

# **AUTOMATION OF PURCHASE ORDER IN MICROSOFT DYNAMICS 365 BY DEPLOYING SELENIUM**

Vijay Biju and Shahid Ali

Department of Information Technology, AGI Institute,  
Auckland, New Zealand

## **ABSTRACT**

*Regression testing is very important for dynamic verification. It helps to simulate a suite of test cases periodically and after major changes in the design or its environment, to check that no new bugs were introduced. Evidences regarding benefit of implementing automation testing which includes saves of time and cost as it can re-run test scripts again and again and hence is much quicker than manual testing, providing more confidence in the quality of the product and increasing the ability to meet schedules and significantly reducing the effort that automation requires from testers are provided on the basis of survey of 115 software professionals. In addition to this, automated regression suite has an ability to explore the whole software every day without requiring much of manual effort. Also, bug identification is easier after the incorrect changes have been made. Genius is going through continuous development and requires testing again and again to check if new feature implementation have affected the existing functionality. In addition to this, Erudite is facing issue in validation of the Genius installation at client site since it requires availability of testers to check the critical functionality of the software manually. Erudite wants to create an automated regression suite for Genius which can be executed at client site for checking the functionality of the software. In addition to this, this suite will also help the testing team to validate if the new features which have been added to the existing software are affecting the existing system or not. Visual studio, Selenium Webdriver, Visual SVN and Trello are the tools which have been used to achieve the creation of automation regression suite. The current research will provide guidelines to the future researchers on how to create an automated regression suite for any web application using open source tools.*

## **KEYWORDS**

*Automation testing, Regression testing, Visual Studio, C#, Selenium Webdriver, Agile- Scrum*

## **1. INTRODUCTION**

Independent Advisory Services Limited (IAS Ltd) is a software company which was founded in 2003 which aims to be the most successful, creative and path breaking consulting agency in Australia and New Zealand. The aim of the company is to approach the clients with unique ideas to develop customized and different strategies. The main strategies offered by the company are Enterprise resource planning (ERP), digital strategies and operational business strategy. Demo data is the basic data set which is released for implementation of support and demonstration purposes. The current demo data set supports verticals like Retail, Distribution, Service Industries, Public Sector and Discrete & Process Manufacturing. Demo data supports around 40 languages around 16 countries. In Dynamics 365 Field Service, a purchase order (P.O.) is created to purchase the products to sell to a customer in a work order. Company must automate the

purchase order creation for a liquor vendor in Microsoft Dynamics 365 and develop the test cases against the requirement. For automating the process, we must select a tool which is an open source, stable and supports all the programming languages. While studying the New Zealand market we found that Selenium will be the ideal tool for automating the purchase order creation of the various features to use like plugins, reporting etc. After analysing the requirements, we have developed a hybrid framework where all types of concept like test driven framework, Keyword driven framework etc all comes together which is easy to use and can be used for the future sprints. We have used two kinds of reporting which is TestNG, Extent reports for sharing the project execution results to the Stakeholders. For attaining this goal company must adopt an agile methodology where all the team members and individuals effectively prioritize the features and works to attain the product quality in the entire development phase.

IAS Ltd is to automate the purchase order creation and make a payment to the vendor. For attaining this goal, we must develop the test cases for automating the purchase order in Dynamics 365. Test results for each test cases will be generated through TestNG and extent reports. This reporting helps to identify the passed and failed scenarios with a simple pie chart showing the progress of the project execution. Each test cases will be written in separate methods and main class will call the methods while execution. A hybrid framework has been developed where test data is read from an excel file and failed test cases is captured by a screenshot method. Automation framework will be constructed as page object model in Maven which helps to maintain the framework and changes can be easily tackled in this approach.

This research project report is organized as follow: Section 2 focuses on the literature review of various studies concentrating on automation regression. Section 3 is focused on the research methodology for this research project. Section 4 of this research is focused on project execution results. Discussion to results of this project are provided in section 5. Section 6 is dedicated towards the future work recommendations. Finally, in section 7 conclusion to the research is provided.

## **2. LITERATURE REVIEW**

In the past different researches had been conducted for designing of automating framework for websites as discussed below:

A study was conducted on Analysis and Design of Selenium WebDriver by [7]. In this study they use designed and implemented automation testing framework for testing web applications using selenium web driver. In this framework tester can easily write their test cases in less time. The developed framework is helpful for developer to analyse and maintain their code due to screen shot property of framework and helps to generate test report which helps in identifying the passed and failed test cases as well as maintaining the test suite.

Another study was conducted about the behaviour driven Test Automation Framework [19]. This study focuses on how the user layer, technical, business layers and data layer constraints can be implemented and maintained without costing and can support test automation of all the different layers of a three-tier architectural system which will improve the test coverage, quality and reliability of the software system.

A research on focuses on the automation testing tools was proposed by [12] to enhance design and execution activity, challenges faced by non-automation tester in executing automation scripts and executing automation scripts using TestNG and its disadvantages and the proposed web application that nullify the problems faced by manual testers which reduce time on set-up environment to execute test scripts and overcome consequences of execution in TestNG.

Similar research was done by [9] to perform automation testing using “Selenium WebDriver” with this data driven framework it separates data to code for the purpose of reusability.

Another research was done by [10] to repair the test suite when modification is done to the application when test cases are to be changed.

A research that focuses on how Selenium tool will fasten the execution process and reduce the cost of the test execution process by [11]. The paper gives the information about how to focus on frequent integration which is the main feature of agile.

Another research by [12] develop an architecture that can be used repeatedly through the common application of the agile software and test action planning for automating the acceptance test.

A research by [13] that focuses on developing a framework which aims at reducing implementation and maintenance costs of automation tests and focuses on creating tests which can be easily understand by testers or stakeholders without previous coding knowledge. When a test fails while locating a page element which is not there due to outdated test script, Smart Driver automatically analyses the element which is moved to another place in the page.

Another study by [14] focuses on the best testing tool for checking the functionalities, security and performance with minimum cost and the paper fails on how to successfully implement the tools.

A study by [15] focuses on a new automated testing framework for testing the web applications which improves the automating process. The proposed framework saves around three-fourth of the total effort involved in the automation process using old automation and 21% compared to using Selenium IDE.

A research by [16] focuses on exploring various types of software testing, techniques and tools to compare manual testing against automation testing.

A research by [17] that focuses on a model for automated agile testing, and an working framework developed on the testing of a Web application and results are evaluated using the agile testing model, and there is a comparison between waterfall and agile models.

Similar research by [18] that focuses on case study by surveys were used for data collection and challenges connected to the testing process for a complex project environment and unscheduled releases were identified, on the identified results its concluded that the described approach addresses well the described issues and furthermore efficient testing environment that combines a number of test frameworks like JUnit, Selenium with custom-developed simulators is presented.

### **3. RESEARCH METHODOLOGY**

Research methodology for the project has been discussed below.

#### **3.1. Agile Methodology**

Agile is the powerful methodology used in the industry today with 52% of companies reporting that more than half of their teams are using agile practices [24]. At the same time, developing secure software is extremely important given the extensive spread of security exploits. NIST

reported around 16,000 software vulnerabilities across the industry in 2018 [6]. For the automation of purchase order creation company has adopted the agile methodology. Agile methodology is a method which has continuous iteration of development and testing throughout the software development lifecycle (SDLC) of a project. To work on test automation, a proven test methodology and an automation framework must be used. However, using Hybrid automation framework requires to learn the techniques of the test tool used and time and effort is required for customizing [5]. Automation of purchase order follows agile methodology as agile methodology is easy for implementing. Through this approach, application is built in short cycles, thereby ensuring reliability for timely releases. This results in building, testing and releasing the software faster and more frequently [23]. The scrum methodology has its benefit as it provides increased customer satisfaction with necessary responsiveness for change requests [22].

### 3.2. Task Schedule

Table 1 represents the schedule of this project in Independent Advisory Services for the automation process for purchase order creation. The whole process was for 5 weeks where each week was labelled as one sprint. In the initial stage of project, we discuss the project and requirements which we have to work on. As part of the first sprint we come across the Microsoft dynamics 365 website where all the procedures were defined clearly. Dynamics website is different from normal web page as it was hard to inspect the elements to find the x path for automation purpose, so the site contains all the shortcuts explained detailly. In sprint 2 we had analysed the New Zealand market we compared and selected the tool which was suitable for automating the purchase order creation. In sprint 3 we had set up the environment by downloading all the jar files need to run the automation scripts. Sample scripts are developed which only contains the needy x paths and check whether its working fine. In the 4<sup>th</sup> sprint we developed a hybrid framework which is the standard industry model which will be easy to maintain by adding configuration, helper and property file storing all the necessary data and methods to run the automation process. In the final sprint we are writing the report of the automation process for purchase order creation and the power point final presentation.

Table 1: Task Schedule

Sprint	Actions	Start Date	End Date	Duration
<b>Sprint 1</b>	Discussion of the project	October 28	October 30	15 hr
	Analysing the Microsoft dynamics site	October 31	November 2	15 hr
<b>Sprint 2</b>	Exploratory testing on Microsoft dynamics 365 for inspecting the shortcuts available.	November 2	November 5	15 hr
	Review the requirements and selecting the tool for automation	November 6	November 8	15 hr
<b>Sprint 3</b>	Set up the environments	November 9	November 10	15 hr
	Sample scripts of the	November 11	November 13	15 hr

	purchase order creation			
<b>Sprint 4</b>	Developing the Hybrid framework	November 14	November 16	15 hr
	Working on different reporting facility available	November 17	November 21	15 hr
<b>Sprint 5</b>	Report writing	November 22	November 24	15 hr
	Presentation of the report	November 25	November 30	15 hr

### 3.3. Test Cases

Table 2 shows the test cases of purchase order creation for this project.

Testcase ID	Test Scenario	Pre-Conditions	Test Steps	Expected Results	Actual Results	Test Results
TC_1	Verify login and Password.	Browser Launched and Navigate to <a href="https://democontosodatadevaos.sandbox.ax.dynamics.com/?cmp=USMF&amp;mi=DefaultDashboard">https://democontosodatadevaos.sandbox.ax.dynamics.com/?cmp=USMF&amp;mi=DefaultDashboard</a>	Enter the valid Username and Password successfully.	Login must be successful.	Login successfully done.	Pass
TC_2	Create new purchase order	Login Successful	1. Click new order in left most part of Dashboard. 2. Enter the Vendor account and site name. 2. Save the details after entering mandatory fields.	Purchase order must be successful.	Purchase order Creation successfully done.	Pass
TC_3	Search Item number	Login Successful	Enter the associated item number for the vendor	Enter the associated item number for the vendor	Item number must be added successfully	Pass
TC_4	Save and confirm Purchase order	Login Successful	Click the save button and confirm the purchase order.	Click the save button and confirm the purchase order.	Purchase order Creation must be successful.	Pass

Table 2: Test Cases of purchase order creation

### 3.4. Build Tool

The build tool is used to set up the project that is essential to run the java code for the whole java project. Build tool will generate a source code, compiling code, packaging code to a jar etc. For automating the purchase order creation, we have used the build tool maven. Maven provides a common platform to perform the activities which makes developers task easy. The core of any maven project is the pom.xml, where all the information's are stored. Most of the Integrated Development Environments (IDE) are available for tools like Eclipse, NetBeans, IntelliJ etc. Maven stores all jars like selenium standalone, TestNG, Extent reports and Apache POI for the project. Library jar is placed in central repository from where maven downloads all the dependency jar for the automation process.

```

1<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
2<modelVersion>4.0.0</modelVersion>
3<groupId>com.samplecrm</groupId>
4<artifactId>com.samplecrm</artifactId>
5<version>0.0.1-SNAPSHOT</version>
6<dependencies>
7<dependency>
8<groupId>org.seleniumhq.selenium</groupId>
9<artifactId>selenium-java</artifactId>
0<version>3.141.59</version>
1</dependency>
2<dependency>
3<groupId>com.aventstack</groupId>
4<artifactId>extentreports</artifactId>
5<version>4.0.0</version>
6</dependency>
7
8
9<dependency>
0<groupId>org.apache.poi</groupId>
1<artifactId>poi</artifactId>
2<version>3.17</version>
3</dependency>
4
5<dependency>
6<groupId>org.apache.poi</groupId>
7<artifactId>poi-ooxml</artifactId>
8<version>3.17</version>
9</dependency>
0
1</dependencies>
2</project>

```

Figure 1: Pom.xml used in the Project

### 3.5. Hybrid Framework

Hybrid Framework has the capability to handle many test cases and it can produce accurate results as per the test case. Developed Hybrid framework completely reduces the manual dependency in automation testing. [4]. The main advantage of test automation comes from fast, precise execution of a set of tests after some changes have been made to a web application [21]. The aim of (test data management) TDM is to improve the effectiveness and reduce the time and cost for testing. [2].

Key features in implementing Hybrid Framework with Selenium:

- Stores the input test data in Excel file.
- Can store the environment related information in a property file.
- Store various objects in the applications where user need to access in object repository file.
- Test suite will contain the verifying tasks mentioned in the requirements.
- Executing on different browsers when needed.
- Generated screenshots to capture the passed/failed test cases.

- Multiple Report generation through TestNG, Extent Reports [2].

A framework is a set of assumptions, concepts and practices that needs to be followed. There are many components of framework:

- Test case standardization: Every test must be in a proper format, in order to achieve this, we must follow a design pattern so for the automation of purchase order creation we are using page object model (POM).
- Logs: For each execution we have to generate the logs to check what went wrong so generation of logs are of primary importance.
- Test data and configuration utility: The most important part of the framework is the test data without the test data we cannot fill the mandatory fields while automating, so hard coring the values are not a proper way in Industries because in the further sprints if we want to change the data we have to keep the test data and utility file separately.
- Helper or Utility library: While working on different pages the excel data, file code to read, different browser action will be changed for accessing these libraries for accessing these kinds of information will be stored in Helper and utility libraries.
- Test Execution engine: We must define how our final test will run, which build tools are we using or for accessing libraries whether we are using any continuous integration tools like Jenkins for configuration. For the automation purpose we are using the build tool Maven and the proper way of using like add all the needful dependency in the pom.xml.
- Reporting: The main component of the framework is reporting without proper reports we cannot show the stakeholders the test execution results. In a proper report we have all the details about the methods we made against the test cases, how much time it takes to execute the test cases and a simple diagram showing the pass and fail status.

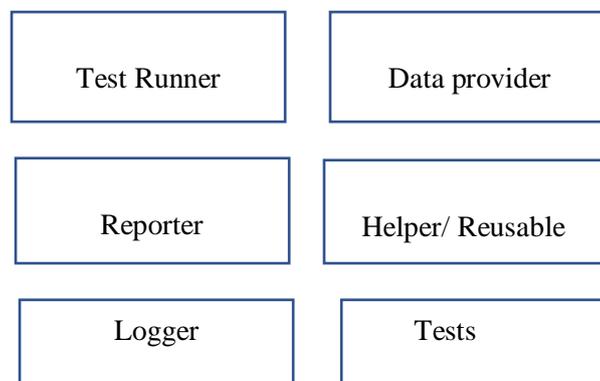


Figure 2: Basic framework design

### 3.6. Reporting

For the formation of robust Hybrid framework most important element is Reporting. A good test report serves as an overview of the project which helps in finding bugs and send the test report to the Stakeholders to show the nutshell of the test case execution.

Main features of the test reports are:

- Short and precise.
- Show the number of passed and failed test cases.
- Show a pictorial diagram of test execution results.

- Must support a format which can be shared by email or integrate with Continuous Integration (CI) tools like Jenkins/Bamboo..

Show the test coverage of application/module under test.

### 3.7. Generated Reports in Selenium Web Driver

As an open source Selenium Web driver does not have a built-in reporting feature, as we have added the Maven dependency for Extent Report and TestNG for generating reports.

#### 3.7.1. Testing Reporting

TestNG is a testing framework modified from JUnit and NUnit having additional features which is easy to handle. TestNG is an open source automated testing framework, where NG means Next Generation. TestNG gives the developer to write more flexible and powerful test scripts which borrows from Java Annotations to define tests in a real production environment. Reporting reduces the problems faced by manual testers and time spent on initial set-up activity to carryout test scripts execution and overcome disadvantages of execution using TestNG [1].

#### TestNG Features

- Supports annotations.
- Supports testing integrated classes.
- Flexible runtime configuration.
- Supports Dependent test methods, load testing, and limited failure.
- Flexible plug-in API.
- Supports multi-threaded testing.

JDK	1.5 or above.
Memory	No special requirements
Disk Space	No special requirements
Operating System	No special requirements

Table 3: System Requirements

The screenshot displays a TestNG test results report. The main title is 'Test results' with a sub-header '1 suite'. Below this, there is a section for 'All suites' with a dropdown menu set to 'Default suite'. The 'Info' section shows the test suite path: 'C:\Users\vijay\AppData\Local\Temp\testing-eclipse-1273045522\testing-customsuite.xml', 1 test, 0 groups, Times, Reporter output, Ignored methods, and Chronological view. The 'Results' section indicates '3 methods, 3 passed' and lists the passed methods: 'confirmpurchase', 'createpurchase', and 'login', each with a green checkmark icon. On the right side, the test suite 'com.purchaseordertestcases.logintest' is expanded, showing the three methods: 'confirmpurchase', 'createpurchase', and 'login'.

Figure 3: Index.html

Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Default suite						
<a href="#">Default test</a>	3	0	0	32,738		
Class	Method	Start	Time (ms)			
Default suite						
Default test — passed						
com.purchaseordertestcases.logintest	<a href="#">confirmpurchase</a>	1574381260965	5112			
	<a href="#">createpurchase</a>	1574381253600	7266			
	<a href="#">login</a>	1574381245147	8010			
<b>Default test</b>						
com.purchaseordertestcases.logintest#confirmpurchase						
<a href="#">back to summary</a>						
com.purchaseordertestcases.logintest#createpurchase						
<a href="#">back to summary</a>						
com.purchaseordertestcases.logintest#login						
<a href="#">back to summary</a>						

Figure 4: Output of TestNG

### 3.7.2. Extent Reports

Different types of reporting are available in selenium, but for customizable report generation which must be shared with Stakeholders we have used extent reports which can be integrated into Selenium WebDriver using JUnit and TestNG frameworks. Extent Reports have more advantages when compared to the reports generated through JUnit and TestNG such as pie chart representation, test stepwise report, adding screenshots etc at every step and an attractive GUI which show the pass and fail test cases which can be shared to stakeholders.

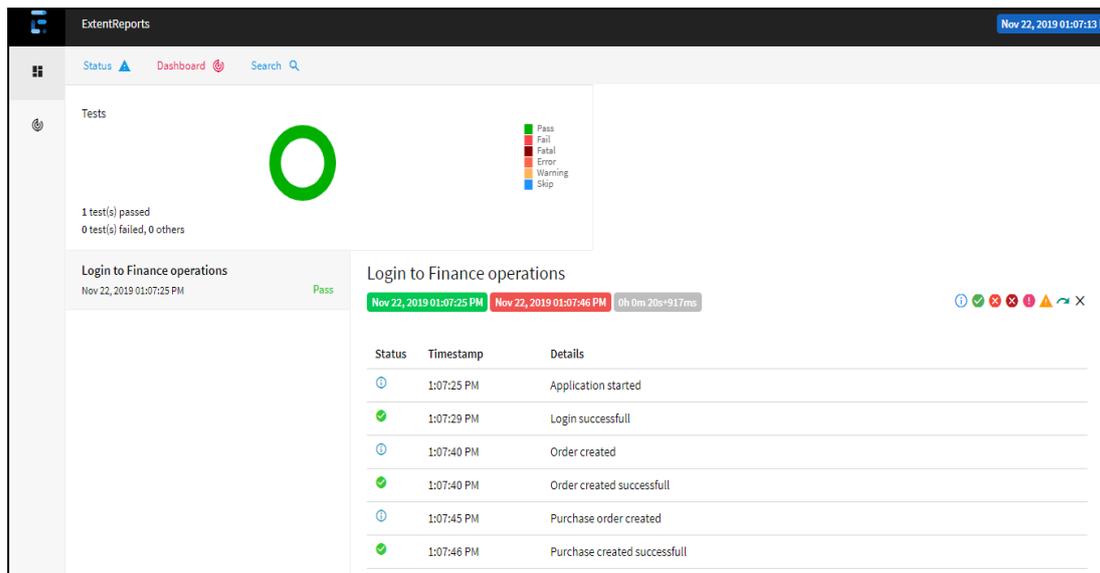


Figure 5: Output report of Extent Report

## 4. DISCUSSION

While automating the purchase order creation we have faced many problems in this testing life span. The problems and solution found are discussed below:

- In the initial stage of automation testing of the purchase order creation we found that x-path is not detectable in the Microsoft dynamics site so we went to the site of Microsoft dynamics where there is a list how to do the basic keyboard shortcuts details are available. Control + Shift + I is the shortcut key for finding the x-path.
- After writing the scripts and while running it may take a certain time limit to load the dynamics web page. So, to overcome we must put implicit wait for the page to fully load to pick the x path for automation process.
- While running the automation scripts the browser may stuck at some point so while clearing the cache or closing the automated browser, we can solve this problem.
- Google web driver may be updated at every point so we have to download the new version according to the current google version we are using, else we use an if case either to select any browser like Firefox, Internet explorer so in the configuration file if you are changing the browser name other than chrome it will run in the selected browser.

### 4.1. Comparative Analysis for Automation Tool Selection

In order to attain maximum robust framework which is stable and easy to maintain we have to search for the various tools for automating the application and compared the best two, Selenium and UFT(QTP) and select the best one for automating the purchase order creation.

Table 4 represents the comparison of Selenium and Microfocus-UFT.

FEATURES	SELENIUM	UFT(QTP)
<b>Cost</b>	Free tool/open source. No issues in licensing or renewal. Just need to download it and use	Paid tool and will cost an average of \$3200, UFT is available as seat-based and concurrent which will be more expensive.
<b>Support</b>	Since it's an open source no professional support is available.	Since it is a paid tool proper support team is available.
<b>Application Type</b>	Inbuilt, selenium supports only Web Applications. It recognizes the elements on screen using id, CSS selector, xpath.	It supports web, mobile, API, hybrid, RPA, and enterprise application.
<b>Languages supported</b>	Java, C#, Ruby, Python, Perl PHP, JavaScript, R etc.	VBS (Visual Basic Script)
<b>Supported Browsers</b>	IE, Firefox, Chrome, Safari, Opera, Headless browsers.	Chrome, Firefox, Safari, IE, and Edge.
<b>Coding skills</b>	Good knowledge of programming language is needed for each Binding.	Less programming knowledge is required as it offers keyword-driven testing which simplifies test creation and maintenance.

<b>Test Report</b>	Selenium has to download the necessary plugins to generate test reports.	Default test reports are generated.
<b>Performance Testing</b>	Selenium is not used for performance testing but can be integrate with JMeter, to run your selenium scripts for performance testing.	We can mimic the user actions in UFT.
<b>Tools Integration</b>	Can be integrated with paid or free tools	Can be integrated limited paid tools

Table 4: Selenium vs UFT

According to google trends survey globally the mostly used is Selenium. Selenium is a one of the efficient open-source automated testing tool which provide a stable testing framework for testing a wide variety of applications and exporting scripts in almost every language including java, .net, c#. The main feature of Selenium is that it supports different browsers for executing the test cases [20]. In this project Selenium Web driver is used to automate the purchase order and to illustrate the use of selenium tool in combination with other tools like the Maven, TestNG, etc., for more easier approach to testing and to improve the quality of testing process [19]. The inclusion of the Page Object pattern has demonstrated that it will be very effective in end-to-end testing. Page objects are classes which abstracts the web pages into required business functions that can be called by the test cases. By decoupling source code from web page information, test cases made for the web pages are more readable and easier to maintain [21]. The main use of selenium for automation purpose is the flexibility. Selenium features like regrouping and refactoring helps the developers and testers to maintain and quickly change the code.

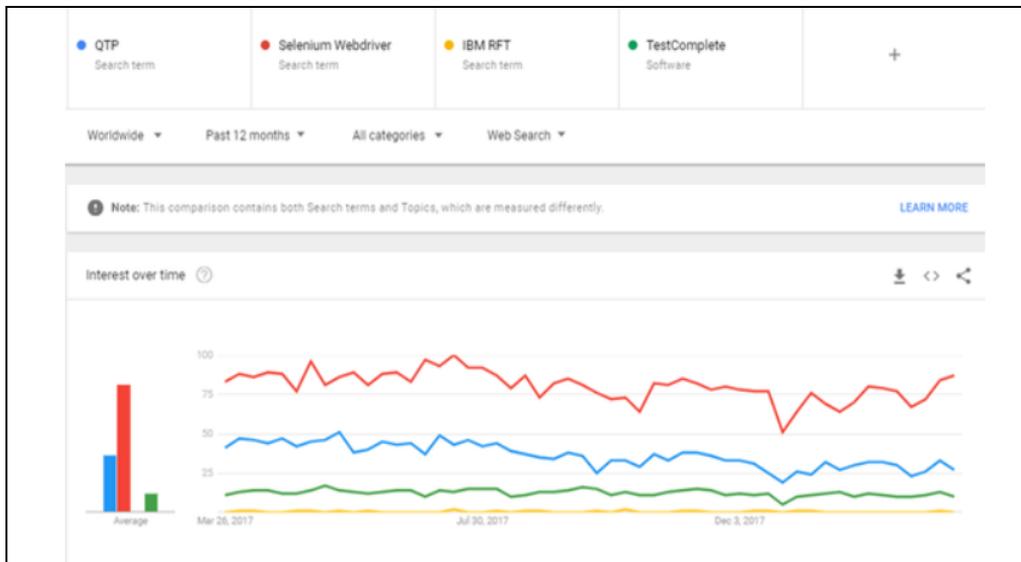


Figure 6: Automation Tools usage Globally

Selenium is selected when compared to the other tool because of the following reasons:

- We can automate the purchase order creation at free of cost.
- Developed hybrid framework is easy to maintain for the future sprints if required.
- For scheduling the jobs, we can integrate with Jenkins.
- Various style of Reporting like Extend reporting, TestNG can be done with Selenium.

- Failed test cases are identified through screenshots which has included in the framework.

## 5. FUTURE RECOMMENDATIONS

Our task about the automation of creation of purchase order in Microsoft Dynamics 365 in selenium web driver. The development process for the automation can be performed by a Behaviour Driven Development (BDD) either by using Cucumber or Spec flow. The acceptance criteria can be added in the feature file so that team members who does not know about the coding like Stakeholder, Business Analyst can understand how the build is made against the requirements. Step definition file is made against each step in the feature file which will be easy to correct the error.

## 6. CONCLUSION

Automation empowers enterprises to reduce resource allocation, reduce cost to company (CTC), enables business that ensures customer satisfaction. In this execution, execution of automating the purchase order creation for the company Independent Advisory Services has been discussed. Selection of optimum tool from the New Zealand market for automating the purchase order has been discussed. Hybrid Framework has been developed for the automation process and how the framework is used for future maintenance in the upcoming sprints. Different methods have been used for report generation have been discussed. Necessary recommendation and the problems faced during this project has been mentioned. The proposal concluded about what are the testing activities that is to be done for automating the purchase order and to identify test cases and prepare the test scripts and generate the HTML reports.

## REFERENCES

- [1] Jain, C. R., & Kaluri, R. (2015). Design of automation scripts execution application for selenium webdriver and test NG framework. *ARN J Eng Appl Sci*, 10, 2440-2445.
- [2] Bajaj, K. S. (2018). Hybrid Test Automation Framework for managing Test Data. *International Journal of Pure and Applied Mathematics*, 118(9), 265-277.
- [3] Bajaj, H. (2015). Choosing the right automation tool and framework is critical to project success. Infosys Limited.
- [4] Lamba, S., Rishiwal, V., & Rana, A. (2015). An automated data driven continuous testing framework. *International Journal of Advanced Technology in Engineering And Science*, 3(1).
- [5] Shim, J. A., Kwon, H. J., Jung, H. J., & Hwang, M. S. (2016). Design of acceptance test process with the application of agile development methodology. *International Journal of Control and Automation*, 9(2), 343-352.
- [6] NIST (2019). National Vulnerability Database.
- [7] Gojare, S., Joshi, R., & Gaigaware, D. (2015). Analysis and Design of Selenium WebDriver Automation Testing Framework. *Procedia Computer Science*, 50, 341-346.
- [8] Subramanian, R., Chen, N., & Zhu, T. (2017). Behavior Driven Test Automation Framework. In *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)* (pp. 17-23). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

- [9] Chandraprabha, C., Kumar, A., & Saxena, S. (2015). Data Driven Testing Framework using Selenium WebDriver. *International Journal of Computer Applications*, 118(18), 18–23.
- [10] Leotta, M., Clerissi, D., Ricca, F., & Spadaro, C. (2013). Comparing the maintainability of selenium WebDriver test suites employing different locators: a case study. *Proceedings of the 2013 International Workshop on Joining AcadeMiA and Industry Contributions to Testing Automation - JAMAICA 2013*.
- [11] Razak, R. A., & Fahrurazi, F. R. (2011). Agile testing with Selenium. *2011 Malaysian Conference in Software Engineering*.
- [12] Jain, R., Johnson, B. K., & Hess, H. L. (2015, July). Performance of line protection and supervisory elements for doubly fed wind turbines. In *2015 IEEE Power & Energy Society General Meeting (pp. 1-5)*. IEEE.
- [13] Bures, M., & Filipisky, M. (2016). SmartDriver: Extension of selenium WebDriver to create more efficient automated tests. In *2016 6th International Conference on IT Convergence and Security (ICITCS) (pp. 1-4)*. IEEE
- [14] Srinivas, K., & Prakash, C. (2017). A Comparative Study of Testing Framework with Special Emphasis on Selenium for Financial Applications. *International Journal of Soft Computing*, 12(3), 148-155.
- [15] Hanna, M., Aboutabl, A. E., & Mostafa, M. S. M. (2018). Automated Software Testing Framework for Web Applications. *International Journal of Applied Engineering Research*, 13(11), 9758-9767.
- [16] Sneha, K., & Malle, G. M. (2017). Research on software testing techniques and software automation testing tools. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 77-81)*. IEEE.
- [17] Dhir, S., & Kumar, D. (2019). Automation Software Testing on Web-Based Application. In *Software Engineering (pp. 691-698)*. Springer, Singapore.
- [18] Berłowski, J., Chruściel, P., Kasprzyk, M., Konaniec, I., & Jureczko, M. (2016). Highly automated agile testing process: An industrial case study. *e-Informatica Software Engineering Journal*, 10(1).
- [19] Ramya, P., Sindhura, V., & Sagar, P. V. (2017). Testing using selenium web driver. *2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT)*.
- [20] Singh, I., & Tarika, B. (2014). Comparative analysis of open source automated software testing tools: Selenium, sikuli and watir. *International Journal of Information & Computation Technology*, 4(15), 1507-1518.
- [21] Stocco, A., Leotta, M., Ricca, F., & Tonella, P. (2016). APOGEN: automatic page object generator for web testing. *Software Quality Journal*, 25(3), 1007–1039.
- [22] Mahalakshmi, M., & Sundararajan, M. (2013). Traditional SDLC Vs Scrum Methodology—A Comparative Study. *International Journal of Emerging Technology and Advanced Engineering*, 3(6), 192-196.
- [23] Petre, I., Smada, D. M., & Boncea, R. (2017). A user-centric approach to test automation of web-based applications. In *RoCHI (pp. 49-52)*.
- [24] VersionOne. (2018). 12th Annual State of Agile Report.

**AUTHOR**

**Vijay** is a qualified Software Test Engineer and an engineering graduate in Information and Technology from AGI institute in Graduate Diploma in Software Testing and have 5 years of experience in software testing competency. Vijay have been working in Cognizant Technology Solutions for around 3 years. After completion of the graduation course Vijay is currently working in Duco Consultancy as a Test Engineer in New Zealand.



**Dr. Shahid Ali** is a senior lecturer and IT program leader at AGI Education Limited, Auckland, New Zealand. He has published number of research papers on ensemble learning. His expertise and research interests include ensemble learning, machine learning, data mining and knowledge discovery.

© 2020 By AIRCC Publishing Corporation. This article is published under the Creative Commons Attribution (CC BY) license.