EasyChair Preprint

№ 1646

A swift approach to perform automated testing in Scrum

Aleena Azhar and Arfan Jaffar

October 13, 2019

A swift approach to perform automated testing in Scrum

Aleena Azhar[1]
Department of computer science
Superior University
Aleenazhar94@gmail.com

Dr. Arfan Jaffar[2]
HOD, Software Engineering
Superior University
Lahore, Pakistan
irfan.jaffer@gmail.com

***Abstract*—As the time of development and delivery of software applications is shrinking day by day, the quality of software is also being compromised. No one knows how much testing should be done, however it becomes quite easy to estimate days required to perform software testing. Any activity is automated to save time and achieve more in short time. However, when software testing is automated then the situation is inverse. Yes! Test automation takes time initially and gradually generates a return on investment. It becomes an issue when software development companies are following agile models like Scrum where the time to deliver is maximum 30 days and quality assurance activities get insufficient time to perform manual testing even. There it is a big challenge to perform test automation for achieving even better quality. As more software companies are adopting Scrum these days, so dealing effectively with test automation is becoming a major issue that needs attention. Therefore, this research activity will be done in software testing industry to find possible ways through which automated testing could be performed in Scrum model and ensure software quality without delaying software delivery. In This research will enable software quality assurance and test engineers operating in Scrum environment to analyze testing requirements, generate test scripts, log defects and present test results to maximize software quality before it is shipped. As of now this research will focus on lifting automated testing scope in Scrum only, however this study could be used in future to address same problem in other Agile models like Kanban, XP, Lean Development and Scrumban.**

Keywords—Return on investment (ROI); Scrum; Kanban; Extreme Programming (XP); Scrumban styling;

I. INTRODUCTION

Test Automation is not new but its practical adaptation has started recently. There exists sufficient information that explains importance of automated testing, comparisons between automated and manual testing, ways to generate ROI through automated testing, different frameworks of automated testing, automated testing of web, mobile and desktop applications. Today companies are investing in automated testing like never before just because it is believed that if testing will be automated, a lot of time could be saved. The ground reality is different from expectations because automated testing does not have an instant power engine. Initially it takes at least six months to get started with automated testing and approximately a year to get proper benefit from it. Software testing in traditional models like Waterfall is comparatively less challenging because the testing teams get sufficient time to ensure quality. Time constraint in traditional models is very low that's why the companies that are still following Waterfall, V-Model or Incremental approach get sufficient time in which they manually test software and write test scripts. Test automation does not contain script development only, but there are many other tasks like development, enhancement and maintenance of test automation framework, test case development, test scenarios identification, identification of test cases that should be automated on first priority and the ones that are most eligible candidates for being automated, conversion of selected test cases into automated test scripts, debugging of newly written test scripts, execution of test scripts in controlled environment and then generation of test results and logs. When we know the number of automated testing tasks is big and complex, it becomes almost impossible to do it in models like Scrum because of tight deadlines, pressure and scarcity of time. The trend of automated testing seems more practical in traditional models but in modern development models like Scrum, the shortage of time is a big problem that needs to be addressed.

This research activity would be conducted by actually studying the experiences of three famous software development companies of Pakistan that are currently performing automated testing in Scrum. NETSOL Technologies, Tekxel and Confiz not only using Scrum model to develop their products but also they are CMMI certified organizations. The information gathered from these companies will be amalgamated to produce a refined and robust solution to the subjected issue. The output of this research would be shared with testing teams of these companies' academic practitioners that are working on Scrum and test automation. It is aimed that the output of this research would allow software testing teams and test professionals to adopt more quick (swift) approach in developing automated test scripts when they are operating in Scrum.

II. RELATED WORK

A lot of work has been done with an aim to develop an affective framework, either for web applications solely or for desktop applications. The techniques used in developing any framework carry some benefits and mixing up multiple techniques into one package could produce a healthy automated testing framework.

Now a day's software development companies are heavily dependent on automated testing and implement different test automation framework [22]. Keyword test automation framework is an enhanced form of traditional data driven framework [27]. Automated Testing is an advanced way for ensuring quality of software under testing. Every organization

has special and unique reason for automating the software testing process but many reason are common between different organizations. The common reasons are; automated test tools can execute test cases, the framework enhances the ease of managing test scripts, a lot of work can be taken from single testing tool [26]. Automated Testing Framework can be considered as a standard for software automated testing, which provides reusable foundation for modules under testing and easy management of automated test functions [27].

Srivastava et al. 2017 [1] have discussed the workflow of Scrum by highlighting its importance and how it is taking over modern software development companies to meet targets of early delivery. An in depth comparison of Scrum with other development models has been that demonstrates why Scrum should be adopted by software development teams. Along with the benefits of generic Scrum model, some disadvantages can cause problems. A solution has been proposed to mitigate the risks present in generic Scrum approach and achieve maximum output.

Kumar Gupta et al. 2017 [2] provides an in-depth over view of challenges faced by companies when they make transition towards Scrum. This research highlights many challenges faced during different phases of SDLC. Issues related to software testing in Scrum have also been discussed that provides an enlightenment about attention should be given to testing phase also. The paper also proposed the ways through which success can be achieved following Scrum.

Nidagundia et al. 2017 [3] have written a paper that focuses on implementation lean canvas model in scrum software testing. A very brilliant mapping has been done between Scrum events and Lean canvas model. This model targets to uplift overall software testing scheme in Scrum.

Farrukh Latif Butt 2017 [4] addressed about software testing problems under agile approach. Unless these problems come up with a "systematic techniques" at certain stage of agile process. In agile, scrum methodology, that is an iterative method. Where it begins with sprint scheduling meetings and ended with different reviews. In this paper the suggested systematic order of analyze the testing progress. Which leads testing teams to end with correct testing techniques. The proposed algorithm boost up testing process in API of the product during development. To put the API test process as a foam of Unit testing just before to make it alpha build. So unit based testing performance on different APIs that causes efficient outcomes by finding logic bugs by "Bug tracking systems". That included (TFS) team foundation server, VersionOne, Flawtrack, played an actual role in scrum developing process.

Andrea Arcuri 2017 [8] reported the current research impact of software engineering in academic and industry. Author conveyed the post doctor working experience in software engineering research based projects and five year industrial worked experience too.

Shurti Sharma at el. 2016 [15] have described the current emerging state of Scrum by industrial surveys. Well there are many development models like Scrum, XP, Crystal, FDD (Feature driven development), DSDM (Dynamic System Development Method) but Scrum evolution is increasing day by day at development and testing level in research community. Traditional technique is no more efficient today because it failed to have all following aspects like flexibility, rapid delivery of product over documentation, customer satisfaction and fast acceptance of change in software. This paper combined literature and industrial Surveys Results. From 2010 to 2014 reports and surveys author analyzed and compared the use of Scrum and adoption of its methodologies by many organizations. So, by 30 related papers critically mapped out the today's crucial need of that model. Author targeted on main two research repositories IEEE and ACM and illustrate that widely technique of Scrum methodologies rather than other agile methodologies.

Kamini (Simi) Bajaj 2018[20] has stated that, the test data paly vital role for manual and automate testing both. Test Data Management (TDM) act as significant element for analysis though automate testing proved to effectively test concentrated data applications. The predominantly focus TDM automation testing improved the quality and shrink the cost along with time. A developed test automation framework introduced called TAFPro. It is suggested for IBM test automation tools. It also mentioned the growth in productivity and reduction in submission and life cycle costs. The developing organizations moving towards open source tool. Main purpose is to make the TAFPro tool open source and independent for JAVA based tools that supported selenium magnificently for TDM.

Srrinidhi, Sharaniya et al 2017 [22] have proposed a test automation framework for testing of web applications. The framework is based on keyword driven methodology with an aim to provide ease to software testers by separating the coding from test cases and test steps. It has been proposed that software testers do not necessarily needs coding knowledge to be a part of automated testing team. It has also been reported that using a framework based on keyword methodology helps in reducing the maintenance cost overall. The framework proposed in this research is capable of automating web applications only. Selenium, which is an open source automated testing tool for web applications, has been used for the purpose.

Sabastain et al. 2017 [23] have stated that, there are many ways to develop an automated testing framework to automate GUI's of web applications. They have concentrated on the development of a hybrid test automation framework which incorporates the use of high level programming with implementation of object oriented design concepts. They have declared it as a hybrid framework because it uses the combination of data driven and library architecture methodologies. External database has been used to provide the test data. The test cases can be repeated differently in a loop to handle different test variations. The researchers have developed a framework which is independent from any technology and platform. The important components of the framework are Main Class, Object Repository, Reusable Functions, Properties File, Custom Exceptions and Test Scripts. The limitation of this framework is that it is also designed to entertain web applications.

Aziz 2017 [24] has reported that keyword driven testing framework helps in reducing maintenance work for agile teams through the ease of updating test scripts. It has been suggested that, the total count of keywords in a keyword base test automation framework should be less than the total number of test cases.

Alotaibi et al. 2017 [25] have proposed a need of test automation framework for testing mobile applications. Their concentration is also on the target to reduce effort and speed up the testing process. The focus is to on the applications which are responsive in nature and can be opened from mobile and desktop computers. So, it means, testing of an exactly same application in two entirely different environments would be a tricky task. The proposed automated testing framework uses data driven testing methodology and Appium test library. The mechanism of this framework relies upon an Appium Server which interacts with Application under Testing and gives a call to Appium Test Library, which further gives a call to Test Driver. Test Driver is responsible for the execution of test cases under consideration. The test data is supplied to the DB connected with the server. Overall the applications for which the test automation framework has been proposed are web applications which are compatible with mobile browsers and desktop browsers.

Da Zun, et al. 2016 [27] have also used a keyword driven methodology for development of test automation framework for mobile applications. Since testing of mobile applications is not under concentration, therefore an insight of keyword engine has been grabbed from this article. The framework has been designed into three layers. Where each layer contains different components, which contributes in the execution of test scripts. The three layers are presentation layer, business layers and data layer. Test case input and test results lie at presentation layer, test script parser and test scripts executer lie at business layer. Test Data storage is present at data layer. The division of framework components into different layers enforces the 'separation of concerns' concept. This research was aimed for developing such an automated testing framework which is capable of testing mobile applications from different platforms like Android and IOS.

Nirmala, et al. 2015 [28] have laid the foundation of test automation framework on four aspects. The four aspects are test strategy, test case generation, test execution and test evaluation. It has been suggested that a framework is a mixture of several strategies like programming, test data injections, modularity, standards, methods, conventions, system hierarchies, and coverage. This framework is aimed to automate the generation of test cases which will be used as test data or for execution of test scripts. The major time is consumed in the development of these test cases therefore it was required there must be a solution for the development of test cases first. The framework is also focused on the generation of test suites which would ultimately increase work efficiency. This framework can be used to execute functional and nonfunctional test cases. The framework developed by the researcher has emphasized on the effectiveness of the quality control mechanism.

Bhondoka et al. 2015 [29] have presented the information related to hybrid test automation framework for web applications. Just like above discussed research work, this effort has been done using keyword driven approach. The basic reason behind adopting keyword approach is same i.e. to increase reusability and speed of test execution. Briefly described here is the simple flow of this smart framework which begins from getting an input from excel sheet? The input is extraction of keywords, on basis of which functions are called accordingly. The functions extract x-path for the object present in web application. Then application is launched and test scripts are executed. After test execution, reports are generated. The power of this framework is also limited to test the web applications. Selenium has been used and suggested for automated testing with this framework. With this approach, accuracy of testing and test coverage increases. This framework is not dependent on any language. The basic principles of the framework should be followed to develop a product like it with any programming language.

Mohan Das et al. 2014 [30] researched about challenges that are faced automation of Graphical User Interface. Software Testing is becoming more challenging as the complexity of application is increasing. There is a great technological diversity coming into view because similar applications are being developed for different operating systems and environments. The other challenge is the rapid application development process like agile, where quick deliveries are to be done, which requires even quicker testing speed without compromising the quality. The challenges under consideration can be easily addressed by the implementation of automated testing system. GUIRobo is an automated testing which has been used for this research activity. Software Testers give an attempt to automate GUI on first hand. Because they have weak programming skills they use record and playback feature offered by test tool. The biggest challenge with this approach is that, as soon as there is a change in GUI of application under testing, the recorded scripts will not be able to playback. Thus, resulting in the wastage of whole effort.

Sabina et al. 2014 [31] have proposed an automated testing framework which is suited for application with complex business requirements and numerous GUIs. Automated Testing Framework has been considered as the representation of the use of software to control the test execution, comparison of actual and expected results, setting up test pre-requisites and generation of testing reports or logs. The object repository of the framework developed here is not based on Page Object model approach. Following this approach, it is much easier for test engineers to directly update the affected objects without disturbing the unchanged objects. But the challenge is that, as soon as the number of web pages will increase, it would be difficult to maintain them over the period.

Kumar et al. 2013 [33] have proposed a test automation framework which automated unit testing of windows applications. Unit testing is known to be done by software programmers. The article is based on development of unit testing framework for testing of windows applications. There are some areas in windows applications which can be easily tested without manual input by software testers. Because of it, overall quality of the application is increased.

III. METHODOLOGY

To present a justifiable solution for subjected problem, enhancement features have been added to hybrid test automation framework, which acts as a vital source for solving the problem of switching between different versions of same application during test execution.

1. ApplicationWrapper
   A feature which checks what type of application is under testing. This check is performed when test script runs.
2. Application Decision
   A feature which decides to choose the action function(s) to perform based on type of application which is under test and the class of objects used in test script.
3. Object Extractor
   A feature which reads the objects of application under testing and then populates required properties of those objects to an excel file.
4. Use of TestComplete
   TestComplete has been used as automated testing IDE for development of the required hybrid framework because of it' capability to test web and desktop applications through same project suite. This capability was rolled out in TestComplete 12 Edition. We have used TestComplete 12.03 for this research activity.

The goal is to provide a complete end-to-end automation solution for AUT. A combination of two or more frameworks approaches makes up a Hybrid Framework. This research is also based on same technique; pulling the strengths of different frameworks and trying to mitigate their weaknesses so that a strong independent and common framework could be developed which is capable of handling web and desktop applications together with common functions. Because of this effort, it would be very convenient for software testing teams to manage test scripts for version based applications. TestComplete automated testing tool allows one project working for any type of application. Most of the automated testing tools in market, either support web applications or only desktop applications. Few expensive tools, provide the support for all types of applications, but all created and maintained in a separate project suite. TestComplete is the only tool which provides the facility to operate and test any type of application through same common project. Suppose there is a software "ABC" which is available in two versions, i.e. a web application (run through browser) and a desktop application (run through operating system). TestComplete's one project suite will be able to execute test scripts for both applications one after another, relying on this "App-Wrapper" function, which intelligently notifies TestComplete about nature of application used in current test script.

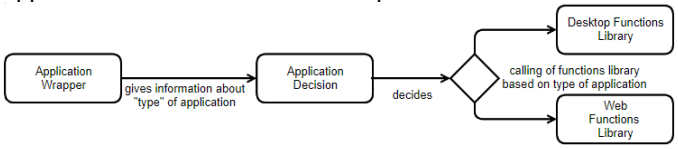

Figure1: Application Decision

This function is developed for deciding, which actions library testing tool should jump in to perform a specific action, based on the "type of application" detected at layer one. For example, if Application Wrapper has detected the type of application as "web" then Application Decision function will give a call to web functions library. Or if "desktop" type of application has been detected then testing tool should give a call to desktop functions library.

Test Data Sheet:

This sheet is used as source of test data. The framework uses Data Source Manager to fetch test data for test execution. This sheet contains following columns:

Column 1: Serial Number (Sr_Nbr)

This column contains the serial number of test data row.

Column 2: JIRA ID

If test cases have been developed through JIRA, this column can be used to keep Jira ID of the test case.

Column 3: Test Case ID (TC_ID)

Test Case IDs are written in this column so that main driver can directly jump on the steps of the test case which is required to be executed.

Column 4: Test Case Step ID (TS_ID)

Test Case IDs are written in this column so that main driver can directly jump on the steps of the test case which is required to be executed.

Column 5: Test Step Description (TS_DESC)

This column contains the description of test steps of test cases.

Column 6: Data Variation (DV)

This column contains actual test data which should be entered, selected or targeted during test execution.

Test Data Sheet					
Jira_ID	TC_ID	TS_ID	TS_Desc	DV_1	DV_2
SK-1	TC_Login_001	1	Enter value in "User Name text field.	Admin	Admin
SK-1	TC_Login_001	2	Enter value in "Password" field.	admin123	admin123
SK-1	TC_Login_001	3	Click on "Login" button.		
SK-2	TC_AddEmployee_002	1	Hover mouse to "PIM" menu.		
SK-2	TC_AddEmployee_002	2	Click on "Add Employee" option.		
SK-2	TC_AddEmployee_002	3	Enter value in "First Name" text field.	Aleena	ali
SK-2	TC_AddEmployee_002	4	Enter value in "Middle Name" text field.	Azhar	na
SK-2	TC_AddEmployee_002	5	Enter value in "Last Name" text field.	Khan	azhar
SK-2	TC_AddEmployee_002	6	Click on "Save" button.		
SK-3	TC_PatientRegistration_Verification_003	1	Enter MRNO to Search "Registerd Patient".		

Figure 2: Test Data Sheet

The next half test data sheet that store the data in excel sheet firstly.

Test Data Sheet					
Jira_ID	TC_ID	TS_ID	TS_Desc	DV_1	DV_2
SK-1	TC_Login_001	1	Enter value in "User Name text field.	Admin	Admin
SK-1	TC_Login_001	2	Enter value in "Password" field.	admin123	admin123
SK-3	TC_PatientRegistration_Verification_003	6	Verify value in "Last Name" text field.		
SK-3	TC_PatientRegistration_Verification_003	7	Verify value in "Father Name" text field.		
SK-3	TC_PatientRegistration_Verification_003	8	Verify value from "Blood Group" list.	A+	
SK-3	TC_PatientRegistration_Verification_003	9	Verify value in "Date of Birth" text field.		
SK-3	TC_PatientRegistration_Verification_003	10	Verify value from "Age" text field.		
SK-3	TC_PatientRegistration_Verification_003	11	Verify value from "Gender" list.	Female	
SK-3	TC_PatientRegistration_Verification_003	12	Verify value from "Marital Status" list.	Single	
SK-3	TC_PatientRegistration_Verification_003	13	Verify value from "Religion" drop-down.	Islam	
SK-3	TC_PatientRegistration_Verification_003	14	Verify value from "Occupation" drop-down.	student	
SK-3	TC_PatientRegistration_Verification_003	15	Verify value from "Birth tehsil/City" list.	Lahore	
SK-3	TC_PatientRegistration_Verification_003	16	Verify value in "NIC No" text field.		
SK-3	TC_PatientRegistration_Verification_003	17	Verify value in "Contact" text field.	03204445556	
SK-3	TC_PatientRegistration_Verification_003	18	Verify value from "Relation" list.		
SK-3	TC_PatientRegistration_Verification_003	19	Verify value in "Address" text field.		

Figure 3: Test Data Sheet 2

IV. FRAMEWORK SUPPORT LIBRARY

This layer includes all the functions to be used by framework itself. Purpose of this layer is to separate framework functions and to take out the scripter from hassle of managing so many other script units which are not directly required for test execution. This layer is empowered to deal with multiple events occurring during automated test execution. This layer is

further divided into three components. Details of all components and their sub-components are given below:

1. Configurations
2. Constants
3. Support Functions

1. Configurations:

How it works?

Software Testers can store URL of their web application and path of their desktop application in a "Config Variables" excel file. This file is developed for the ease of users to set all properties of their testing environment so that they don't have to do it again and again. List of properties required to be set in advance are:

Name	Value
Desktop App Auto Launch	Yes
Desktop Application Path	C:\Program Files\Desktop App
Desktop Application Process	App.exe
Desktop Application Server IP	10.39.100.55
DB Connection String	
Web App Auto Launch	Yes
Browser	Chrome
Browser Launch Path	C:\Program Files\Google\Chrome.exe
Browser Process Name	Chrome.exe
Web Application URL	www.application.com

Table 1: Configurations File Format:

2. Constants:

How it works?

To increase the reusability factor in all test scripts, "hard coded" values or test data must be avoided. To mitigate the risk of writing hard coded data in test scripts, Constants unit file would serve the purpose.

For example, there is a variable with same value, which needs to be called in multiple test scripts. Instead of creating this variable again and again in every test script, software testers can declare it in "Constants" script unit and assign it your desired value. Just call this variable in your all test scripts and there will be no need of variable declaration again and again. Later, when there is an update in the value of that variable, it would be updated at a single spot i.e. Constants file. Testers do not have to open all test scripts separately and update variable value in all those test scripts.

3. Support Functions

Framework Support Library contains different functionalities clubbed together in "Support Functions" group which will facilitate the software testing teams to save their time for setting up testing environment, before every test execution.

1. Support Functions
2. Network Availability
3. Database Availability
4. Events Handler
5. Object Extractor
6. Descriptive Programming

A. Data Source Manager

Management of Test Data is the responsibility of this layer. The lists of functions present at this layer are:

DATA SOURCE ACTIONS

1. Get Driver Instance (Data Source Type, Data Source Name)

This function is responsible for creating a connection of test data source (excel or SQL) with TestComplete. This function creates a "Driver Object" which is an actual connection of test data source with TestComplete. This function takes two parameters:

1) Excel Work Book Name or SQL Table Name
2) Excel Sheet Name

2. Set Source Data (Driver Object)

This function iterates through test data source to capture all rows of test data and store them in a logical memory called as "Record Set". This function requires only one parameter i.e. the driver object which was created through function "Get Driver Instance".

3. Get Item Value (Record Set, Column Name, Row ID)

This function iterates through test data sheet to return the exact value required by test step executor. This function will be used in every test script for fetching the data from external source. This function requires three parameters 1. Record Set 2. Column Name 3. Row ID

4. Get Row Count (Record Set)

This function returns the total count of rows present in test data sheet. It takes "Record Set" as a single parameter.

5. Get Column Count (Record Set)

This function returns the total count of columns present in test data sheet. It takes "Record Set" as a single parameter.

6. Get Item Row Index (Record Set, Column Name, Value)

This function returns row index of the any value given as an input parameter. It takes 1) Record Set 2) Column Name of value 3) Actual Value as parameters.

7. Get Column Index (Record Set, Column Name)

This function returns column index of the any value given as an input parameter. It takes 1. Record Set 2. Column Name of value as parameters.

B. Object Repository

Object Repository is a collection of objects. Object Repository works as container for the objects. TestComplete have a feature for Object Repository known as Name Mapping.

C. Name Mapping

In TestComplete, Name Mapping lists all objects in hierarchy. Name Map consist of two containers Mapped objects and Aliases.

D. Mapped objects

Mapped objects allow to map/store the AUT objects with their original hierarchy.

E. Aliases

Aliases allow us to name objects in simple user defined way. Aliases is used to reduce the complex hierarchy as well. This framework will use name map for storing object. In Name Map, objects will be captured in static way with Aliases name(s) as displayed on the screen for objects.

F. Objects Naming Convention

This section describes how objects will be captured in the Aliases container of Name Map. The objects will be captured with names exactly the way they are present on screen. Each Screen will be the next node after the application and screen will have all its child objects in it. We are following the below rule for mapping object in Name Map so that update of Name Map does not affect the tests. Following criteria are followed for mapping objects in Name Mapping.

1. ["App_Name"]["ScreenName_Screen"]["UILabelName_ControlType"]

Example: ["Facebook"]["Home"]["Status_TextBox"]

2. Each screen name will be followed by "_Screen" suffix and each object by "_ClrClassName" property as suffix.
3. Two properties will be added in object Name Map

1) ClrClassName
2) WPFControlName.

If these properties are not unique then we can add any other property.

V. REPORTING

After testing first test script the summery report is generating. The resultive summery shows time efficiency and improve the testing ability of testing web pages

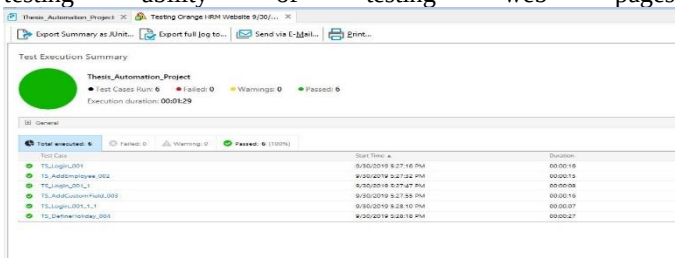

Figure 1: TestComplete Test summery1

Another test script running summery is also attached here. It simply describe the efficiency of time and improved version of software quality by automation testing.

Test Data Sheet					
Jira_ID	TC_ID	TS_ID	TS_Desc	DV_1	DV_2
SK-1	TC_Login_001	1	Enter value in "User Name text field.	Admin	
SK-1	TC_Login_001	2	Enter value in "Password" field.	admin123	
SK-2	TC_AddEmployee_002	1	Hover mouse to "PIM" menu.		
SK-2	TC_AddEmployee_002	2	Click on "Add Employee" option.		
SK-2	TC_AddEmployee_002	3	Enter value in "First Name" text field.	Aleena	
SK-2	TC_AddEmployee_002	4	Enter value in "Middle Name" text field.	Azhar	
SK-2	TC_AddEmployee_002	5	Enter value in "Last Name" text field.	Khan	
SK-2	TC_AddEmployee_002	6	Click on "Save" button.		
SK-3	TC_AddCustomField_003	1	Hover mouse to "PIM" menu.		
SK-3	TC_AddCustomField_003	2	Click on "Custom Field" option.		
SK-3	TC_AddCustomField_003	3	Enter value in "Field Name" text field.	Edu	
SK-3	TC_AddCustomField_003	4	Select Value From "Screen" DropDown.	Job	
SK-3	TC_AddCustomField_003	5	Select Value From "Type" DropDown.	Text or Number	
SK-3	TC_AddCustomField_003	6	Click on "Save" button.		
SK-4	TC_DefineHoliday_004	1	Hover mouse to "Leave" menu.		
SK-4	TC_DefineHoliday_004	2	Click on "Holiday" option.		
SK-4	TC_DefineHoliday_004	3	Enter value in "Name" text field.	Winter Holiday	
SK-4	TC_DefineHoliday_004	4	Select Value From "Date" Field.	2020-01-01	
SK-4	TC_DefineHoliday_004	5	Select Value From "FullDay/HalfDay" DropDown.	Full Day	
SK-4	TC_DefineHoliday_004	6	Click on "Save" button.		

Figure 5: TestComplete Test Script

TestComplete has a built-in feature to generate a test log after every test script is executed. However, the built-in test log was not much justifiable, as higher management needs see more detailed and comprehensive reports. Therefore, custom functions were developed to produce high quality and detailed reports which shows the results of every executed test sequence, test case and test step. Below given is the screen shot of detailed test log generated through this framework in TestComplete.

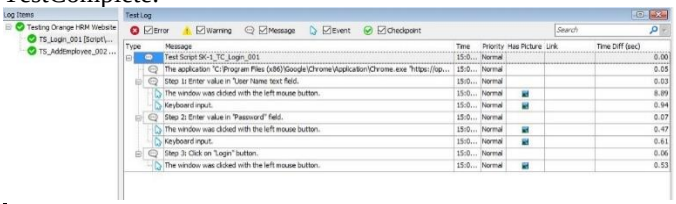

Figure 6: TestComplete Test Log 1

The screenshot is attached here:

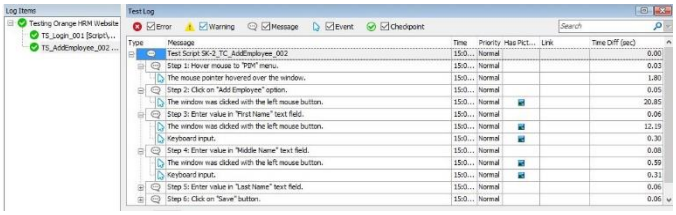

Figure 7: TestComplete Test Log 2

TEST EXECUTION WITH MULTIPLE TYPES OF APPLICATIONS

The mechanism which controls the execution of two different types of applications within a single test execution has been described below.

1. When the test execution begins, Application Wrapper detects the type of application which is under testing.
2. In the mean while Main Driver is busy in regulating the entire process of test steps execution, Application Wrapper continues to check the type of application which is required in upcoming test steps.
3. As soon as any test case come which should be executed on another type of application, Application Wrapper gives an instruction to Main Driver about another type of application.
4. Rest of the steps will be repeated as described above.
5. It can be concluded that, the proposed test automation framework is capable of switching between two different types of applications during the on-going test execution.

VI. ACKNOWLEDGMENT

The acknowledgment surpasses the encouragement of all those who helped and motivated me through the research activity. First and foremost I would like to thanks Allah Almighty who gave me the ability to work and to face all the difficulties that encountered during this long journey. There were many supportive people who helped me in this research including my friends, teachers and my family. As a final point I did like to show gratitude to my family members and teachers who supported me during some difficult times.

REFERENCES

[1] S. Bhardwaj, S. Saraswat and Apoorva Srivastava, "SCRUM Model for Agile Methodology," International Conference on Computing, Communication and Automation, pp. 864-869, 2017.

[2] P. M. A. K. C. Rajeev Kumar Gupta, "Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey A case study in a multi-location legacy software product development team," ACM, pp. 147-156, 2017.

[3] L. N. Padmaraj Nidagundia, "Introducing Lean Canvas Model Adaptation in the Scrum Software Testing," Elsevier B.V., vol. 104, pp. 97-103, 2017.

[4] S. N. B. S. M. J. A. S. Farrukh Latif Butt, "Optimized Order of Software Testing Techniques in Agile Process – A Systematic Approach," International Journal of Advanced Computer Science and Applications,, vol. 8, no. 1, pp. 347-353, 2017.

[5] S. E. D. S. K. L. KristianWiklund, "Impediments for software test automation: A systematic literature review," Wiley , pp. 1-20, 2017.

[6] J. L. H. Jie, "Industrial Case Study of Transition from V-Model into Agile SCRUM in Embedded Software Testing Industries," ACM SIGSOFT, vol. 41, pp. 1-3, March 2016.

[7] M. K. Lech Madeyski, "Continuous Test-Driven Development: A Preliminary Empirical Evaluation Using Agile Experimentation in Industrial Settings," Springer International Publishing AG , pp. 105-118, 2018.

[8] A. Arcuri, "An experience report on applying software testing academic results in industry: we need usable automated test generation," Springer Science+Business Media, LLC , 2017.

[9] A. F. d. P. W. L. d. S. S. M. d. S. F. P. a. L. F. P. Pedro Lopes de Souza, "Combining Behaviour-Driven Development with Scrum for Software Development in the Education Domain," ICEIS, vol. 2, pp. 449-458, 2017.

[10] F. E. Vahid Garousi, "Test Automation Not Just for Test Execution," IEEE, 2016.

[11] T. S. T. M. a. I. K. Geir K. Hanssen Børge Haugset, "Quality Assurance in Scrum Applied to Safety Critical Software," The Author(s), p. 92–103, 2016.

[12] Y. P. Shaik Mohammad Shahabuddin, "Integration Testing Prior to Unit Testing: A Paradigm Shift in Object Oriented Software Testing of Agile Software Engineering," Indian Journal of Science and Technology, vol. 9, pp. 1-10, 2016.

[13] N. d. D. Daniela S. Cruzes, "Communication between Developers and Testers in Distributed Continuous Agile Testing," IEEE, pp. 59-68, 2016.

[14] K. N. R. T. Meryem ELALLAOUI, "Automatic generation of TestNG tests cases from UML sequence diagrams in Scrum process," IEEE, pp. 65-70, 2016.

[15] N. H. Shruti Sharma, "A Comprehensive Study on State of Scrum Development," IEEE, pp. 867-872, 2016.

[16] M. T. a. T. Klima, "PENETRATION TESTING IN AGILE SOFTWARE DEVELOPMENT PROJECTS," International Journal on Cryptography and Information Security (IJCIS), vol. 5, pp. 1-7, March 2015.

[17] P. P. Pratibha Singh, "Impact of agile testing over traditional testing," IJCTM, vol. 1, no. 2, pp. 1-6, 2015.

[18] Product Backlog Rating: A Case Study On Measuring Test Quality In Scrum, 2014.

[19] O. H. a. Y. D. David Talby and Arie Keren, "Agile Software Testing in a Large-Scale Project," IEEE, 2016.

[20] K. Simmi. Bajaj, "Hybrid Test Automation Framework for managing Test Data," International Journal of Pure and Applied Mathematics, vol. 118, pp. 265-277, 2018.

[21] R. S. B. S. B. W. S. S. Sulabh Tyagi, "Development of Reusable Hybrid Test Automation Framework for Web Based Scrum Projects," Journal of Applied Science and Engineering, vol. 21, pp. 455-462, 2018.

[22] S. Srrinidhi and S. Sharaniya, "Action Based Keyword Driven Framework for Testing Web Applications," IJSTE - International Journal of Science Technology & Engineering , vol. 3, no. 09, p. 4, March 2017.

[23] S. Raju and V. Vaidhehi, "Design and Implementation of Hybrid Test Automation Framework for Web Based Application," International Journal of Innovative Research in Advanced Engineering (IJIRAE), vol. 4, no. 03, p. 5, March 2017.

[24] Y. Aziz, "Exploring a keyword driven testing framework," Uppsala Universitet, Uppsala, Uppsala, 2017.

[25] A. A. Ashwaq and R. J. Qureshi, "Novel Framework for Automation Testing of Mobile Applications using Appium," 04 February 2017. [Online]. Available: http://www.mecs-press.org/). [Accessed 2 September 2017].

[26] R. Patidar,, A. Sharma, and R. Dave, "Survey on Manual and Automation Testing strategies and Tools for a Software Application," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 7, no. 4, p. 10, April 2017.

[27] D. Zun, T. Qi and C. Liping, "Research on Automated Testing Framework for Multi-Platform Mobile Applications," in CCIS2016, Bejing, 2016.

[28] D. Nirmala and T. L. Maheswari, "Automated Test Automation Framework for Software Quality Assurance," International Journal of Computer Science and Mobile Computing, vol. 4, no. 12, p. 11, December 2015.

[29] B. Bhagyashree , P. Ranawade, S. Jadhav and M. Vibhute, "Hybrid Test Automation Framework for Web Application," International Journal of Engineering Research & Technology (IJERT), vol. 4, no. 4, p. 6, April 2015.

[30] G. M. D. Gandhi and A. S. Pillai, "Challenges in GUI Test Automation," International Journal of Computer Theory and Engineering, vol. 6, no. 2, p. 4, April 2014.

[31] S. AMARICAI and R. CONSTANTINESCU, "Designing a Software Test Automation Framework," Informatica Economică, vol. 18, no. 1, p. 10, 2014.

[32] L. Bose and S. Thakur, "GRAFT: Generic & Reusable Automation Framework for Agile Testing," in 5th International Conference-Confluence The Next Generation Information Technology Summit (Confluence), -, 2014.

[33] A. S. Kumar and S. Vasavi, "Effective Unit Testing Framework for Automation of Windows Applications," in ICAdC, India, 2013.

[34] Jiujiu Yu, "Design and Application on Agile Software Exploratory Testing Model," in IEEE, China, 2018.