Human Readable Automation Framework for Web Application

Anita V Patil[1], Prof. Dr. S. Sathish Kumar[2]

[1]Computer Science and Engineering, RNS Institute of Technology, Channasandra, Bangalore, India

[2]Professor, Computer Science and Engineering, RNS Institute of Technology, Channasandra, Bangalore, India

Abstract*— Software Framework is a universal software platform in software application. A framework proposes to provide generic functionality of software. Best practice of framework will be used in very software application. A specific software application changes a framework and reuses it. With test framework improves the reusability of test environment. C# language used for creating the framework. The Objective is to test the project fully through automation testing framework. Create the automated test scripts which will do the unit, functionality and UI testing. The test scripts were written in gherkins using Protractor and Selenium in the framework. Gherkin is a Test driven and behavior driven development language which is easy to understand. After writing the test scripts, they are integrated into the concerned test playlist. The test scenarios having feature files are run and they create a test report. The report will say whether the test has passed or failed. When a test fails, a bug is created in the project management tool and reported to the developers. Also, role-based testing, the roles of DevOps, Editor and Approver in my project so based on the roles, testing scripts, integrated them into a playlist and ran the scripts and generated the report.

***Keywords*—Software Framework, Test Scripts, Gherkin**

I. INTRODUCTION

In the Framework element introduction there are several parts in it. The whole framework contains 4 parts. Selenium and Protractor: This part provides the basic function for the webpage operation. The three main operations are Click, Input, Clear. the Selenium to do the web automatic QA and the protractor is the test framework for Angular JS based on the Selenium. Gherkin and Spec flow: Gherkin is a language that is used by Spec flow. And Spec flow is a tool to define, manage, and automatically execute human readable tests in .NET projects. Gherkin and Spec flow provides 3 parts First is about the Gherkin and Spec flow introduction. Second is about the Spec flow customization generator plugin. Third is about the version compatibility. Spec flow is designed based on the Gherkin. Cucumber understands the language Gherkin. It is a "Business Readable, Domain Specific Language". Gherkin serves two purposes documentation and automated tests the NuGet tool. This package can provide many function for the feature file. For the spec flow deep usage, framework provide and the two new projects files that is., Complex Step, Extended Feature Generator. Complex Step Generator Plugin This project is used as the Generator plugin for feature file. It can generate the .cs file from. feature file. We design two different generation methods. One is generating the Unit Test file, the other is the spec flow function file. The Spec flow Version Compatibility using the version 1.9.0 of the Spec flow to write the package Complex Step and Extended Feature Generator, but after upgrading the Spec flow version, there is some error happened. So, we add a new project to handle the version compatibility issue. UIQA Framework: This part cantinas many web operation functions for our environment. Also, it contains many special function for the special element, such as Table, Nav Bar and so on.dd See the project UIQA Framework in Visual Studio,

you can find 5 folders: Action Step, Base Step, Protractor, Result Class and Test Data. Base Step: This folder is made up by .cs file. They are the basic C# function. The most basic C# function is in the folder Under layer. The Operation Widget file define the basic operations on the webpage Click, Clear and Input. Search Widget file define the element basic search function. The Base Step file define server function, contains 3 parts special step, common step and get element step. Action Step: Action Step folder is constituted by spec flow feature file. Those files are all feature function. The folder Special Element Base Step and Special Element Step is designed to the special element, such as table in the webpage. Now I have written 3 kinds of special element: Grid Table, Nav Bar, Panel Side Nav.

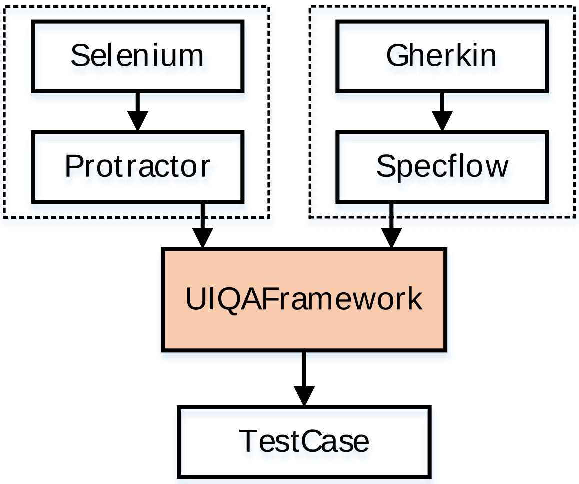


FIGURE1: Framework Introduction

Result Class: There are the total 4 folders in the Result Class Error Email, Screen Shot, Step Log, Web Report. In Error Email the framework will send the error information email to the tester. In Screen Shot the framework will catch the screenshot when the error happened. In Step Log writing the detail step of the test case to a WORD file and using the different color to standard for the error log. In Web Report will get a webpage after running the test cases Test Data: This folder will provide different kinds of the test file, such as WORD, PDF, PPT and so on. This can make our test case more standard. Test Case: This part can contain many projects, which are created to test the specific angular JS projects. They can reference the STEP functions from the UIQA Framework part This project is stand for the project. If we have other project which need to enable the auto-QA function, we can create a new project. Than we refer the project UIQA Framework to write the test case.

II. RELATED WORK

Automated software testing framework [1] has described that testing can show the presence of faults, but it cannot prove that there is no any single fault remaining. The system testing is the responsibility of a separate team. By using interface testing we can minimize defects in the interface and attempts to assess the influence of software development methodologies, software testing techniques and relevant parameters for their applicability conditions so as to help software developers and testers making a proper choice. Agile software methodologies will be best suited technique for software development project. [2] has described that the Selenium is a very good framework for testing the user interface of a web application. It is an extremely powerful tool for running end-to-end functional tests We can write tests in several programming languages and Selenium executes them into one or multiple browsers. In this research paper we did a systematic study of all components of Selenium suite. [4] has described the different testing techniques, the different strategy, different testing methodologies and also different automated testing tools adopted by various Organizations. It also has introduced the latest development in the software testing like test driven development, iterative and incremental testing etc. and has focused on the need of software testing. He has described the different approaches of implementing the integration testing and also reviewed some of the common approaches using the integration testing especially for object-oriented programs. [5] has examined the role of automated software testing against manual testing. The main result of the research paper is that for developing and providing a system aimed for the cloud is effective with agile development process and has described the basic aspect of testing, their objective and the different techniques of software testing. It tries to adopt the best testing suite for the software by the comparative study. [6] has described that Agile Software Testing is more effective when it is used with the tactic of Automation testing. Automation testing is optimum technique to gel with agile framework, as it saves time and testing will be effective. These best practices form the basis for effective Agile Software Test automation. Implementing them can help us to avoid common mistakes and improve your testing process regardless of the automation tool.

III. METODOLOGY

In Agile methodology, software is developed in the all incremental, rapid cycles. Interactions amongst customers, developers and client are emphasized rather than processes and tools. The agile methodology focuses on responding to change rather than extensive planning. Incremental testing is used in agile development methods and hence, every release of the project is tested thoroughly. This ensures that any bugs in the system are fixed before the next release. It is possible to make changes in the project at any time to comply with the requirements. This incremental testing minimizes risks. The project includes several Web methods rolled up in one Web service, so we are authoring and tracking our test cases of the Web methods in the tool.

A. Usage of the Framework

Block1:

```
#region I click the current widget
    [When(@"I click the current widget")]
    [Then(@"I click the current widget")]
    public void I click the current widget()
    {
Ng Web Element widget = Scenario Context .
Current .Get<Ng Web Element>
("Current Widget");
        Click(widget);
    }
#end region
```

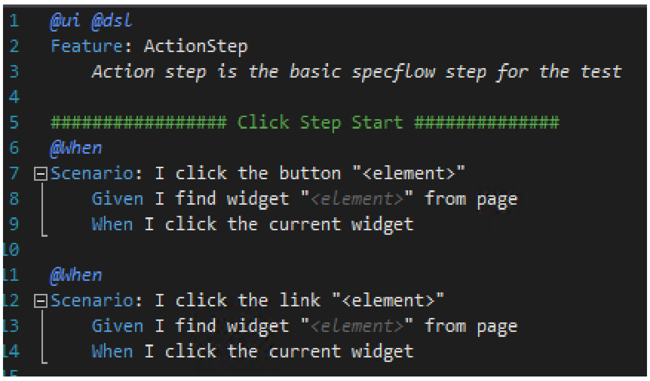
Block2:

```
            @When
            Scenario: I click the image "<element>"
            Given I find widget "<element>" from page
            When I click the current widget
```

Block1 is the C# language and Block2 is the Gherkin language. You can see the step “When I click the current widget” is one of tags in Block2. So, when we execute the step “When I click the current widget”, it will call the function " I click the current widget" in the Block1. This is the relationship between the spec flow step and the C# function.

B. How to create a Feature Function File

Most functions the framework provided are the feature functions. So, it is very important to know how to create a new feature function. First, you need to add feature file in the project specific folder Second, you need to add the tag "@dsl" on the Feature name. This is defined in the project ComplexStep. If you see the following picture, you will find 2 tags on the feature name. Only the "@dsl" is used to define the following scenario are functions not unit tests. The tag "@ui" is to define which C# function the feature file can use. Third, beginning to write the scenario block. On the scenario name, there is a tag "@When". This tag represents this block is a "When" step function. As the first scenario show, we can use the "When I click the button "Add"" in another Scenario to call this one. Next, in the Scenario name, we use "<xxx>" to represent the parameter. This parameter can be used in the following steps. As the picture shown.



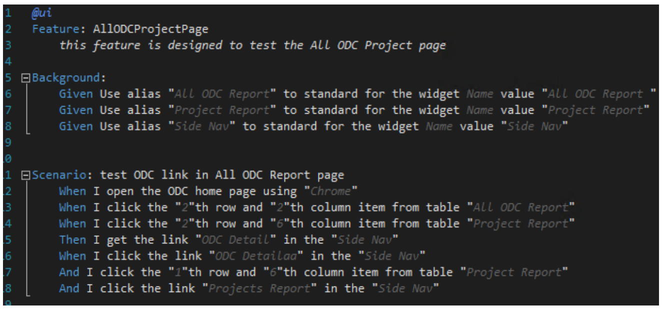
```
@ui @dsl
Feature: ActionStep
    Action step is the basic specflow step for the test

################# Click Step Start ##############
@When
Scenario: I click the button "<element>"
    Given I find widget "<element>" from page
    When I click the current widget

@When
Scenario: I click the link "<element>"
    Given I find widget "<element>" from page
    When I click the current widget
```

FIGURE 2: Feature Function File

Finally, the feature function class will be completed. You will find the framework will auto-generate a .cs file under the feature file. Here will have the whole C# code converted by the Gherkin step language. The conversion process is defined in the project "ComplexStep.

C. How to create a Unit test file

The steps of creating a unit test file is very similar with the feature function file. But no tags "@dsl" and "@When



```
@ui
Feature: AllODCProjectPage
    this feature is designed to test the All ODC Project page

Background:
    Given Use alias "All ODC Report" to standard for the widget Name value "All ODC Report "
    Given Use alias "Project Report" to standard for the widget Name value "Project Report"
    Given Use alias "Side Nav" to standard for the widget Name value "Side Nav"


Scenario: test ODC link in All ODC Report page
    When I open the ODC home page using "Chrome"
    When I click the "2"th row and "2"th column item from table "All ODC Report"
    When I click the "2"th row and "6"th column item from table "Project Report"
    Then I get the link "ODC Detail" in the "Side Nav"
    When I click the link "ODC Detailaa" in the "Side Nav"
    And I click the "1"th row and "6"th column item from table "Project Report"
    And I click the link "Projects Report" in the "Side Nav"
```

FIGURE 3: Unit Test File

IV. RESULT

AD create users, when the user wants to create new user Id then the user must fill all the mandatory fields and then after filling all the mandatory fields then user needs to click on the create user button then after few seconds if all the fields are correct it will display successful message, if the fields are incorrect then it will be display the error message.

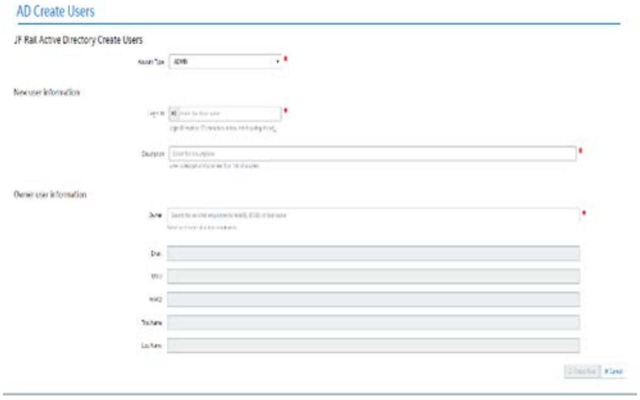
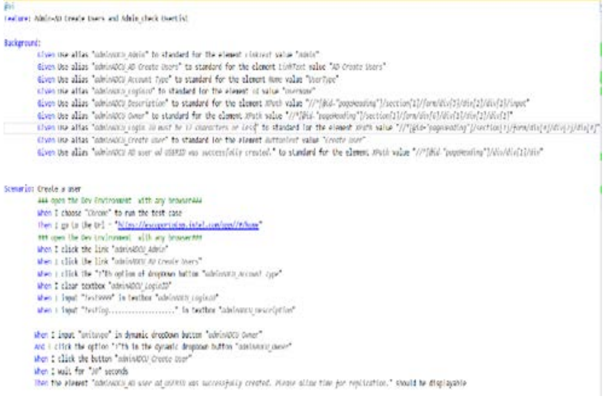


FIGURE 4: AD Create Users



```
@ui
Feature: Admin-AD Create Users and Admin_check UsertList

Background:
    Given Use alias "adminADCU_Admin" to standard for the element LinkText value "Admin"
    Given Use alias "adminADCU_AD Create Users" to standard for the element LinkText value "AD Create Users"
    Given Use alias "adminADCU Account Type" to standard for the element Name value "UserType"
    Given Use alias "adminADCU_LoginID" to standard for the element id value "Username"
    Given Use alias "adminADCU Description" to standard for the element XPath value "//*[@id-"pageHeading"]/section[1]/form/div[3]/div[2]/div[2]/input"
    Given Use alias "adminADCU Owner" to standard for the element XPath value "//*[@id-"pageHeading"]/section[1]/form/div[6]/div[1]/div[2]/div[2]"
    Given Use alias "adminADCU_Login ID must be 13 characters or less" to standard for the element XPath value "//*[@id="pageHeading"]/section[1]/form/div[[illegible]]"
    Given Use alias "adminADCU_Create User" to standard for the element ButtonText value "Create User"
    Given Use alias "adminADCU AD user ad USERID was successfully created." to standard for the element XPath value "//*[@id-"pageHeading"]/div/div[1]/div"


Scenario: Create a user
    ### open the Dev Environment with any browser###
    When I choose "Chrome" to run the test case
    Then I go to the Url - "[illegible]"
    ### open the Dev Environment with any browser###
    When I click the link "adminADCU_Admin"
    When I click the link "adminADCU AD Create Users"
    When I click the "1"th option of dropdown button "adminADCU_Account Type"
    When I clear textbox "adminADCU LoginID"
    When I input "Test####" in textbox "adminADCU LoginID"
    When I input "Testing......................" in textbox "adminADCU Description"

   When I input "[illegible]" in dynamic dropDown button "adminADCU Owner"
   And I click the option "1"th in the dynamic dropdown button "adminADCU_Owner"
   When I click the button "adminADCU Create User"
   When I wait for "30" seconds
   Then the element "adminADCU_AD user ad USERID was successfully created. Please allow time for replication." should be displayable
```

FIGURE 5: AD Create Users Scenario

FIGURE 4.3: AD Create Users Output

This one example for the framework, the FIG 4.1 describes the framework how it is building and the FIG 4.2 containing the test scripts or scenario and explains whole about the framework and the FIG 4.3 explains the output of that scenario that scripts pass or fail, and then automatically its generates the report.

V. CONCLUSION

The paper explains about a new generation test automation framework integrated with the concepts of both input and output as a human readable code. By the implementation of the framework the product quality can be realized on daily basis and quality gaps can be filled at the earliest management gets a better control over the product releases with adequate test coverage. Product developments with long term scope are ideal candidates for automation using the proposed framework. The advantages of automated testing include the time saved performing functional testing allowing learning technology staff to evaluate the software under investigation, instead of just testing it. Automated testing can be used to support service transition as well as regression testing. In the future tests could be reused to perform more realistic load testing and service availability monitoring. The challenges of an automated testing model include maintenance of tests scripts. As UIs change, new scripts require development and existing scripts updating. Unless test scripts are going to be reused and maintained then the initial investment in time required writing the scripts may prove too onerous. In addition, some scripts still require human intervention to verify the test was successful, for example checking a screen shot output.

ACKNOWLEDGEMENT

I would like to thank Dr. S. Sathish Kumar (Dept of CSE, RNSIT, Bangalore) for his valuable suggestions and comments that helped improving this works, this support is greatly appreciated

REFERENCES

[1] Fewster, M. and Graham, D. (1999). Software test automation. Harlow: Addison-Wesley.

[2] Shahamiri, S.R., Kadir, W.M.W.K., Ibrahim, S., Hashim, S.ZM. (2011). An automated framework for software test oracle. Information and Software Technology 53. 774–788. DOI: 10.1016/j.infsof.2011.02.006

[3] Santos, P., Hernndez-Leo, D., Pérez-Sanagustn, M., Blat, S. (2012). Modeling the Computing Based Testing domain extending IMS QTI Framework, models and exemplary implementations. Computers in Human Behavior 28. 1648-1662. DOI: 10.1016/j.chb.2012.04.003

[4] A. Jain and S. Sharma, "An Efficient Keyword Driven Test Automation Framework for Web Applications," International Journal of Engineering Science & Advanced Technology, vol. 2, no. 3, pp. 600-604, 2012

[5] S. Lamba, V. Rishiwal and A. Rana, "An Automated Data Driven Continuous Testing Framework," International Journal of Advanced Technology in Engineering, vol. 3, no. 10, 2015.

[6] T. Kosa, M. Mernikb and T. Kosarb, "Test automation of a measurement system using a domain-specific modelling language," Journal of Systems and Software, vol. 111, p. 74–88, 2016.

[7] "Test Automation for Web Applications," 2017. [Online]. Available: http://www.seleniumhq.org/.

[8] Software Testing-Brief introduction to security testing by Nilesh Parekh published on 14-07-2006 available at http://www.buzzle.com/editorial/7-14-2006-102344.asp

[9] Open source security testing methodology manual of PETE HERZOG and the institute for security and open methodology-ISECOM.

[10] Software testing by Jiantao Pan available at http://www.ece.cmu.edu/~roopman/des- 899/sw_testing/