Journal of Applied Science and Engineering, Vol. 21, No. 3, pp. 455–462 (2018) *DOI: 10.6180/jase.201809_21(3).0017*

Development of Reusable Hybrid Test Automation Framework for Web Based Scrum Projects

Sulabh Tyagi[1]*, Ritu Sibal[1], Bharti Suri[2], Bimlesh Wadhwa[3] and Shanuj Shekhar[4]

[1] *Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India*
[2] *Guru Gobind Singh Indraprastha University, New Delhi, India*
[3] *National University of Singapore, Singapore*
[4] *National Institute of Technology, Jalandhar, Punjab, India*

Abstract

Web based applications are gaining widespread popularity and most of these applications are being developed using agile development methods mainly scrum. This invariably makes the testing of any web application necessary and important before it goes online. Agile methods advocate automated testing, which is the only way to assure faster thorough testing of any web application. In this paper, authors propose a reusable hybrid test automation framework (RHTAF) using page object model for automated testing of web applications. RHTAF combines the features of both data driven and keyword driven frameworks and developed using Selenium 2.0 tool. The primary focus of this framework is to ensure reusability and maintainability of test scripts so as to speed up the testing process of web applications.

***Key Words*:** Test Automation, Agile Software Development, Scrum, Frameworks, Web Applications

1. Introduction

Testing is an integral part of any agile development method and is primarily automated. Agile methods are being embraced by primary areas of computing for software development and one of these is web engineering. According to [1] the most widely used agile methods are scrum and XP. Of these two, scrum follows sprint based development and gives leverage to customers by providing potentially shippable product at the end of every sprint.

To deliver a quality product at the end of every sprint it is important to overcome certain challenges like short development cycles, automating daily activities like smoke testing, regression testing, continuous integration and deployment process. Furthermore, with every subsequent sprint, it becomes inevitable to check proper functioning of existing functionalities. Automated testing assures thorough testing in spite of tight deliverable schedules. Our previous paper [2] reported the test script maintenance as one of the key challenges faced by agile practitioners while practicing test automation on their projects, this further leads us to develop our research questions:

RQ1: How to develop a test automation framework that can provide high reusability and maintainability of test scripts?

RQ2: How to use this framework in actual agile software development methods like scrum?

In order to answer these questions, it is important to establish the understanding of different test automation frameworks and automation tools that are prevalent in software industry.

An automation framework is defined as an overall architecture that contains test data, test cases, test scripts, test automation tool, test reporting and proper strategy to execute the testing problem. Automation frameworks are broadly of three types, namely:

1. Data Driven Framework - It is primarily used for executing set of test steps with multiple sets of data [3]. In

*Corresponding author. E-mail: sulabhtyagi2k@yahoo.co.in this framework, set of test input is read from data files like CSV files, Excel files, ADO objects, etc. and are loaded into variables in captured or manually coded script.

2. Keyword Driven Framework - In this framework each test step of every test case is associated with a keyword. Keywords are stored in excel file and driver script reads them to perform user actions.
3. Hybrid Framework - It is a combination of both data driven and keyword automation framework. This frameworks stores keyword for each test case step and at the same time test data can also be read from excel files to run multiple sets of data.

Many test automation tools are available in the market today like QTP (quick test functional) test complete and selenium. We used selenium tool to implement RHTAF and reasons to prefer selenium over other tools are; firstly, selenium is open source tool and cannot levy any license fee. Secondly, selenium gives leverage to automate web applications with multiple programming language support like Java, C# and Python. Recently, Selenium RC and Web Driver are merged to form Selenium 2.0 which offers many advantages like direct communication with the browser, Simple API's without any redundancy and easy architecture [3]. Selenium 2.0 supports multiple web browsers like Chrome, Firefox, Internet Explorer and Safari [4].

In this paper, we propose a hybrid automation framework with Selenium 2.0 using Page Object Model (POM). Normally, any web application comprises of multiple web pages and each webpage has a different set of objects also know as UI elements to interact with. Moreover, most of the time a single UI element is being used in multiple test cases sometimes as many as twenty or thirty odd test cases so if that UI element undergoes any change than one has to traverse these twenty or thirty odd test cases to update that change. By using POM, change has to be made under that particular page object class containing the reusable method for that particular UI element and all the test cases need to call that reusable method of that particular POM class. Hence, a page object model simply models the objects within the test code. This reduces the amount of duplicated code and thus, if the UI changes frequently, the fix needs only to be applied at one place.

The rest of this paper is structured as follows: next section briefs about background; the third section describes the implementation and experimental analysis of this study; the fourth section presented the results; and the last section concludes the paper.

2. Background

Today most of the software industries are migrating from bulkier development methods to lightweight agile methods. In agile methods automation testing strategies are followed in order to improve quality of the product, reduce cycle time and for optimal utilization of manpower [5]. According to [6], technical debt accumulates if development team opts shortcut approach, hacks in quick fixes, skips writing automating test and eventually code base gets harder and harder to maintain and this leads to the higher maintenance cost and lower team velocity. Tyagi et al. [2] reported the use of test automation frameworks in solving the test script maintenance problem for agile software development projects. Moreover, other studies [7,8] also depicted the importance of handling the test script maintenance and effective test automation. Rafi et al. [9] discussed the superiority and benefits of test automation when several regression rounds are needed.

In study by Puleio's [10] test automation was seen as a key factor in agile testing to keep development and testing in synchronization. Test automation is of prime importance in agile projects, continuous testing would have been impossible without test automation. Hui et al. [11] argued that keyword driven automated testing framework has better reusability than recording/playback and data driver frameworks and presented keyword driven distributed framework for Linux platform. In [12] authors presented extension of selenium RC tool to perform automated database testing in web applications.

3. Proposed Framework

In this paper, we propose a reusable hybrid test automation framework using page object model. Use of page object model in designing this hybrid test automation framework makes it unique. The need of continuous delivery of features and change requests with every subsequent sprint in an agile development project makes it prone to higher maintenance costs and development time, hence without in place test automation framework it is very difficult to cope up with such challenges. Our test automaton framework which is based on POM would inherently make script maintenance easier and also makes test execution simpler and faster. Eclipse IDE is used to develop RHTAF using Selenium 2.0. Eclipse is preferred over Visual Studio due to many reasons like eclipse is open source IDE, test script creation and test execution is also fast and easy in eclipse IDE.

3.1 Three Layer Architecture of Proposed Reusable Hybrid Automation Framework

We implemented reusable hybrid test automation framework using three layered architecture as shown in Figure 1 below.

Layer 1 - It comprised of Application Under Test (AUT) that could be run on any browser such as internet explorer, google chrome orfirefox. This study used two famous e-commerce portals-www.amazon.in and www.flipkart.com as its web applications under test. Chrome is used as a web browser to test our AUT.

Layer 2 - Actual hybrid test automation framework that is build using selenium web driver was implemented at this layer. Major components of layer 2 are as follows:

Driver Script - Driver script consist of main() function that consist of set of tests that calls the reusable methods under the respective POM Class along with the ReadExcelFile() function which reads the excel file containing sheets row by row which shows the hybrid nature of framework (Combination of Keyword and Data Driven Framework) as shown in Figure 2.

POM Class - It contains all the objects and actions to be performed on that object within a particular webpage. Example-POM class LoginPage.java is created for automating the login functionality and reusable methods were written for automating different user actions like enter username, enter password etc. on that page. Similarly for each webpage, a separate POM class has been created which contains its own member functions.

Error/Exception handling - These are functions which take care of unexpected errors such as error or warning messages during execution of the test case.

Result (Report) - This refers to the reporting of the test cases execution report in the prescribed format for the analysis. We used extent report plug-in to generate the reports for every executed test case.

Layer 3 - It comprises of input data files in .xls format containing test data for each test scenario that is one column containing all the reusable keywords for each web page of our application under test and other column is for the data driven framework mainly in the form of excel file as shown in Figure 2. Our framework also has the screenshots functionality which captures the screenshot in case of failed test case to record any error or warning messages during execution of the test script.

3.2 Implementation and Experimental Analysis of RHTAF in Scrum Environment

The implementation of our reusable hybrid test auto-

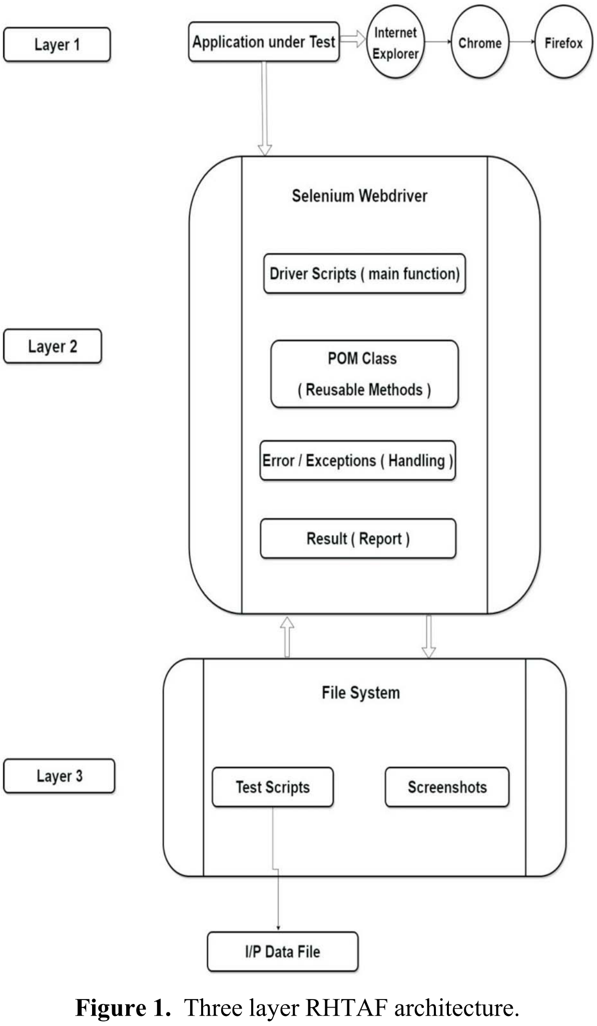
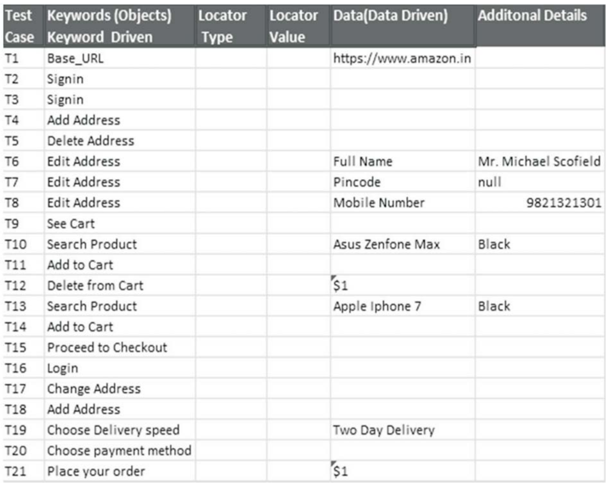


Figure 1. Three layer RHTAF architecture.



Test Case	Keywords (Objects) Keyword Driven	Locator Type	Locator Value	Data(Data Driven)	Additonal Details
T1	Base_URL			https://www.amazon.in	
T2	Signin				
T3	Signin				
T4	Add Address				
T5	Delete Address				
T6	Edit Address			Full Name	Mr. Michael Scofield
T7	Edit Address			Pincode	null
T8	Edit Address			Mobile Number	9821321301
T9	See Cart				
T10	Search Product			Asus Zenfone Max	Black
T11	Add to Cart				
T12	Delete from Cart			$1	
T13	Search Product			Apple Iphone 7	Black
T14	Add to Cart				
T15	Proceed to Checkout				
T16	Login				
T17	Change Address				
T18	Add Address				
T19	Choose Delivery speed			Two Day Delivery	
T20	Choose payment method				
T21	Place your order			$1	

Figure 2. Input excel file containing keywords and data for RHTAF.

mation framework (RHTAF) using page object model was performed on www.amazon.in and www.flipkart.com, two widely used e-commerce websites in India. RHTAF is implemented and experimentally analyzed by simulating scrum environment, one of the most popular and widely used agile software development methods where manual test cases have been written for the business critical functionality for the two websites, then we split the total number of test cases in three successive sprints (sprint is a time box effort to deliver potentially shippable product). Our sprint duration was two weeks per sprint, RHTAF was developed and applied to automate these test cases on sprint by sprint basis.

3.2.1 Insprint Test Automation Test Strategy

We also propose insprint automation test strategy to use our RHTAF on web based projects developed under scrum environment. In insprint automation test strategy, firstly, test scenarios have been identified for each sprint then regression test cases were designed for that particular sprint based on the business criticality. Finally regression test cases from each sprint were automated using our RHTAF. Present study implemented RHTAF over three sprints and automated regression test cases of these two websites has been shown in Table 1. In sprint 1, regression test cases that cover important business critical test cases were automated that includes Login, Add Address, Delete Address and Edit Address functionalities for both the websites. These test cases from T1-T8 became an initial part of our regression test suite denoted by R1.1 which means regression test suite for release 1 and sprint 1. Similarly, In sprint 2, test cases from T9-T14 were automated and added to regression test suite R1.1 that became our regression test suite R1.2. In the same way, we automated test cases from T15 to T20 that were added to regression test suite R1.2 and we got our completed regression suite for release 1 that is R1.3 as shown in Figure 3 and our complete regression test suite can be calculated as: complete regression suite (R1) = (T1-T20) = R1.3.

Automating our regression tests using RHTAF on sprint by sprint basis gave us leverage to had one subset of regression suite ready by the end of sprint 1 which we ran at the start of sprint 2. Similarly, we had one subset of regression suite ready by the end of sprint 2 which we ran at the beginning of sprint 3 and finally at the end of sprint 3 our whole regression test suite was ready which we ran at the hardening sprint to check overall functionally of our application under test. Due to space reasons we are showing implementation of our framework with the help

Table 1. Different regression test cases automated over 3 sprints

Sprint number	Test case number	Test case name	Test case scenario (positive or negative)
1.	T1	Open browser	Positive
1	T2	Login	Negative
1.	T3	Login	Positive
1.	T4	Add address	Positive
1.	T5	Delete address	Positive
1.	T6	Edit address	Negative
1.	T7	Edit address	Positive
1.	T8	Cart lookup	Positive
2.	T9	Search product	Positive
2.	T10	Add to cart	Positive
2.	T11	Delete from cart	Positive
2.	T12	Search product	Positive
2.	T13	Add to cart	Positive
3.	T14	Proceed to checkout	Positive
3.	T15	Login	Positive
3.	T16	Change address	Positive
3.	T17	Add address	Positive
3.	T18	Choose delivery speed	Positive
3.	T19	Choose payment method	Positive
3.	T20	Place your order	Positive

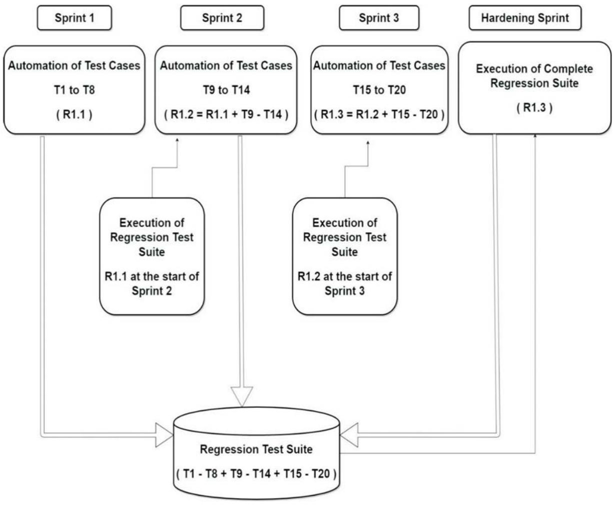

Sprint 1
Sprint 2
Sprint 3
Hardening Sprint
Automation of Test Cases T1 to T8 (R1.1)
Automation of Test Cases T9 to T14 (R1.2 = R1.1 + T9 - T14)
Automation of Test Cases T15 to T20 (R1.3 = R1.2 + T15 - T20)
Execution of Complete Regression Suite (R1.3)
Execution of Regression Test Suite R1.1 at the start of Sprint 2
Execution of Regression Test Suite R1.2 at the start of Sprint 3
Regression Test Suite (T1 - T8 + T9 - T14 + T15 - T20)

Figure 3. Block diagram showing implementation of RHTAF over three sprints in scrum environment.

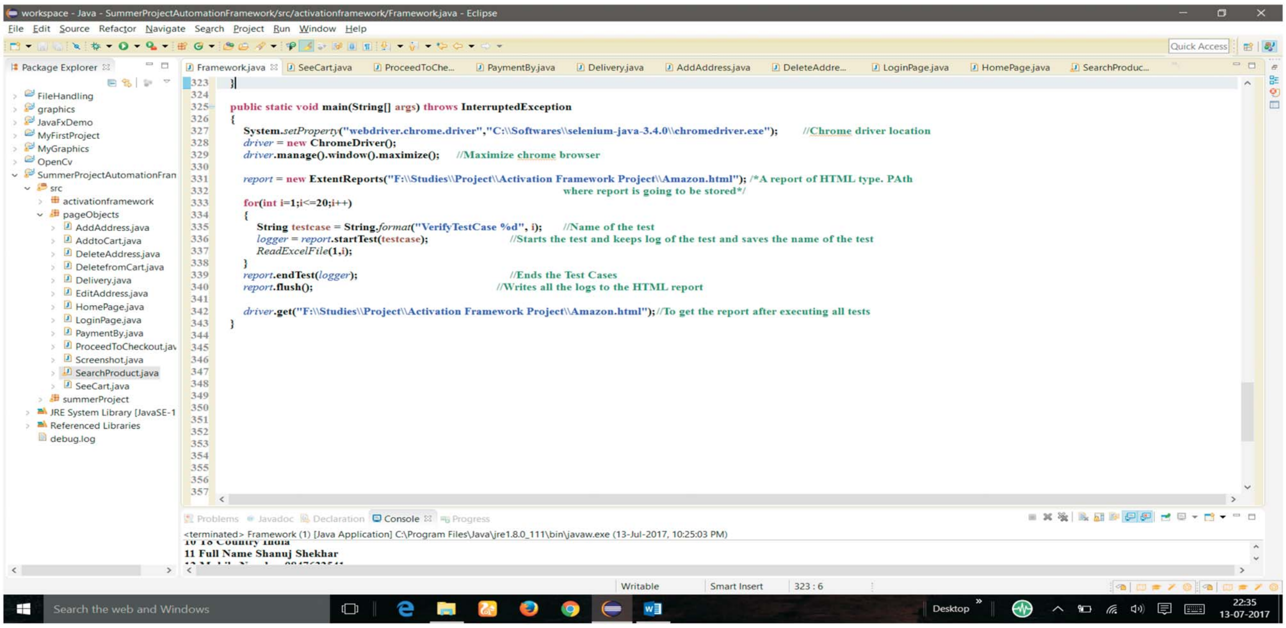
of some selected Figures from every sprint as given below. Firstly, Driver script that is Main function() calls the ReadExcelFile() function as shown in Figure 4 which reads the input excel file row by row and when it reads the second element (keyword) as shown in Figure 2 of the row then corresponding page object class is called and corresponding member function (reusable method) is being executed. Hence for the second test case i.e. T2, main function() calls the ReadExcelFile() function which reads the input excel file and on finding "Signin" keyword, it calls the LoginPage.Java POM class which in turn executes the corresponding member function for executing login test case for amazon.in as shown in Figure 5 and Figure 6.

In sprint 2, for Search Product test case; main function calls the ReadExcelFile() function which reads the excel file row by row and on finding "Search Product" keyword, it calls the SearchProduct.Java POM class which in turn executes the corresponding member function for executing Search Product test case for amazon.in as shown in Figure 7 and Figure 8.

Similarly, in sprint 3, for Choosing a Payment Method test case; main function calls the ReadExcelFile() function which reads the excel file row by row and on finding "Payment By" keyword, it calls the PaymentBy.Java POM class which in turn executes the corresponding member function for executing Search Product test case for amazon.in.

4. Results

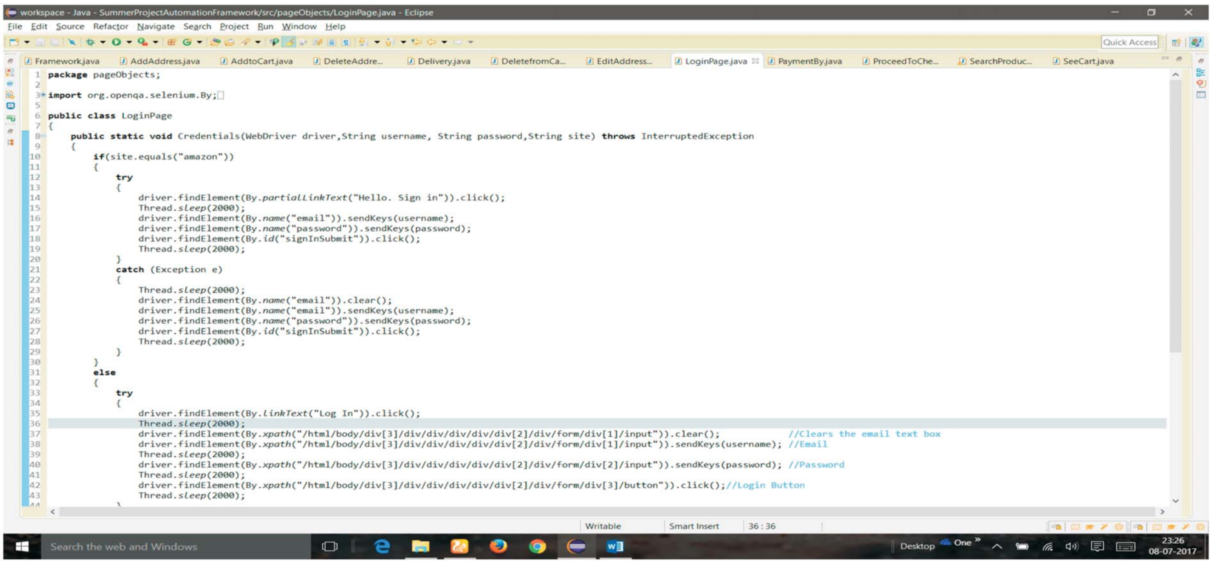
We have successfully implemented reusable hybrid test automation framework using page object model for web based applications in simulated scrum environment. Our experimental analysis of our RHTAF on amazon.in and flipkart.com showed that our regression testing effi-



```
}

public static void main(String[] args) throws InterruptedException
{
    System.setProperty("webdriver.chrome.driver","C:\\Softwares\\selenium-java-3.4.0\\chromedriver.exe");        //Chrome driver location
    driver = new ChromeDriver();
    driver.manage().window().maximize();    //Maximize chrome browser

    report = new ExtentReports("F:\\Studies\\Project\\Activation Framework Project\\Amazon.html"); /*A report of HTML type. PAth
                                                        where report is going to be stored*/
    for(int i=1;i<=20;i++)
    {
        String testcase = String.format("VerifyTestCase %d", i);      //Name of the test
        logger = report.startTest(testcase);                     //Starts the test and keeps log of the test and saves the name of the test
        ReadExcelFile(1,i);
    }
    report.endTest(logger);                                      //Ends the Test Cases
    report.flush();                                             //Writes all the logs to the HTML report

    driver.get("F:\\Studies\\Project\\Activation Framework Project\\Amazon.html");//To get the report after executing all tests
}
```

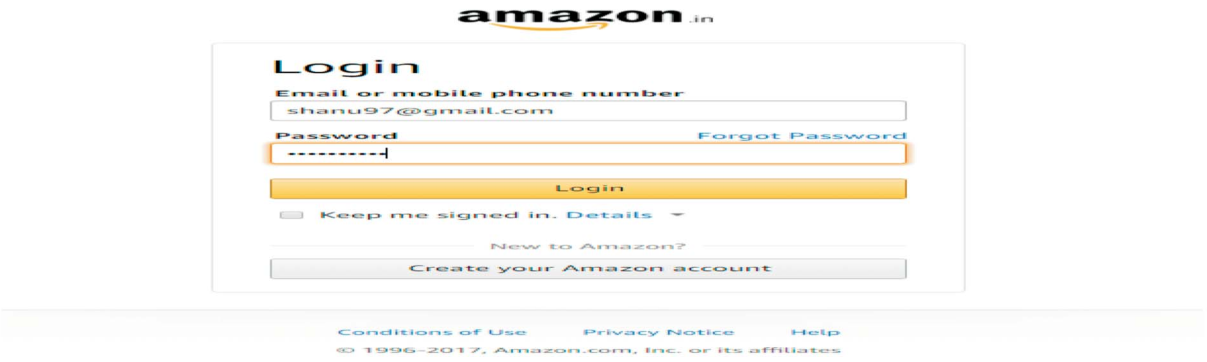
```
<terminated> Framework (1) [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (13-Jul-2017, 10:25:03 PM)
10 18 Country India
11 Full Name Shanuj Shekhar
```

Figure 4. Driver (main) function of www.amazon.in.



```
package pageObjects;

import org.openqa.selenium.By;

public class LoginPage
{
    public static void Credentials(WebDriver driver,String username, String password,String site) throws InterruptedException
    {
        if(site.equals("amazon"))
        {
            try
            {
                driver.findElement(By.partialLinkText("Hello. Sign in")).click();
                Thread.sleep(2000);
                driver.findElement(By.name("email")).sendKeys(username);
                driver.findElement(By.name("password")).sendKeys(password);
                driver.findElement(By.id("signInSubmit")).click();
                Thread.sleep(2000);
            }
            catch (Exception e)
            {
                Thread.sleep(2000);
                driver.findElement(By.name("email")).clear();
                driver.findElement(By.name("email")).sendKeys(username);
                driver.findElement(By.name("password")).sendKeys(password);
                driver.findElement(By.id("signInSubmit")).click();
                Thread.sleep(2000);
            }
        }
        else
        {
            try
            {
                driver.findElement(By.linkText("Log In")).click();
                Thread.sleep(2000);
                driver.findElement(By.xpath("/html/body/div[3]/div/div/div/div/div[2]/div/form/div[1]/input")).clear();              //Clears the email text box
                driver.findElement(By.xpath("/html/body/div[3]/div/div/div/div/div[2]/div/form/div[1]/input")).sendKeys(username); //Email
                Thread.sleep(2000);
                driver.findElement(By.xpath("/html/body/div[3]/div/div/div/div/div[2]/div/form/div[2]/input")).sendKeys(password); //Password
                Thread.sleep(2000);
                driver.findElement(By.xpath("/html/body/div[3]/div/div/div/div/div[2]/div/form/div[3]/button")).click();//Login Button
                Thread.sleep(2000);
```

Figure 5. Sprint 1 - automation of login functionality using Page Object Model for amazon.in.

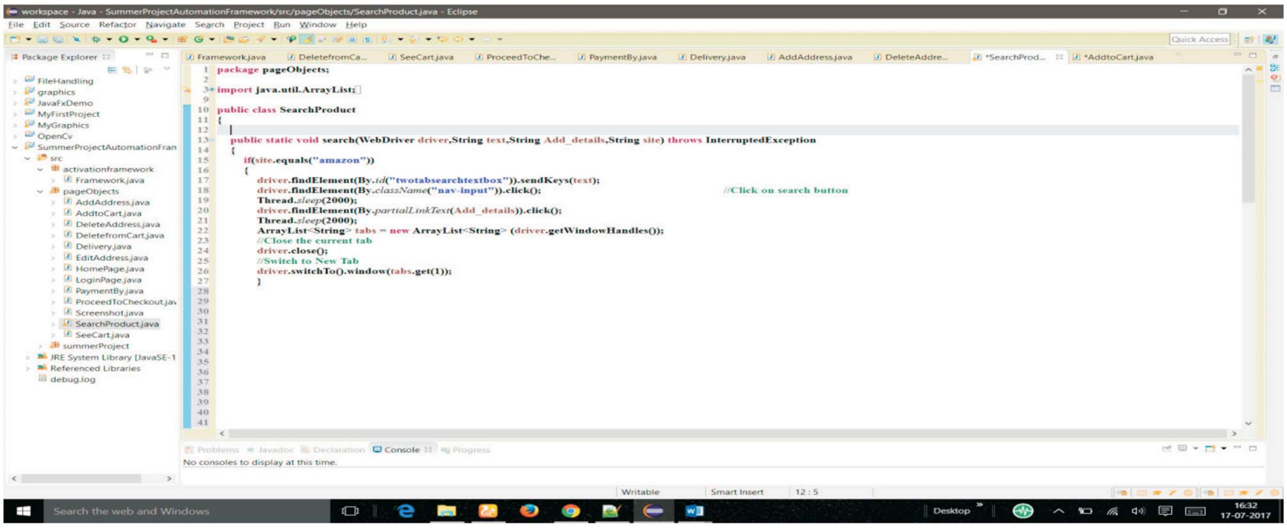

amazon.in
Login
Forgot Password
Login
Keep me signed in. Details
New to Amazon?
Create your Amazon account
Conditions of Use
Privacy Notice
Help
© 1996-2017, Amazon.com, Inc. or its affiliates

Email or mobile phone number: shanu97@gmail.com

Password: ●●●●●●●●●●

□ Keep me signed in.

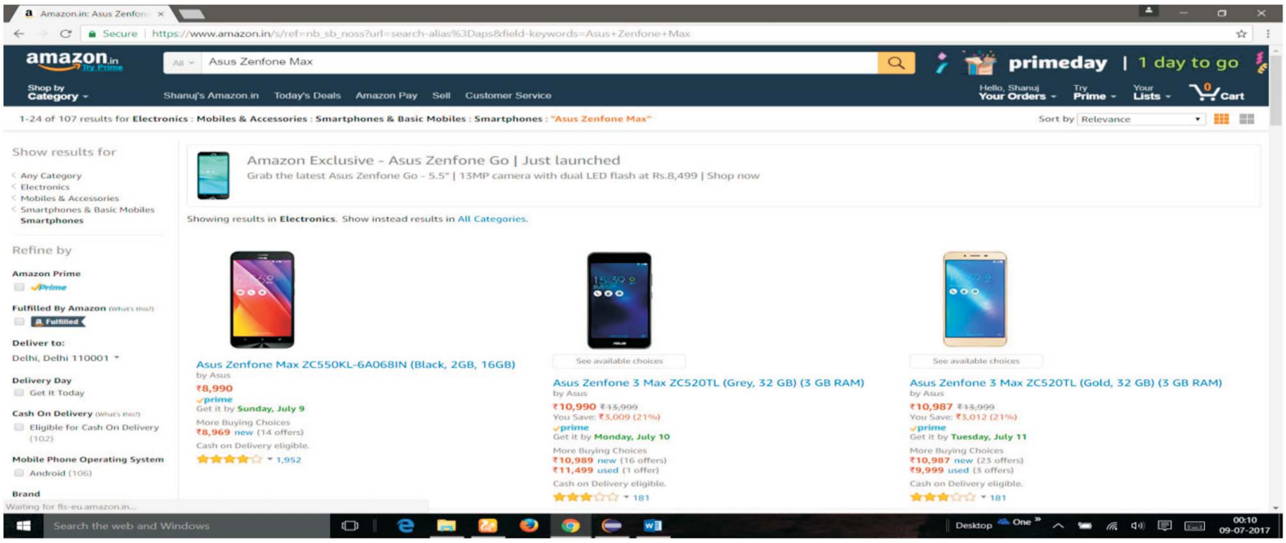
Figure 6. Execution of login test case of amazon.in.



```
package pageObjects;

import java.util.ArrayList;

public class SearchProduct
{

    public static void search(WebDriver driver,String text,String Add_details,String site) throws InterruptedException
    {
        if(site.equals("amazon"))
        {
            driver.findElement(By.id("twotabsearchtextbox")).sendKeys(text);
            driver.findElement(By.className("nav-input")).click();                          //Click on search button
            Thread.sleep(2000);
            driver.findElement(By.partialLinkText(Add_details)).click();
            Thread.sleep(2000);
            ArrayList<String> tabs = new ArrayList<String> (driver.getWindowHandles());
            //Close the current tab
            driver.close();
            //Switch to New Tab
            driver.switchTo().window(tabs.get(1));
        }
```

No consoles to display at this time.

Figure 7. Sprint 2 - automation of search product functionality using Page Object Model for amazon.in.



1-24 of 107 results for Electronics : Mobiles & Accessories : Smartphones & Basic Mobiles : Smartphones : "Asus Zenfone Max"

Show results for
- Any Category
- Electronics
- Mobiles & Accessories
- Smartphones & Basic Mobiles
- Smartphones

Refine by

Amazon Prime
□ Prime

Fulfilled By Amazon (What's this?)
□ Fulfilled

Deliver to:
Delhi, Delhi 110001

Delivery Day
□ Get It Today

Cash On Delivery (What's this?)
□ Eligible for Cash On Delivery (102)

Mobile Phone Operating System
□ Android (106)

Brand

Waiting for fls-eu.amazon.in...

Amazon Exclusive - Asus Zenfone Go | Just launched
Grab the latest Asus Zenfone Go - 5.5" | 13MP camera with dual LED flash at Rs.8,499 | Shop now

Showing results in Electronics. Show instead results in All Categories.

Asus Zenfone Max ZC550KL-6A068IN (Black, 2GB, 16GB)
by Asus
₹8,990
prime
Get it by Sunday, July 9
More Buying Choices
₹8,969 new (14 offers)
Cash on Delivery eligible.
★★★★☆ 1,952

See available choices
Asus Zenfone 3 Max ZC520TL (Grey, 32 GB) (3 GB RAM)
by Asus
₹10,990 ~~₹13,999~~
You Save: ₹3,009 (21%)
prime
Get it by Monday, July 10
More Buying Choices
₹10,989 new (16 offers)
₹11,499 used (1 offer)
Cash on Delivery eligible.
★★★☆☆ 181

See available choices
Asus Zenfone 3 Max ZC520TL (Gold, 32 GB) (3 GB RAM)
by Asus
₹10,987 ~~₹13,999~~
You Save: ₹3,012 (21%)
prime
Get it by Tuesday, July 11
More Buying Choices
₹10,987 new (23 offers)
₹9,999 used (3 offers)
Cash on Delivery eligible.
★★★☆☆ 181

Figure 8. Sprint 2 - execution of search product test case of amazon.in.

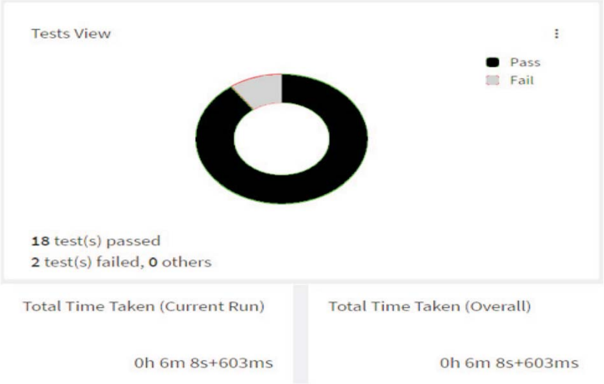
ciency is improved remarkably as we were able to run our regression test suite right from the beginning of sprint 2 which is very much useful in the agile software development projects. The use of POM makes this framework highly reusable and maintainable as we have used the same framework for two different ecommerce websites, and lastly testing cycle time also been reduced considerably as shown in Table 2 which addressed our RQ1. We saved around thrice the test execution time by automating our regression tests using RHTAF. Our insprint test automation strategy proved to be very useful in applying RHTAF right from the sprint 1 and we had the subset of regression suite ready that was executed before the beginning of the next sprint, this addressed our RQ2. Moreover, we also integrated the result reporting facility in our framework with the help of which entire test summary is readily available after test case execution and also graphical representation of test result summary is also displayed to have a quick view on the pass/fail status of the test cases as shown in Figure 9 for amazon.in and Figure 10 for flipkart.com.

Table 2. Total time taken manual vs. automated test case execution

E-commerce website	Total time taken (manual test execution)	Total time taken (automated test execution using RHTAF)
Amazon	15 m 36 sec	6 m 8 sec
Flipkart	10 m 10 sec	3 m 42 sec

5. Conclusion and Future Work

This paper makes two important contributions. First,

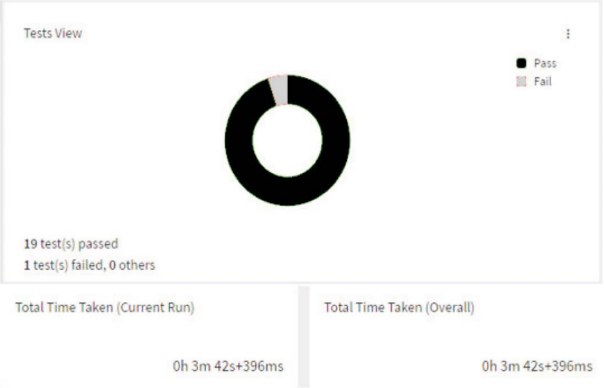


18 test(s) passed
2 test(s) failed, 0 others

Total Time Taken (Current Run) 0h 6m 8s+603ms

Total Time Taken (Overall) 0h 6m 8s+603ms

Figure 9. Graph showing test result summary along with total test execution time for amazon.in.



19 test(s) passed
1 test(s) failed, 0 others

Total Time Taken (Current Run) 0h 3m 42s+396ms

Total Time Taken (Overall) 0h 3m 42s+396ms

Figure 10. Graph showing test result summary along with total test execution time for flipkart.in.

we implemented the reusable hybrid test automation framework using page object model for web based agile projects. Use of hybrid i.e. combination of keyword driven and data driven frameworks along with the page object model concept makes this framework reusable and maintainable and efficient for automating regression tests.

Secondly, we demonstrated the application of this framework on two famous e-commerce platforms i.e. amazon.in and flipkart.com in real agile environment within sprint test automation strategy. We demonstrated the effectiveness of our RHTAF while automating regression tests for both of these websites right from the very first sprint and gradually build up our regression suite which proves very useful in faster test execution and finding defects right from the beginning of the sprint which in turn improves the overall quality of the application under test. In the future research, we will try to develop this framework for other IDE's like Microsoft Visual Studio and will also try to explore other frameworks like behavioral driven framework (BDD) for automating test cases in agile software development environment.

References

[1] Dyba, T. and Dingsoyr, T., "Empirical Studies of Agile Software Development: a Systematic Review," *Inform. Softw. Technol.* (2008). doi: 10.1016/j.infsof.2008.01.006

[2] Tyagi, S., Sibal, R. and Suri, B., "Adopting Test Automation on Agile Development Projects: a Grounded Theory Study of Indian Software Organizations," In: Baumeister H., Lichter H., RiebischM. (eds) Agile Processes in Software Engineering and Extreme Programming. XP 2017. Lecture Notesin Business Information Processing, Vol. 283. Springer, Cham. (2017). doi: 10.1007/978-3-319-57633-6_12

[3] Information on http://seleniumdotnet.blogspot.in/ Last accessed 08 Oct 2016.

[4] Information on http://www.seleniumhq.org. Last accessed 12 Oct 2016.

[5] Crispin, L. and Gregory, J., *Agile Testing: a Practical Guide for Testers and Agile Teams*, Addison-Wesley (2009). ISBN 0-321-53446-8

[6] Collins, E. F. and de Lucena, V. F., "Software Test Automation Practices in Agile Development Environment: an Industry Experience Report," 7th International Workshop on Automation of Software Test (AST), Zurich, 2012, pp. 57–63 (2012). doi: 10.1109/IWAST.2012.6228991

[7] Karhu, K., Repo, T., Taipale, O. and Smolander, K., "Empirical Observations on Software Testing Automation," Proceedings of the 2nd International Conference on Software Testing, Verification, and Validation (ICST 2009), Denver, Colo, USA, pp. 201–209 (2009). doi: 10.1109/ICST.2009.16

[8] Deak, A., "A Comparative Study of Tester's Motivation in traditional and Agile Software Development," Product – Focused Software Process Improvement, pp. 1–16 (2014). doi: 10.1007/978-3-319-13835-0_1

[9] Rafi, D. M., Reddy, K., Moses, K., Petersen, K. and Mäntylä, M. V., "Benefits and Limitations of Automated Software Testing: Systematic Literature Review and Practitioner Survey," 7th International Workshop on Automation of Software Test (AST), Zurich, pp. 36–42 (2012). doi: 10.1109/IWAST.2012.6228988

[10] Puleio, M., "How Not to Do Agile Testing," Proceedings of the Conference on AGILE 2006 (AGILE '06). IEEE Computer Society, Washington, DC, USA, pp. 305–31 (2006) doi: 10.1109/AGILE.2006.34

[11] Hui, J., Yuqing, L., Pei, L., Shuhang, G. and Jing, G., "LKDT: a Keyword-Driven Based Distributed Test Framework," International Conference on Computer Science and Software Engineering, Wuhan, Hubei, pp. 719–722 (2008). doi: 10.1109/CSSE.2008.1036

[12] De Castro, A. M., Macedo, G. A., Collins, E. F. and Dias-Neto, A. C., "Extension of Selenium RC Tool to Perform Automated Testing with Databases in Web Applications," 8th International Workshop on Automation of Software Test (AST), San Francisco, CA, pp. 125–131 (2013). doi: 10.1109/IWAST.2013.6595803

Manuscript Received: Nov. 13, 2017
Accepted: Dec. 23, 2017