest money to do the things quickly.

Now like we discussed in above section, that paid tools are much faster than Selenium, we can say that to save time and to deliver things quickly, we should invest in paid tools rather than Selenium.

People could say that we can get faster deliveries by increasing resources rather than getting a paid tool; but we all know that reducing number of licenses is much easier and better approach than removing resources or consuming them in their non-expertise areas.

Also, we should remember that "Cost" factor in any project is a major thing but not everything.

6   TIME FACTOR

Since we talked a bit that with paid tools, scripting and debugging is much faster; which means they save lots of time.

Let's try to prove this using some figures.

I tried to compile the figures (authoring rate) from my past projects experience and are based on mixture of high to low complexity level of cases in any suite.

**The results are derived using 3-point estimation technique.*

Selenium

Total cases	100

	Authoring rate/day by single resource
Authoring/ Scripting rate when everything is perfect (A)	5
Authoring/ Scripting rate when everything is not perfect (C)	1
Authoring/ Scripting rate when things are ok but not perfect (M)	3

A	C	M
20	100	33.33

Estimate (E)=(A+4M+C)/6	SD= (C-A)/6
42.22	13.33

Paid Tools

Total cases	100

	Authoring rate/day by single resource
Authoring/ Scripting rate when everything is perfect (A)	8
Authoring/ Scripting rate when everything is not perfect (C)	3
Authoring/ Scripting rate when things are ok but not perfect (M)	5

A	C	M
12.5	33.33	20

Estimate (E)=(A+4M+C)/6	SD= (C-A)/6
20.97	3.47

Here,

A = Total cases/ Authoring/ Scripting rate when everything is perfect

C = Total cases/ Authoring/ Scripting rate when everything is not perfect

M = Total cases/ Authoring/ Scripting rate when things are ok but not perfect

E (Estimate) = (A+4M+C)/6

SD (Standard deviation) = (C-A)/6

**Please read 3-point estimation technique in detail to understand.

From above claculations, we can easily say that estimated time to script same amount of cases by a single resource using any paid tool is much less when compared to Selenium webdriver.

7   PROS/CONS

Let's see few major comparisons between Selenium and paid tools

Selenium	*Paid tools*
Cost effective	*License cost is involved*
Complex setup/installation	*Easy installation*
Scripting & Debugging consumes lot of time	*Scripting & Debugging consumes less amount of time as scripts are created mostly using record/ playback or using keywords*
Support Webapps only. For mobile & desktop, needs to get integrated to other tools.	*Most of the tools supportWeb (UI & APIs), Mobile, Desktop and some support packaged apps too.*
Selenium is less reliable as it is a free tool and can be shut down anyday (Selenium IDE is already obsolete)	*Paid tools cannot be shut down any day due to legal compliances and formal agreements. Much more reliable for long term.*
Not much additional features	*Many additional features like test scheduler, test management, business process testing etc…*
Complex repository management	*Easy repository management*
Only community support	*Both Community/Priority Support*
Linking to multiple Element properties means more coding	*Element properties explorer is well developed*
Advanced programming skills needed to integrate various tools	*Programming skills not required. Recommended for advanced test scripts.*
Conceive lots of learning time (almost 2-3 months), as resources needs to get familiar with programming too	*Learning curve is short (2-3 weeks) as resources are not required to learn programing*

Resources easily available in market	*Not much peoples may be available for the selected paid tool*

8 CONCLUSION

With above discussed points we can conclude that Selenium is only efficient with cost & resource availability but not in features, productivity and time saving.

So, I would suggest we should go for a good paid tool since we can manage cost to not exceed the budget and resources can be trained to use any new tool.

REFERENCES

[1]   Part of the content present in background is taken from another whitepaper written by me, having ID as - I0127138 submitted to IJSER.