IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-11 E-ISSN: 2347-2693

Automation Testing Frameworks for SharePoint application

Madhu Dande[1]*, Neelima Galla[2]

[1*,2]*Department CSE, TCS, India*

www.ijcseonline.org

Received: Oct/22/2015 Revised: Nov /05/2015 Accepted: Nov/20/2015 Published: Nov/30/2015

***Abstract*—**This Automation Framework has been designed and developed for the SharePoint migration/up gradation for a large insurance company in UK. For any SharePoint migration Project to be automated in the different phases to detect the defects/bugs at early stages of the SDLC. The Current IT industry follows either waterfall model or an Agile methodology to execute a project. For the last two years, all the projects around the world are moving towards the Agile Methodology to execute it. Irrespective of the process or methodology to be followed, the following testing phases: Unit, Integration, System and User acceptance testing need to be followed before moving into production. Design and develop the automation framework which will be used as wrapper to test the SharePoint application based on the requirement from the client. Automation Framework will be using the Selenium/QTP Functional testing tool to develop the Automation Framework Engine or a Wrapper scripts to test the application. Based on the Architecture of the application, developed a framework which is very much similar to test the application. These frameworks were developed either Scenario based or Functionality based. In this Paper, designed and developed both the automation frameworks to achieve the goal of the testing in System Testing phase. At the same time, come-up with the steps to validate the Unit testing phase by simple automation framework. Even in this unit testing phase, can be identify the defects in early stages of the testing which will bring down cost of the defect. As well move the application faster to market because of the testing done by the automation framework in Unit and System testing. Main objective of this paper is to reduce the cost of the defect and delivering defect free quality to the customer.

***Keywords*—**Middleware, Database, CAM model, Cloud networks, Migration Testing, Unit Testing, Integration Testing, System Testing, Automation Testing Framework

I. INTRODUCTION

Automated software testing is a logical application which validates the other applications with the goal as delivering the defect free quality. This approach will increase the testing efficiency, Testing effectiveness and faster to market i.e. moving the application into production.

Software Applications follow a process of SDLC models based on the organization and the project to execute. As discussed earlier in the abstract section, most of the organizations are now interested to run their projects using the Agile methodology where there is a possibility of change in the requirements at any moment of the phase or iteration/cycles. The Application requirement specifications are not needed to be frozen before moving to design.

Considering the Microsoft SharePoint is a web application framework in Windows platform. Microsoft SharePoint integrates intranet, content management, document management and yammer etc.

The SharePoint provides intranet portals, document and file management, collaboration with social networks, extranets, websites, enterprise search, and business intelligence. [6] It also has system integration, process integration, and workflow automation capabilities.

Basic architecture of the SharePoint is 3 tiers or 4 tiers which can be called as Web based or distributed systems. These systems will have client, Middleware server and database server.

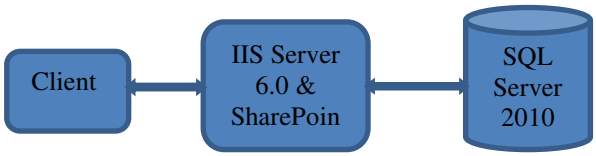


Fig. 1

[3]An enterprise wise application is a business application which stores the huge data with heterogeneous files; most of the users will call it as big application. In today's corporate environment, enterprise applications are complex, scalable, distributed, component-based, and mission-critical. They may be deployed on a variety of platforms across corporate networks, intranets, or the Internet.

They are data-centric, user-friendly, and must meet stringent requirements for security, administration, and maintenance. In short, they are highly complex systems.

Fig. 2

II. ARCHITECTURE OF MIGRATION FROM MICROSOFT SHAREPOINT-2010 TO SHAREPOINT-2013

SharePoint Server 2013[6] brings a vast number of new features, capabilities, and enhancements in Web Content Management. All these exciting changes and the new SharePoint licensing model will likely mean that we see more and more Internet-facing Web sites based on SharePoint technology.

Based on the above SharePoint Architecture, testing team can develop automation framework to validate the SharePoint application at unit and integration testing phase.

A. AS-IS SharePoint Architecture-SharePoint-2010

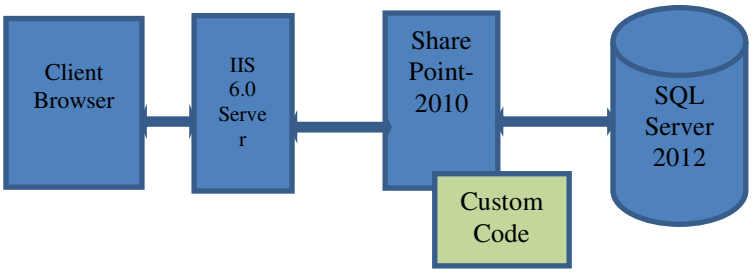


Fig. 3

B. To-Be SharePoint Architecture-SharePoint-2013

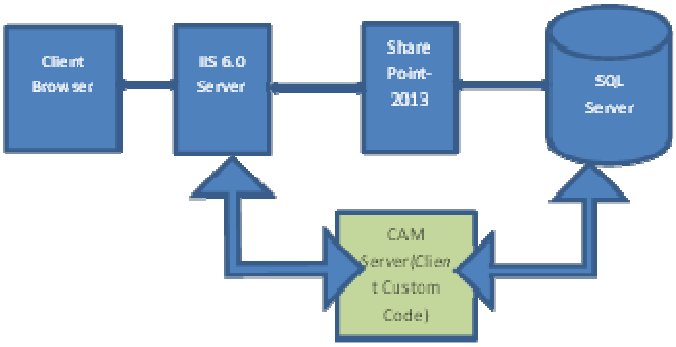


Fig. 4

In the Agile methodology, based on the user story testers complete the Card Level Testing (Unit/Integration Level Testing), Integration Testing and System Testing. The Testing team is completely involved in end to end testing. The card level testing can be done for the web services, the Integration testing is done to verify the user story if it is working as expected at the front end, the system Testing is done to complete the end to end testing to make sure to cover all possible test scenarios.

Automation can be done using the Nunit[7] at unit level Testing. In IT industry the development cost is not same as the testing cost i.e. the cost of the each resource depends on the technical knowledge and capabilities in developing the code based on the requirements and knowledge on how to debug/test the code at unit level testing.

As in the project development phase, resource ramp Up/Down occurs based on the stage of the application. There is a possibility that development team will ramp up in the middle of project journey with the resource based on the schedule due to that, project can be in risk(Quality/Standards will not be able follow in the entire project).

Development team starts developing the code based on the requirement specifications. In this paper we consider the project is following an Agile process, considering more number of Developers to develop the application with less time and there is a possibility of not validating all the conditions at unit level testing for all the features which has been developed by the developers. In the Integration Testing phase, developers validate the Access Specifiers of the Class/Functions of the methods.

SharePoint has list of functionality

SNo	SharePoint Functionality	Features
1	Functional and GUI	• Uniform style implementation • Custom grids & navigations

		• Branding schemes
2	Search	• Search Relevance • Cross Site Search • Search Admin and Management
3	Integration	• Search Integration • Notification Integration • Web Services
4	Core Services	• Site templates • Alerts • Navigation
5	Personalisation	• Site & Role management • Auditing Services • Policy Management services
6	Business Process & Forms	• Browser based forms • Centralized forms • Business data features
7	Collaboration	• Email and Calendar • Document Collaboration • Issue Tracking
8	Business Intelligence	• Dash Boards • Data Connection libraries • Excel Services

Table-1

Developing the Automation Framework based on the customized SharePoint application
Each functionality or component to be analyzed, design and develop the automation framework using the selenium.

- Functional and GUI
- Search
- Integration
- Core Services
- Personalisation
- Business Process & Forms
- Collaboration
- Business Intelligence

These features were designed and developed by the Microsoft as a product and it has been tested thoroughly. When the customization has been done for a specific functionality, need to make sure to understand the requirements accordingly and develop the automation framework based on the functionality.

On SharePoint functionalities, design and develop the automation Framework where it can test the application thoroughly.
Considering the features to be validated

1. Templates
    a. SharePoint-2013 (Content types)
    b. Customized Templates
2. Master Page
3. Search conditions
4. User profile
5. Provisioning process
6. CAM solution validation

Security: when you use the CAM model
Due to CAM model, Performance testing of the application to be done but it is not in the scope of this paper.

Work to be validated for the SharePoint application.

1. The depreciated features/function code
2. New enhancements
3. Architecture of the newly introduced SharePoint-2013 CAM model
4. Validation of the Templates between SharePoint 2008 to SharePoint 2013.
5. Validation of Search conditions with different combinations/conditions of test data
6. Validation of the different languages between the two versions.(SharePoint application is multilingual application)
7. Promoting Sites to Site Collections, Demoting a site collection to a site
8. Security Permissions: What are the different permissions to the folder structures?
9. User Roles: What are the different roles?
10. Features: What are the different Features?
    - Standard
    - Project/Team Requirement
    - Customization Code
    - Feature mappings, including deprecated features and enhance new features etc..
11. Interface: Is there any integration with other(internal/external applications) systems
    - Automatically uploading of documents
    - Manually pushing of the documents/contents
    - Batch jobs to push the contents
12. Mobility Testing: Different devices to be validated
    - Laptops
    - Computer Monitors
    - ipads
    - Touchpads
    - Mobiles
13. Usability testing : Responsible for validating those contents display on those devices
1. Updating the document from the different teams, whether document is in review or approved state.
2. No format or symbol of the document is to be same in the SharePoint.
3. Microsoft office products integration testing
4. Third party products integration testing.
1. Custom code to be written
    a. Custom code
    b. Custom list templates
    c. Custom site templates
    d. Custom web parts/ Web services
    e. Third party web services for integration

2. Designer workflow Microsoft drag and drop functionality, we can complete the workflows. (Less code) with limitations.
   List of limitations:
   a. We will not be able to achieve the features/functionality
3. System Workflow (In-Built Workflows)

Note: Microsoft is suggested not to go with custom code workflow better to choose the designer workflow. From SharePoint 2013 onwards there is no need to use the work flows.

Unit/Integration Testing Phase:

As SharePoint consists of the different features/web parts/ templates and web sites which will be different from 2010 to 2013. In the Migration/Upgradation of the SharePoint application features and technology will be different in both SharePoint 2010 and SharePoint 2013. But the Features, Sites, Web Parts and Templates will be same but there is a possibility of the tweaking.

List down all the features between the two versions of the SharePoint in the same order as follows

1. Compare the Size of the .dll and .exe files of the SharePoint 2010 to 2013
2. Compare the Size of the .dll and .exe files of the client custom code 2010 to 2013
3. All the functions and classes of the SharePoint and Custom code to be validated by using the Reflection concept in VSTS-2012, extract the Methods and arguments between the two different versions and compare it.
4. Validate all the web sites which are created the Business users by using the utility driver to execute all the web site links
   By using above tool
   - Validate all the URL links
   - Validate the Images
   - Try to capture the download time of the each image
   - Calculate the number of images on the pages download time i.e. response time.
   - Generate the metrics
   - Generate the graphs
   - Create Analysis report based on the execution details.
5. Validate the Templates between SharePoint 2008 to SharePoint 2013.
6. Search condition to be validated with different combinations of test data
7. Web service to be validated in both in-flow and out-flow of data/transactions
8. Third-party web services to be validated which consumes the in-flow data
9. Validate the different languages which need to be validated between the two versions.
10. SQL DB content can be validated between the 2008 and 2012
    a. Size
    b. Number of Records
       i. Folder structures

As per the It industry standards, the software testing estimation is done based on the development effort i.e. Testing effort = 30 to 40% of Development effort.
To reduce the cost of the testing at unit/integration testing, need to create generic Automation Framework for migration/upgradation projects.

Based on the above SharePoint Architecture, testing team can develop automation framework to validate the SharePoint application at system testing phase.

Different ways to automate the system testing of SharePoint application migration

1. Automate the SharePoint application using QTP or Selenium functional testing tool by creating a framework or following the above scenarios.
2. Microsoft also provided the CodedUI testing tool to validate the scenarios

By using the above testing tools, SharePoint application can be tested during the system testing phase.
As mentioned, testing the SharePoint application in two ways either you can follow the scenario based or Screen based with certain workflows.

C. Scenario Based Validation:

Based on the User Stories/Use cases provided by the Business analyst, manual test cases are created.

Architecture of the Framework for Scenerio based test cases:

Automation Framework is built based on the scenarios based functionality, which is standard in the SharePoint application. Even in the next version or new version whether the architecture of the application is changing still automation testing team can use the same Framework to test the SharePoint application as shown in the fig.5.

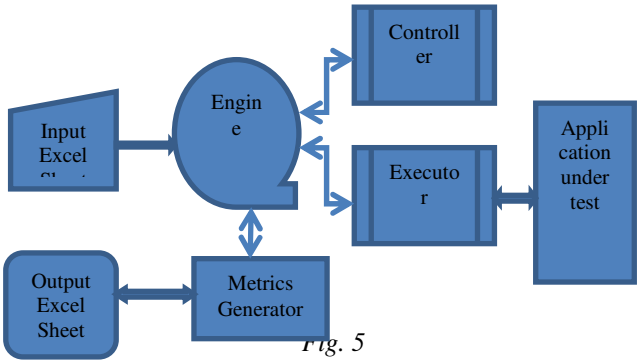


Fig. 5

Architecture of the Framework for user Story based test cases:

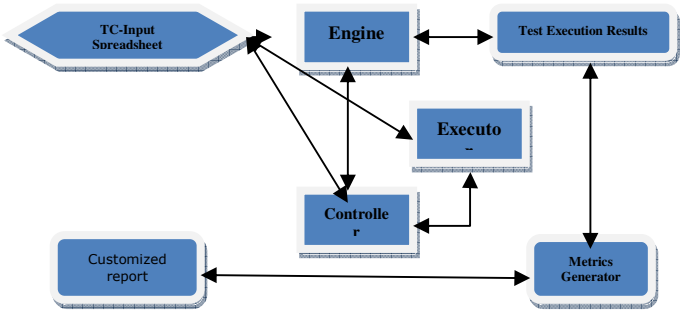


Fig. 6

Automation Framework is built based on the User story for the agile methodology, which is standard in the SharePoint application. Even in the next version or new version whether the architecture of the application is changing still automation testing team can use the same Framework to test the SharePoint application as shown in the fig.6

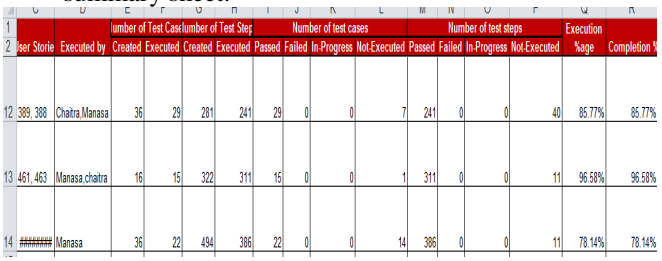
Automation Test Case Creation: Test cases are created based on the User Stories / Use Cases.

Using the User Stories/Use Cases generate the test scenarios, based on the scenarios create the manual test cases by the testers.

Automation Test case Creation			
Test case number	Test Case Name	Test Step	Description
Action/Functions	Object	Input Value	Expected Value
Actual Value	Results		

Table-2

1. **Engine**: Engine is the Heart of this Framework, which will create a copy of the spreadsheet and drive through entire workbook with worksheets.
2. Collect all the updated results in the spreadsheet and create a Manual Test cases Execution result sheet.
3. **Controller**: Read the each test cases and test steps
4. **Executor**: Based on the actions/functions, each test step will be executed and updated the results
5. **Metrics**: Results are calculated and updated in the summary sheet.

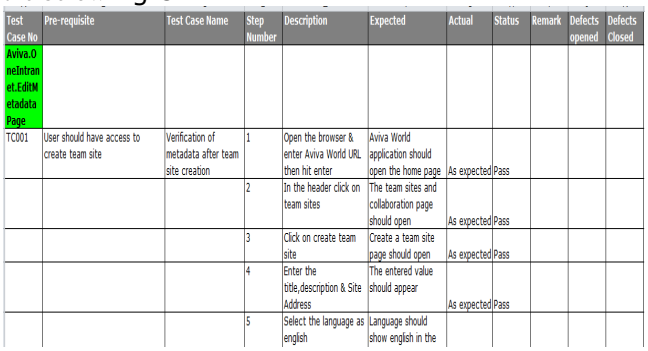


			Number of Test Cases		Number of Test Steps		Number of test cases				Number of test steps				Execution	
2	User Stories	Executed by	Created	Executed	Created	Executed	Passed	Failed	In-Progress	Not-Executed	Passed	Failed	In-Progress	Not-Executed	%age	Completion %
12	389, 388	Chaitra,Manasa	36	29	281	241	29	0	0	7	241	0	0	40	85.77%	85.77%
13	461, 463	Manasa,chaitra	16	15	322	311	15	0	0	1	311	0	0	11	96.58%	96.58%
14	########	Manasa	36	22	494	386	22	0	0	14	386	0	0	11	78.14%	78.14%

Fig. 7

Based on the manual test cases which are already created for the previous versions will be used to automate it using Selenium in Visual Studio.

Test cases where created in the standard format as show in the below fig.8



Test Case No	Pre-requisite	Test Case Name	Step Number	Description	Expected	Actual	Status	Remark	Defects opened	Defects Closed
Aviva.OneIntranet.EditMetadataPage										
TC001	User should have access to create team site	Verification of metadata after team site creation	1	Open the browser & enter Aviva World URL then hit enter	Aviva World application should open the home page	As expected	Pass			
			2	In the header click on team sites	The team sites and collaboration page should open	As expected	Pass			
			3	Click on create team site	Create a team site page should open	As expected	Pass			
			4	Enter the title,description & Site Address	The entered value should appear	As expected	Pass			
			5	Select the language as english	Language should show english in the					

Fig. 8

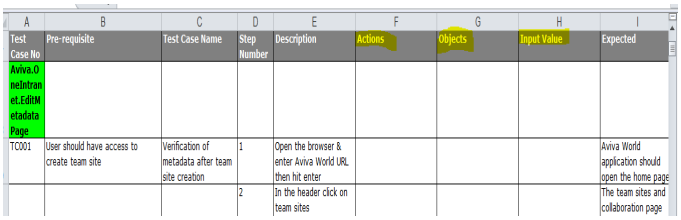
Basically Automation is done using this Spreadsheet, where the manual test cases which are already present. Automation testing framework is to be developed in such a way that, Framework engine should be able to create an object of the manual test cases workbook and read the worksheet.

Steps to create Automation Test Framework for scenarios/Story/Use case based test cases.

1. Create object Excel file
2. Copy the existence manual test cases workbook
3. Count the number of worksheets present in the workbook
4. Read each worksheet name
5. Start reading the Worksheet details, which includes Test Case Number (Column-A), Test Case Name (Column-C), Step Name (Column-D).
6. Once you encounter the Step Name, read the column downwards to count the number of rows.

E.g.: Considering a test case that has 10 test steps, if by any chance the test step number is wrong. In that case, it should read the previous row of the column-c to check for the test case name is present. If the test case name is present then stop reading the test steps elseif read the column-D for the

next row value until the value is decrements and apply the same.



A	B	C	D	E	F	G	H	I
Test Case No	Pre-requisite	Test Case Name	Step Number	Description	Actions	Objects	Input Value	Expected
Aviva.OnelntranetEditMetadataPage								
TC001	User should have access to create team site	Verification of metadata after team site creation	1	Open the browser & enter Aviva World URL then hit enter				Aviva World application should open the home page
			2	In the header click on team sites				The team sites and collaboration page

Fig. 9

7. Read the Actions (column-F), Objects (Column-G) and Input Value (Column-H).
8. Note: Refer the snapshot, columns are highlighted.
9. Drive the entire test steps.

Advantages of this Framework

1. Easy to develop the Manual test cases, based on these test cases current framework will run through i.e. Execute the test cases.
2. Automation Script Knowledge is not needed to develop the manual test cases
3. Any change in the architecture, testing will be easy
4. Execution time will be easier
5. Changes of the Test steps through excel is easy
6. Manual intervention will be less
7. For different version of the SharePoint application can be testing using the same framework

Case Study of Return on Investment (ROI):

Number of Manual Test cases: 1421
Assume: The number of test steps for each test case is 20
Based on that 1421*20 = 28420 test steps
As per the software testing standard: Number of test steps executed in a day is 220
28420 /220 = 130 days to complete the test execution with one resource
26 days will take to complete one test execution cycle with 5 resources
As per current state of manual test execution is done in 30 days with Six resources.
Let's consider the test case creation is followed the standards
Eg: Each test case will be having the following test steps default for all the test cases.
1. Login
2. Enter the valid UID and PWD
3. Validate the home screen
4. Logout
Based on the above assumption we can reduce the 1421*4 = 5684 test steps while execution of the Manual test cases.
Test Execution done by 3 resources within one month, we can do two cycles of test automation execution. With this efficiency and effectiveness of the testing is increases and cost of the execution cycle is reduced by **40-50%.**

CONCLUSION

Microsoft SharePoint application has upgrades to the newer version with new features and depreciating the old features. With the help of Automation frameworks, it can reduce the effort when frequency of testing is increased. In that case, Automation Framework will be used based on the requirement. These two approaches will be applicable for a SharePoint application testing. As any business domain companies will use the SharePoint application for longer duration. So it needs automation testing to be done, when there is a change in the Hardware or Software (Either Versioning or changes in the software). Any organization needs to save the cost and increase the efficiency of the testing by implementing automation. If the Automation is introduced or implemented then the monetary benefits 40 to 50% of the cost (Return on Investment (ROI)).

REFERENCES

[1] "What is sharepoint?". Microsoft Office 2010 Answers. Microsoft.
[2] "What is an Enterprise Application?". Microsoft Office 2010 Answers. Microsoft.
[3] "SharePoint-2013". Microsoft Office 2010 Answers. Microsoft.
[4] "What is sharepoint?". Microsoft Office 2010 Answers. Microsoft.
[5] Gilbert, Mark R.; Shegda, Karen M.; Phifer, Gene; Mann, Jeffrey (19 October 2009)."SharePoint 2010 Is Poised for Broader Enterprise Adoption". Gartner. Retrieved13 August 2011
[6] www.microsoft.com
[7] Laurie Williams1 , Gunnar Kudrjavets "On the Effectiveness of Unit Test Automation at Microsoft"

AUTHORS PROFILE

Madhu Dande, received the M.Tech degree in Computer Science & Engineering from Visvesvaraya Technological University during 2001 - 2003, completed his B.Tech in Electrical & Electronics Engineering from Sri Venkateswara University during 1995 - 1999. He is member of IETE and CSI. He has worked in Centre for Development of Telematics (C-DOT), Bangalore, where he has implemented Voice of Ethernet project successfully. Later he has joined TCS as Automation Test Lead in 2003. He has shown his skills design and development of the Automation Framework. He had patent on Generic Unit and Integration Automation Framework Testing granted by USPTO. In the process of filing a patent on Demon Web UI Utility to validate the broken and orphan links at run time of the business critical applications. Designed and developed a Framework on Robust Automation Testing tool for functional testing. Currently working on Human Vs System Resources utilization in Production Environment to reduce the utilization of power, Maintenance Cost and increase the availability.

Neelima Galla, recevied M.S degress in Maths and Computer science from Alcorn state university in 2012-2015. She is interested in the area of Automation Testing Framework and Big Data Testing.