International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Transforming Quality Assurance: The Impact of Automation on Product Development

Shreekant Majge

Malvern PA, USA
Email: *smajge[at]gmail.com*

Abstract: ***Organizations commonly believe that adopting test automation tools and methodologies leads to cost savings and enhanced quality assurance. In today's dynamic business environment, companies are compelled to adopt faster and more efficient testing methodologies to ensure a better return on investment. Test automation is a crucial step towards achieving a mature level of quality assurance. By implementing a well-planned test automation process, productivity can be increased without compromising quality. However, there are several challenges associated with using existing commercial off-the-shelf (COTS) automation tools available in the market. This study examines the challenges associated with test automation, addresses these challenges with effective strategies, and presents research results aimed at revolutionizing QA and transforming products through automation.***

Keywords: Quality Assurance, Software product development, Automation, Test Automation Framework, COTS

1. Introduction

Software Testing is an integral, costly and time-consuming activity in the SDLC. The challenge of the day for IT companies is to

- Improve productivity without compromising on Quality,
- Reduce the cost of Testing
- Providing the right approach to deliver real value to the Business.

Changes in the environment like new patches, service packs etc. also require testing. Maintenance testing is a burden from typical changes in the application. Manual testing alone can't keep pace with the rapid development of applications. It is very difficult to test the complete application for the frequent releases in the SDLC.

2. Problem to be addressed

There are several problems in using existing Commercial off the shelf Automation tools that are available in the market. Following are some problems we face,

- Support for third party controls that are written to increase the functionality of applications are not available to any practical extent in the existing Automation tools, Automation of such controls requires a separate mechanism to identify the components called as Proxies.
- Cost of License is High
- We often hear something like "It looked so easy when the tool vendor (salesperson) showed the demo, but my people couldn't get it to work.", or "We spent 6 months trying to implement this tool effectively, but we still have to do most of our testing manually.", or, "It takes too long to get everything working properly. It takes less time just to manually test." these generally results as the tool ends up on the shelf as just another "purchasing mistake".

3. Test Automation Life cycle

Test Automation plays a pivotal role in effective testing and reduction in test effort. Test Automation is the process of automating the steps of manual test cases using an automation tool or utility to shorten the testing life cycle with respect to time.

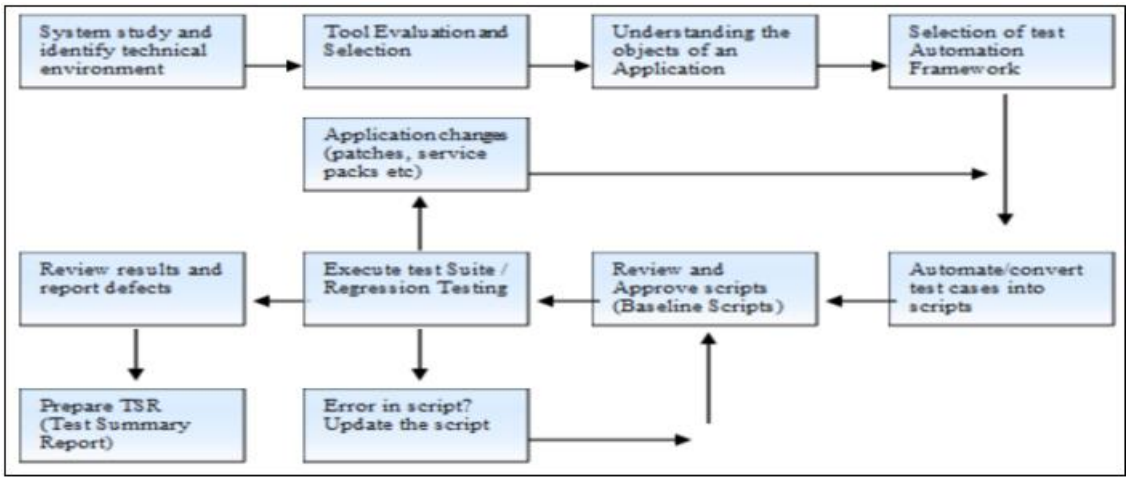

Figure 1: Test Automation Life Cycle

By using the systematic approach outlined within the Test Automation Life cycle (Fig.1), organizations can arrange and execute test activities in a way that maximizes test coverage within the limits of testing resources. This structured test methodology involves a multi-stage process that supports the detailed and inter-related activities that are required to introduce and utilize an automated test tool. The methodology supports the development of test design, the development and execution of test cases and the development and management of test data and the test environment. It also supports documentation and tracking allowing the test team to obtain closure on issue/trouble reports.

Automated functional testing using automated tools to execute based on a script format. Basically, this means capturing the user interactions on an application such as field entries. This gets captured into a script language which can be played back later duplicating the user interactions. It's a three-step process.

Automation-A Three Step Process

- **Record actions into a script**
  Tool supported language - either VB or Java
- **Optionally enhance scripts**
  Custom coding – Update recorded scripts to remove the human intervention and parametrize the data
- **Execute scripts**
  Ideally run scripts overnight to increase test productivity to 24/7

4. Test Automation - Objects and Controls

Applications, such as Windows can have Objects as well as Text that may need to be verified. Although objects may contain text, they also contain other behaviors that can impact the application such as clicking on buttons. Objects also have States; many of us have seen this as Buttons, Menu items etc (Fig. 2).

What are Objects?

Objects are components that make up a Windows Application; these are typically referred to as Windows and Controls. Controls are found in Windows.

What are Controls?

Some familiar Controls (Fig. 2):

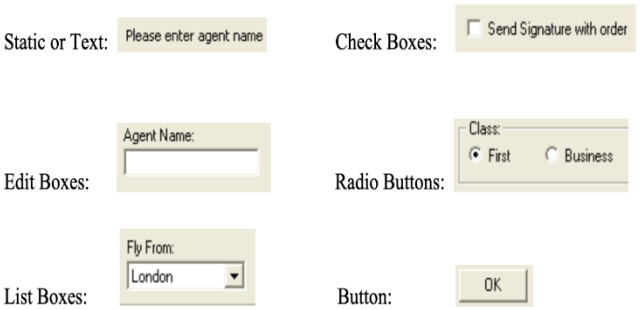

Figure 2: Standard controls used in application development

Some Controls may not be so familiar (Fig. 3):

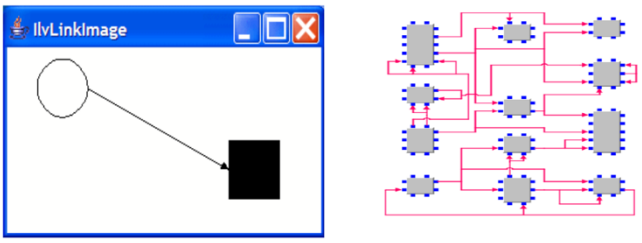

Figure 3: Electrical signal diagram and connector nodes

What are Custom Controls?

They are sometimes called Third Party Controls. They are specially programmed controls that can have very unique features that are written to increase the functionality of applications. However, they also may only be recognized as a single window with bit-map graphic images that change based on mouse click or keyboard behaviors.

Several successful approaches were there to give a complete solution for the customized components as well as other normal components too. By using Rational functional tester, the following approaches can be achieved.

Fixing the coordinates

The basic functionality like clicking can be achieved by predetermining the coordinate location of the component. The dragging and double clicking of the component also could be done the same way [2]. This approach is cost effective and requires less effort

E.g.

```
ilogViewsIlvManagerView().click(atPoint(596,508));

ilogViewsIlvManagerView().drag(atPoint(595,509), atPoint(624,512));
```

Drawbacks: Some of the test cases couldn't be automated properly by this approach. Also, the location of the component changes if the screen resolution changes.

OCR technique

A small image of the component is stored in a repository, and it is compared with the existing window image, thereby finding the location. By this approach we could also find the location of all the similar kinds of component in the application window. We can use the basic methods like clicking, double clicking, dragging etc. and we can also detect the change in color of the component or change in size. The approach is cost effective, and it can be achieved by writing a few functions or using any OCR tools.

Drawbacks: Although by this approach we can automate many test cases but the execution time for automation will increase as bitmap comparison is compared pixel by pixel. Also, the repository was an overhead component for this approach. The approach fails if the screen resolution is changed.

Development of proxies

Rational Functional Tester needs UI control-specific information to perform functional testing operations such as recording, playback, verification points, and data driving. It tries to map the closest proxy if it finds a new control for which it has no proxy.

The Rational Functional Tester architecture enables developers to write a proxy for a particular UI control. Developers can enable Rational Functional Tester to process the specifics of a control by writing proxies.

The development of proxies for the custom controls is the best possible approach in the long run. It gives a most robust and reliable solution. The following Test Domain Implementation classes are available with Rational Functional Tester (Fig. 4).

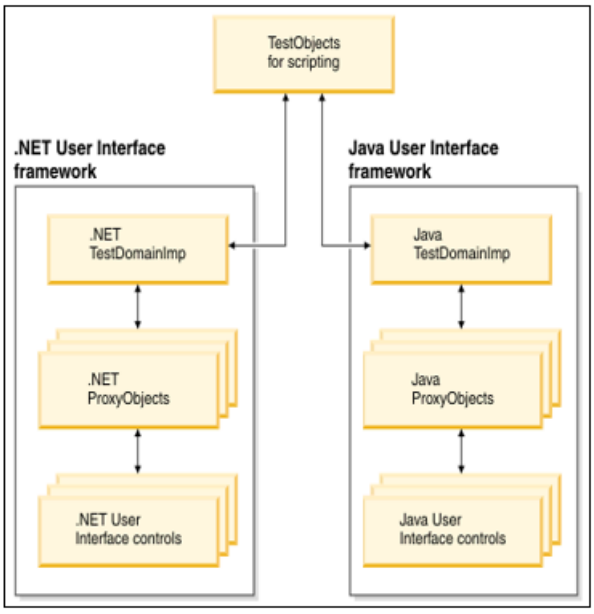

Figure 4: Test Domain Implementation classes of RFT.

5. Test Automation – Common challenges

Despite the clear benefits of test automation, organizations are not able to build effective test automation scripts. Test automation becomes a costly effort that finds few bugs and is of questionable value to the organization. Automated test scripts can become outdated quickly as the application evolves, leading to high maintenance costs. Automating tests for applications with dynamic content (e.g., frequently changing UI elements) can be tricky.

There are several reasons why test automation efforts are unproductive. Some of the most common include:

Don't have everyone up to speed

The Test Analyst and the Business users working on the test automation are not up to the speed for converting the test cases into test scripts due to the lack of knowledge in scripting and Automation tools.

Way more expensive than you first thought

The cost of securing a license is very high, The Automation team using the license for different purposes like Scripting, Debugging, enhancement of scripts, Test plan building etc. The ROI on the test Automation is really a challenging job without proper Automation strategy.

Table 1: QA Team resource mix without automation framework

	Analysts	Automation Engineers	Number of Automation Tool Licenses
Team 1	2	4	6
Team 2	3	3	6
Team 3	4	2	6

A Team 1 having 2 Test Analyst and 4 Automation Engineers require 6 licenses for each one of them to work on Automation like scripting, debugging etc., A Team 2 having 3 Test Analyst and 3 Automation Engineers require 6 licenses and Team 3 having 4 Test Analyst and 2 Automation Engineers also require 6 licenses. The team for all the combination, you require an equal number of licenses.

6. Test Automation – Frameworks and standards to address challenges.

A test automation framework is a set of guidelines, standards, tools, and practices that provide support for automated software testing. There are different types of test automation frameworks [1] like Linear, modular, Data driven, hybrid, BDD and keyword driven, and these has its own with pros and cons. In this article we explore the keyword driven framework which will address some of the challenges mentioned in section 5.

6.1. Key-Word driven framework

Sometimes it is difficult to address all challenges with the help of Test Automation tool which will support the functionality and test environment we need. This requires developing customized test framework to address the requirements. A Test Automation Framework is a set of assumptions, concepts, and practices that provide support for automated software testing (Fig. 5).

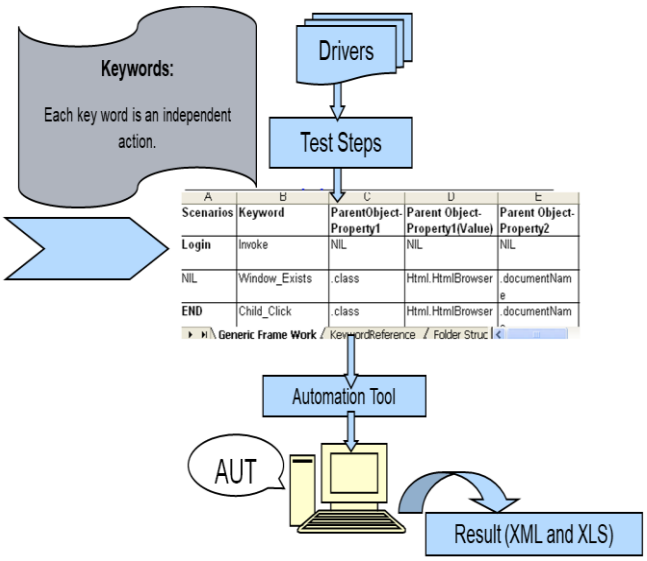

Figure 5: Architecture of the key-Word driven framework

Key-Word driven method uses the actual Test Case document developed by the tester using a spreadsheet containing special "Key-Words". This framework aids in Automating the manual test cases using an excel sheet and it also Replaces the scripting by intelligent framework "engine" (Fig. 6). It has the inbuilt generic functions/ Key-Words which in turn help in converting the manual test cases into the automation test scripts

	A	B	C	D	E	F	G	H
1	**Scenarios**	**Keyword**	**ParentObject-Property1**	**Parent Object-Property1(Value)**	**Parent Object-Property2**	**Parent Object-Property2(Value)**	**Child Object Property1**	**Child Object-Property2(Value)**
2	**Login**	Invoke	NIL	NIL	NIL	NIL	NIL	NIL
3	NIL	Window_Exists	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	NIL	NIL
4	NIL	Window_Min	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	NIL	NIL
5	NIL	Window_Max	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	NIL	NIL
6	NIL	Window_Activate	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	NIL	NIL
7	NIL	Edit_Set	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	.class	Html.INPUT.text
8	NIL	Edit_Set	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	.class	Html.INPUT.password
9	NIL	Child_Click	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/login.do	.class	Html.INPUT.submit
10	**END**	Child_Click	.class	Html.HtmlBrowser	.documentName	http://127.0.0.1/tasks/otasklist.do	.class	Html.IMG

Figure 6: Test case for Automation in Excel

Each of the "Key Words" in Column B causes a "Utility Script" to be called which processes the remaining columns as input parameters to perform specific functions.

The main phase of testing is the reporting part, which will be taken care of by the framework by generating the reports in both xls and xml (Fig. 7).

Figure 7: Automated Test run result

The keyword driven framework addresses the challenges mentioned above with the ease of scripting by using an excel engine and it addresses the usage of tool licenses by using Key-Words which are readily available. During test case conversion we don't require the Automation Tool

The keyword driven framework addresses the challenges mentioned above with the ease of scripting by using an excel engine and it addresses the usage of tool licenses by

Table 2: QA Team resource mix with automation framework

	Analysts (Framework)	Automation Engineers (Automation Tool)	Number of Automation Tool Licenses
Team 1	2	4	4
Team 2	3	3	3
Team 3	4	2	2

A Team 1 having 2 Test Analyst and 4 Automation Engineers require 4 licenses of Automation tool for each one of them to work on Automation like scripting, debugging etc., A Team 2 having 3 Test Analyst and 3 Automation Engineers require 3 licenses and Team 3 having 4 Test Analyst and 2 Automation Engineers also require 2 licenses. The team with a good combination like team 3 is in place then you can save the licenses of Automation Tools. Hence the ROI on Test Automation can be achieved soon in the project.

7. Conclusion

This study and research cover automation testing and its revolution in QA to transform software products. These strategies can be implemented with the proper Automation approach to Improve productivity without compromising on Quality, Reduce the cost of Testing and by following best practices, organizations can successfully implement automation testing and significantly enhance the quality and reliability of their software products. In addition, this article explains the challenges in test Automation that can be resolved by using different approaches. Also, the cost-effective test Automation solution using Keyword driven framework which also eases the scripting by using excel sheets. Automation testing, while highly beneficial, comes with its own set of challenges that require careful planning and execution to overcome. However, there are many tools and approaches available to address each situation which can be explored to address various challenges.

Conflicts of Interest

The author declares that there is no conflict of interest regarding the publication of this paper

References

[1] E. H. Kim, J. C. Na and S. M. Ryoo, "Implementing an Effective Test Automation Framework," 2009 33rd Annual IEEE International Computer Software and Applications Conference, Seattle, WA, USA, 2009, pp. 534-538, doi: 10.1109/COMPSAC.2009.188. keywords: {Automatic testing;Automation;Software testing;Automatic control;System testing;Application software;Programming profession;Computer languages;Graphical user interfaces;Control systems;Test Automation Framework;Automated Testing;Continuous Integration;Fit;FitNesse;STAF;STAX}

[2] The IBM Website. [Online]. Available: https://public.dhe.ibm.com/software/websphere/ilog/docs/visualization/jviews86/jviews-diagrammer86/ps_usrextdiag.pdf

[3] Shreekant Majge, 2024, Healthcare Interoperability through Rigorous HL7 Testing, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 13, Issue 01 (January 2024).