

Software Automation Testing using LeanFT Tool

Reshma S.G., Ashoka S.B., Manu Ankanahalli Gowdaiah

Abstract:- Software industries are giving more importance for to developing, maintaining, servicing and delivering the best quality of software products always to the customer. To achieve this goal software industry's main objective is having the best quality of software testing. Nowadays the automation testing activity is performing the big role in the software industry. Automation testing tools are makes easier to identify the bugs during the regression testing by running them n number of times and rectified with all effective manners within short period of time with low cost expenses. Here we are focusing on to make use of the latest and having advanced feature in LeanFTv14.52 (Lean functional Testing) software automation testing tool in our automation testing project, This LeanFT tool is been introduced after the UFT Developer (Unified Functional Testing) tool from the Micro Focus microfocus.com. Software test automation user will get more benefited in his automation scripting activity by using this LeanFT software automation testing tool as it is having the new benefited features for to testing the application by consolidated with high level of tool support, having better customization support as per the project requirement and perfect compatibility support as per the test environment requirement, technical help from customer care support to customize the tool as per the testing application needs in all possibilities by comparing to other software automation testing tool. LeanFT software automation testing tool can be used to accelerate test with intelligent solution for more than 200 technologies across web application, standalone application, mobile application, mainframe application, SAP, Salesforce, Java, PDF, Citrix, API(Application Programming Interface), RPA(Robotic Process Automation) and enterprise apps also including financial and highly secured and confidential applications are can be tested.

Keywords:- Allure, ALMOctane, Azure DevOps, Eclipse, Emailable, Git-Hub, LeanFT, Maven, TestNG, UFT Developer.

I. INTRODUCTION

The LeanFTv14.52 (Lean functional Testing) software automation testing tool is supports parallel test execution in the different test environment of multilingual sites and multiple applications to test both front-end and back-end functionality service parts to increase the test coverage across the UI and API and intelligently improve the productivity, speed and resiliency with application integration based capabilities and accelerates continuous testing feedback through direct integration with source control and CI/CD tools. The main usability of this LeanFTv14.52 software automation testing tool is depends on the automation framework implemented and web drivers and data driven supports.

Revised Manuscript Received on September 25, 2020.

*Corresponding Author

*Reshma S.G., Assistant Professor, Department of Computer Science (MCA) MCU Bengaluru, Karnataka, India, Email: reshmasg.it@gmail.com

Dr. Ashoka S.B., Assistant Professor, Department of Computer Science MCU Bengaluru, Karnataka, India, Email: dr.ashoksbs@gmail.com

Manu Ankanahalli Gowdaiah, Test Manager, Software Application Automation Testing Team Head, iFocus Systec India Pvt. Ltd. Bengaluru, Karnataka, India, Email: manugowdaiah@gmail.com

This suite will be designed in the IMR (Internal Maven Repository) by having the combination of framework Jars, LeanFT Jars and central maven repository. Git-hub use to supporting for the code merging activities and TestNG tool is an executor are merged with the framework, finally this automation suite will be run or executing through the help of Jenkins or Azure DevOps by accessing or controlling through the virtual machine or remote control system and also we can get the test executed report in the ALM-Octane if it is integrated with this automation framework.

II. AUTOMATION REGULAR PROCESS

The following below steps are followed in the automation process

- Test tool selection:-** Test tool selection largely depends on the technology[1], application under test is built on, usability and supporting on the application as expected[2].
- Define scope of automation:-** Scope of automation is the area of application and necessity application features and environment under test which will be automated[3].
- Planning, design and development:-** During this phase create automation strategy, automation framework, automation environment and its utility and plan.
- Test execution:-** Automation scripts are executed as per the release plan with report during this phase[4].
- Maintenance:-** As new functionality are added to the system under test with successive cycles. Automation scripts need to be added, reviewed and maintained for each release cycle. Maintenance becomes necessary to improve effectiveness of automation scripts.

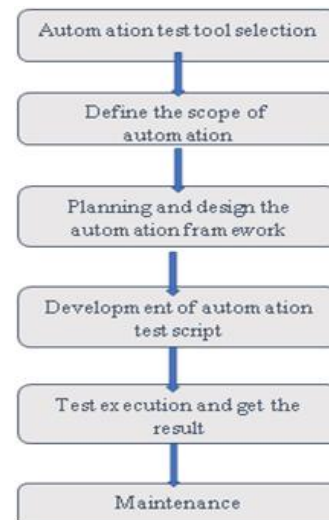


Fig 1. Automation process

III. LEANFT AUTOMATION TESTING USAGE

We are explaining here the advantages and usability of LeanFTv14.52 (Lean functional Testing) tool advanced features by giving an example[5] of testing a software application with obtained yield effective result.

A. LeanFT Automation Test Execution

The LeanFTv14.52 start executes the individual project or parallel multiple projects in the same time on the standalone or virtual machine as per the scheduled time and we have different options for to extract the execution resultant report as per the user convenient.

B. Parallel Executions

Parallel execution option can be setup in the automation framework, also this features are applied in the software development organizations, then they can be perform the automation test execution for the multiple project at a time which can be reduces the overall test execution time and cost. All the assigned different projects test suites are executing in this TestNG parallel execution system,

C. LeanFTv14.52 Test Report

We have different reporting mechanisms like TestNG report, Emailable Report, Allure Report and all these types of reporting options[6] are supporting in the LeanFTv14.52 tool. These reports contains

- Graphical representation of number of test cases executed, Test Pass Percentage, Execution Time.
- Detailed steps for each test case
- Screenshot of each action performed
- Validation points
- Test Status

A sample test execution reports are mentioned below.

D. TestNG emailable-report

TestNG emailable-report.html report are displayed as below, it has recorded the screenshot of an each action during the test execution.

Test	Methods Passed	Scenarios Passed	# skipped	# failed	Total Time	Included Groups	Excluded Groups
Default test	5	5	0	0	1,069.5 seconds		

Class	Method	# of Scenarios	Start	Tim (ms)
Default test — passed				
newgen.testcases.EF_Loan_less than_10M	TestCase1: Verify user is a	1	1588678280907	4088
	TestCase2: Verify user is a	1	1588678690773	1494
	TestCase3: Verify user is a	1	1588678940232	1527
	TestCase4: Verify user is a	1	1588679004450	1492
	TestCase5: Verify user is a	1	1588679130552	1889
newgen.testcases.EF_Loan_less than_10M: TestCase6				


```

05-May-2020 05:15:50 PM: Verify that As a Requestor user, I can able to find the Active Facility a Facility' you
05-May-2020 05:15:50 PM: Launch the application .
05-May-2020 05:15:58 PM: Enter the UserId
05-May-2020 05:15:59 PM: Enter the Password
launchAndSignIn/NewGenTest
    
```

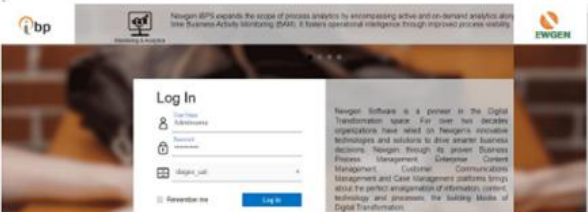


Fig 2. TestNG emailable-report.html and screenshot

E. Allure Dashboard report

Attaching a sample Allure Dashboard for reference.

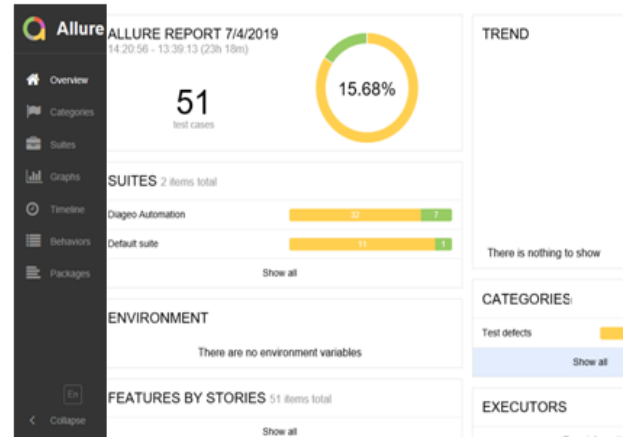


Fig 3. Allure dashboard -report details with graph.

IV. PREREQUISITES OF LEANFT TOOL USE

There are few prerequisite actions are required before make using of the LeanFTv14.52 software automation testing tool, details are mentioned below in detailed.

A. LeanFT Test Environment Setup

Setup the Maven: Download the maven supporting software from <https://maven.apache.org/download.cgi> and install maven.

Add a new system variable: Go to environment variables and add a new system variable %MAVEN_HOME% and point same to the maven bin folder.

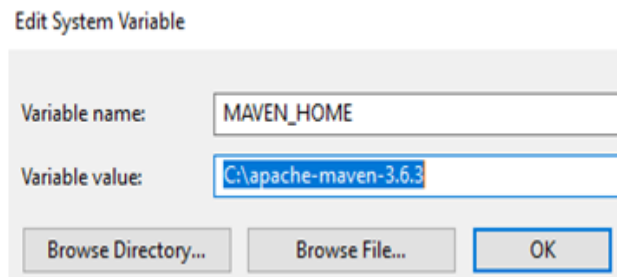


Fig 4. Add the new system variable

Node.js: Download the node.js supporting software from <https://nodejs.org/en/download/> website and install the node.js

Download the LeanFT tool software: Download the LeanFTv14.52 software automation testing tool from Micro Focus www.microfocus.com website and install in to the system. User can make use of this same version of the trailed version tool to verify the tool usability and also can be make sure to supporting all the feature of the testing environment and application successfully or not as before getting the licensed tool.

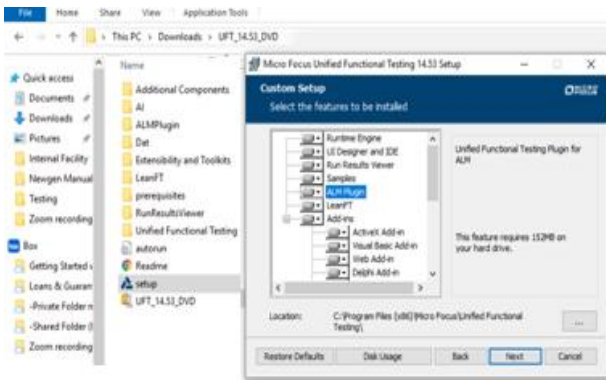


Fig 5. Download LeanFT software

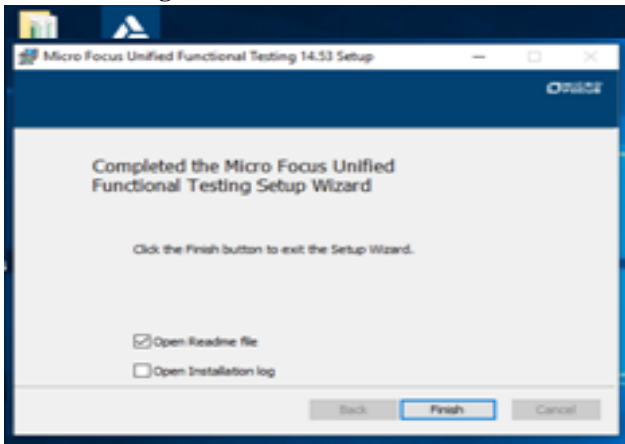


Fig 6. Completed the Installation of LeanFT software

B. Set the LeanFT home

Set the LeanFT software home in System variables page.

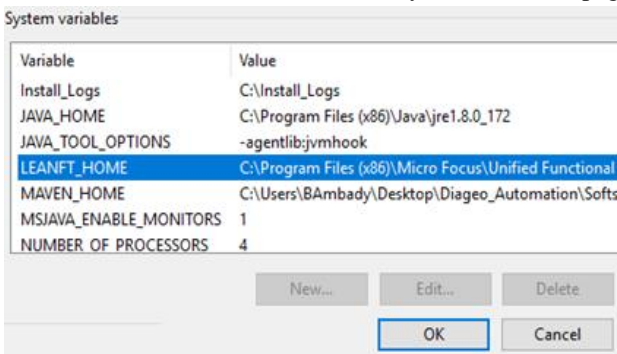
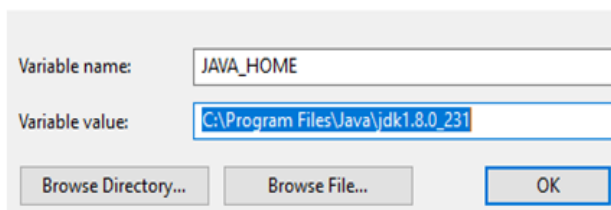


Fig 7. Set the LeanFT home by browsing the drive

C. JAVA

Install Java and set JAVA home. (JAVA_HOME/Installation path)

Edit System Variable



D. Allure

Download and install allure from the website mentioned here <https://docs.qameta.io/allure/> and add allure to the path variables. Allure which helps for to display the automation script executed report.

C:\allure-2.9.0\bin

E. Download and install eclipse from

<https://eclipse.org/downloads/>



Fig 8. Install and open eclipse

F. Time Engine activated

Time Engine is activated or not. If activated, you should get this below Active popup. If not activated then make it active by opening the Functional Testing License Wizard from the run command field.



Fig 9. LeanFT engine activated popup

G. If Time Engine is not activated

If the Time Engine activated pop-up doesn't appear then mouse over on the show hidden icons to confirm is the active icons displayed or not, if not then navigate to Start icon and follow the steps.

Start-> Functional Testing License Wizard-> Choose License Type -> Connect -> Install License Enable LeanFT License and check if UFT Run.

INSTALLATION REPORT

LICENSE INSTALLED SUCCESSFULLY

Product UFT One Concurrent User

Type Concurrent License

EXIT WIZARD

Fig 10. Activated license



H. LeanFT tab on the eclipse header



Fig 11. LeanFT tab on the eclipse header

The eclipse and IDE should be chosen during the LeanFT installation. After done with successful LeanFT installation in a standalone application, open the eclipse and find a new tab for LeanFT in eclipse as shown. Displayed LeanFT tab on the eclipse header is indicates as the confirmation of successful configuration.

V. FILE AND FOLDER STRUCTURE

LeanFT project folder structure is depends on the developed automation framework. Sample structure arranged as Maven dependencies, JRE System library, result, build and test data folders as below

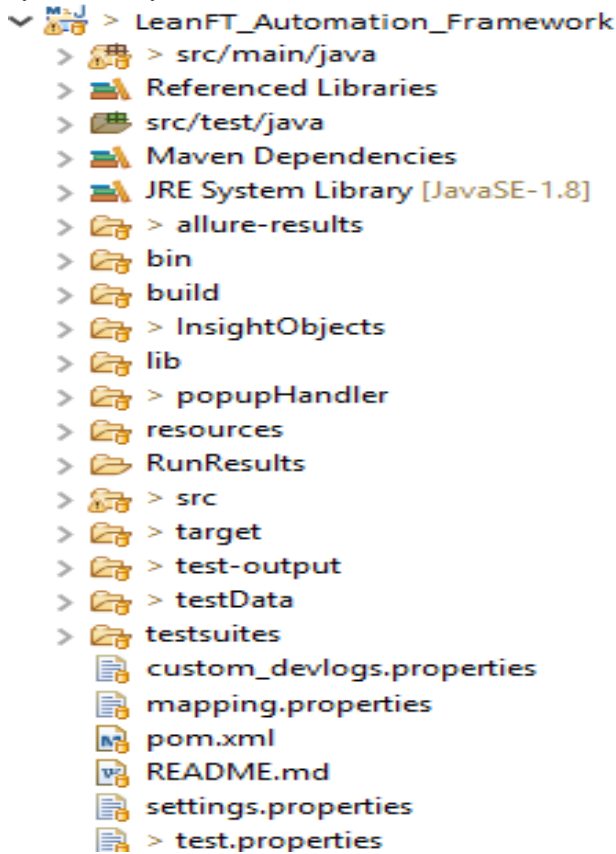


Fig. 12. File and folder structure

VI. SAMPLE AUTOMATION PROJECT

As discussed from the above details is clearly explained the usability and applicability and advantages of the LeanFTv14.52 software automation testing tool.

From here onwards we would like to give a sample of automation project implementation and execution using LeanFT software automation testing tool.

Page Object Model (POM) is the design pattern, used in our LeanFT automation framework[7]. The advantage of this model is that it reduces code duplication and improves

test maintenance. Automatically highlighted and displayed the error on the duplication.

- **Actions**
- **Pages**
- **Testcases**

Actions: Under this model, all these options are supporting for to implement the automation scripting of an each web page in the application.

Action methods are going to be creating an each action performs on the application individually, which perform the operations on those Web Elements.

Pages: Also there should be a corresponding **Page Class**. This Page class are going to be creating for the prepuces of identifying an each object and the element of the page. This will identify the Web Elements of that web page and contains.

Testcases. Test cases folders are going to be creating for the prepuces of implementing the complete functional flow of the application as per the test scenarios by

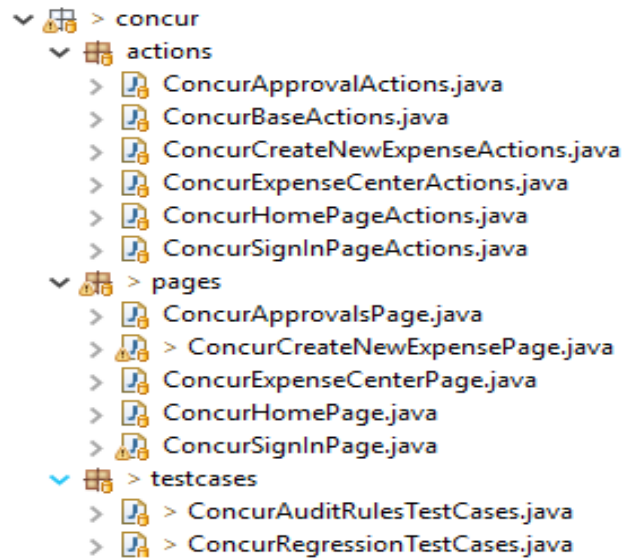


Fig 13. Automation testcase development structure

A. Creating a LeanFT Automation Test

By referring to the above we are going to be implementing a new automation test as following below.



Creating the packages individually as testcases and assigning them for the LeanFTfactory setup.

- Create <applicationname>.testcases package. (E.g.: salesforce.testcases)
- This package contains classes with all testcases
- All classes in this package should extends *LeanFTFactory.class*

```
@Test(priority=34)
public void auditRule_USL_Communication_expense_FutureDate()throws Exception {
    String methodName= this.getTestName().toString();new ConcurCreateNewExpensePage().clearingVariables();
    new ConcurHomePage().clearingVariables();
    TestNGUtils.reportLog("*****Testcase: "+methodName+"*****");
    Start.runAsConcurTester.launchAndSignInConcurApplication_USLgeneralUser()
        ._inhomePage()
            .verifyConcurHomePage()
            .selectProfileUser()
            .searchbynameorID(methodName)
            .navigateToCreateExpensePage()
            ._inCreateNewExpensePage()
            .enterReportDetails(methodName)
            .createExpenseReportauditrule(methodName)
            .enterAuditRuleExpenseDetails(methodName)
            .validateException(methodName)
        ._inhomePage()
        .doLogout()
        ._inSignPage()
        .verifyLogout();
}
```

Fig 14. Creating a new LeanFT Automation Testcase

B. Creating a new Action

A new action script has been created for to verify a successful login action by referring to the functional test case flow.

- Create <applicationname>.actions package. (E.g.: salesforce.actions)
- This package contains classes with all actions related to a testcase.
- All classes in this package should extends its *BaseActions.class*
- All base actions should extend *LeanFTFactory.class*

```
import framework.initialise.PageFactory;
public class GuinnessHomePageActions extends GuinnessBaseActions {
    PageFactory webAppPageFactory = new PageFactory();
    public GuinnessHomePageActions verifySuccessfulLogin() throws Exception {
        webAppPageFactory.guinnesshome().VerifySuccessfulLogin();
        return this;
    }
    public GuinnessHomePageActions validateLanguage(String methodName) throws Exception {
        webAppPageFactory.guinnesshome().validateLanguageDesktop(methodName);
        return this;
    }
    public GuinnessHomePageActions navigateToDateTimeSelection(String methodName) throws
        webAppPageFactory.guinnesshome().ChooseDateTimeSlots(methodName);
        return this;
    }
}
```

Fig 15. Creating a new Action

C. Creating a new Page

A new page script has been developed for the concur page by referring to the functional test case flow.

- Create <applicationname>.pages package. (E.g: concur.pages)
- This package contains classes with page level actions related to a testcase
- All classes in this package should extends *LeanFTBasePage.class*
- All base pages should extend *LeanFTFactory.class*



```

16 public class ConcurExpenseCenterPage extends LeanFTBasePage {
17
18     public static LeanFTElements UPLAOD_RECEIPTS_BUTTON = new LeanFTElements("upload receipts + button", "//di
19     public static LeanFTElements BROWSE_BUTTON = new LeanFTElements("Browse button", "//div[contains(@class,'
20     public static LeanFTElements SAVE_AS = new LeanFTElements("Save as window", "Open", LeanFTElements.TEXT);
21     public static LeanFTElements SAVE_AS_BUTTON = new LeanFTElements("Save as button", "Button", LeanFTElemen
22     public static LeanFTElements SAVE_AS_NAME_TEXT_FIELD = new LeanFTElements("filename edit field", "Edit", L
23     public static LeanFTElements OPEN_BUTTON = new LeanFTElements("FileName", "&Open", LeanFTElements.WINDOW
24     public static LeanFTElements CANCEL = new LeanFTElements("FileName", "Cancel", LeanFTElements.TEXT);
25     public static LeanFTElements BROWSE_BUTTON_VAL1 = new LeanFTElements("FileName", "ext-comp-1149", LeanFTE
26     public static LeanFTElements BROWSE_BUTTON_VAL2 = new LeanFTElements("FileName", "DIV", LeanFTElements.TA
27
28     public ConcurExpenseCenterPage() throws Exception {
29     }
30     public ConcurExpenseCenterPage clickOnUploadButton() throws Exception {
31         TestNGUtils.reportLog("click on upload receipt button in expence center page");
32         Thread.sleep(5000);
33         getBaseAction().Click(UPLAOD_RECEIPTS_BUTTON);
34         return this;

```

Fig 16. Creating a new page

D. Creating a Page Level Action:

- All actions are available in LeanFT base actions.
- Newer actions can be written as and when needed on to *LeanFTBaseActions*
- Call `getBaseAction()` method to perform a page level action. (E.g: Clicking on an object)

```
getBaseAction().click(SAP_REQUESTS);
```

E. Click Method

Writing a basic method for to perform an action as per the instruction by calling the methods in the script when and there where required.

```

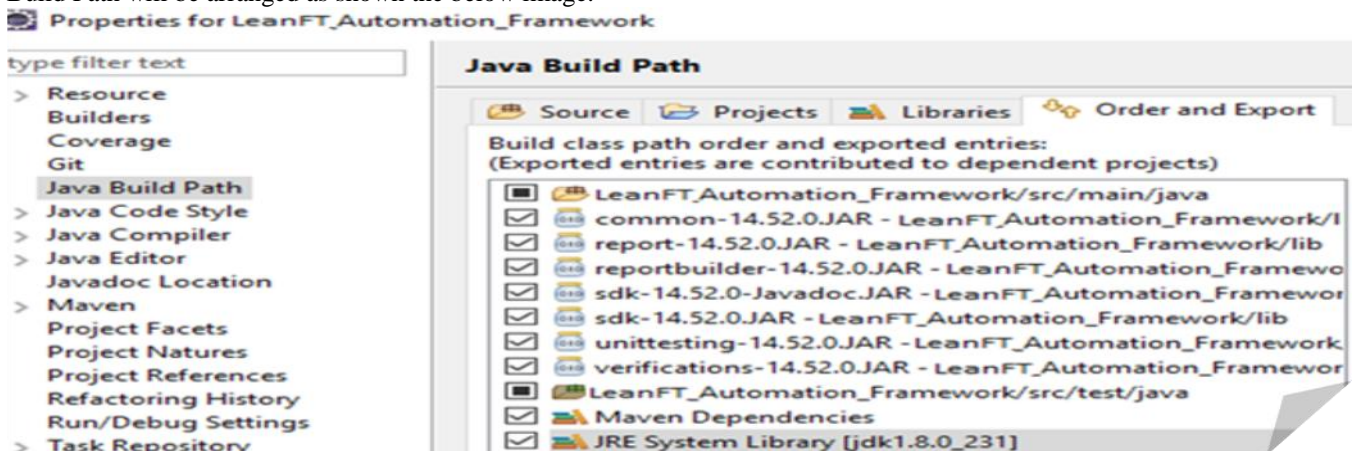
public void click(LeanFTElements objValue) throws GeneralLeanFtException {
    try {
        if (objValue.getelementType().contains("WINDOW")) {
            //System.out.println("Window" + "Click");
            Button button = getWindowButton(objValue);
            button.click();
        } else {
            WebElement element = getWebElement(objValue);
            //System.out.println("Web" + "Click");
            element.click();
        }
    } catch (Exception e) {
        TestNGUtils.reportError("Unable to click on object :" + objValue.getelementName());
        throw new TestNGException(e.getLocalizedMessage());
    } finally {
        TestNGUtils.reportSubLog(("Click on object :" + objValue.getelementName()));
    }
}

```

Fig 17. Creating a new method for click action

F. Build Path

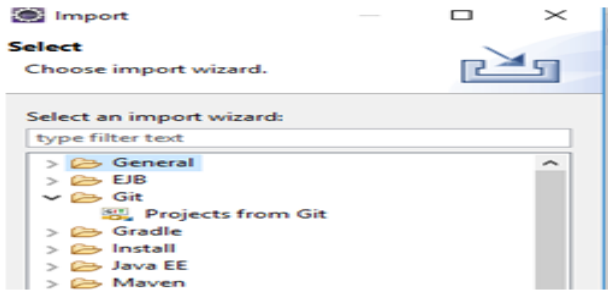
Build Path will be arranged as shown the below image.



VII.LEANFT TEST EXECUTION PROCEDURES

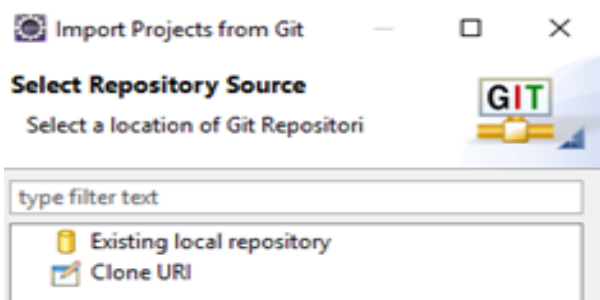
- LeanFT test execution standard procedures followed, Import the projects from Azure Git to Eclipse IDE.





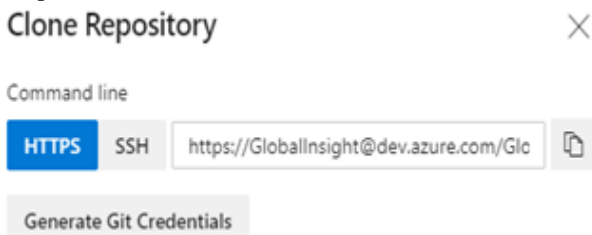
b) Clone URI

Import the Project from Git hub and clone URI.



c) Make it Clone

Copy the URL path of https:\\ of the project to be imported from Azure Git.



d) Provide the url in the eclipse, enter username and Password for to proceed the further actions

e) Click Next > Provide destination location > Click Finish.

f) Project will be imported in to the eclipse workspace

g) POM.xml

Below dependencies should be available in the POM xml of any of the automation project.

```
<dependency>
  <groupId>com.hp.lft</groupId>
  <artifactId>sdk</artifactId>
  <version>14.52.0</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/sdk-14.52.0.JAR</systemPat
</dependency>
<dependency>
  <groupId>com.hp.lft</groupId>
  <artifactId>sdk</artifactId>
  <version>14.52.0</version>
  <classifier>javadoc</classifier>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/sdk-14.52.0-Javadoc.JAR</s
</dependency>
<dependency>
```

```
<groupId>com.hp.lft</groupId>
<artifactId>report</artifactId>
<version>14.52.0</version>
<scope>system</scope>
<systemPath>${project.basedir}/lib/report-14.52.0.JAR</system
</dependency>
<dependency>
  <groupId>com.hp.lft</groupId>
  <artifactId>unittesting</artifactId>
  <version>14.52.0</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/unittesting-14.52.0.JAR</s
</dependency>
<dependency>
  <groupId>com.hp.lft</groupId>
  <artifactId>verifications</artifactId>
  <version>14.52.0</version>
  <scope>system</scope>
  <systemPath>${project.basedir}/lib/verifications-14.52.0.JAR<
</dependency>
<dependency>
```

Fig 18. Pom.xml file

- Build the framework
- Right Click and select the Project >> Maven >> Update Project in Project structure.

VIII. LEANFT TEST SCRIPT EXECUTION

LeanFT test script execution method is also having the common options which are all applicable in the other automation testing tools.

Like open the test project and directly clicking on the test run button or run a particular testcase, but can be use the better and advanced options in the LeanFT Test Execution as followed here.

A. XML suite Execution with pom.xml

Maven surefire plugin is used to run the project tests. It also allowing us to configure the pom.xml file with XML suites to execute when we build our project.

Below configuration will tell maven surefire plugin to execute only concurTestSuite.xml and its test parameters.

Sample TestSuite.xml:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <version>2.20</version>
  <configuration>
    <source>1.8</source>
    <target>1.8</target>
    <suiteXmlFiles>
      <suiteXmlFile>${project.basedir} {1}
    </suiteXmlFiles>
  </configuration>
</plugin>
{1}/testsuites/concurTestSuites.xml</suiteXmlFile>
```

B. Sample TestSuites.xml file

A sample concur TestSuites.xml file for reference.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "http://testng.org/testng-1.0.dtd" >
<suite name="Diageo_Automation_Concur" parallel="classes"
thread-count="5">
<listeners>
<listener class-name="framework.utils.TestListener" />
<listener class-name="framework.reporter.TestNGReporter" />
</listeners>
<test name="ConcurRegression_TestCases">
<classes>
<class name="concur.testcases.ConcurAuditRulesTestCases"/>
</classes>
</test>
</suite>
```

C. Runtime creation of testing.xml

Runtime creation of testing.xml is creating by using Trigger class, this main class[9] helps in and saving the time for creation and maintenance of testing.xml. If group name is mentioned to all the testcases which needs to be executed at a time, this class will pick those testcases and create testing.xml on runtime and execute.

```
<plugin>
<groupId>org.codehaus.mojo</groupId>
<artifactId>exec-maven-plugin</artifactId>
<version>1.6.0</version>
<executions>
<execution>
<phase>test</phase>
<goals>
<goal>java</goal>
</goals>
</execution>
</executions>
<configuration>
<mainClass>framework.initialise.Trigger</mainClass>
<cleanupDaemonThreads>false</cleanupDaemonThreads>
<classpathScope>compile</classpathScope>
</configuration>
</plugin>
```

Fig 19. Runtime creation of testing.xml

This testing.xml is one of the ways to execute a main method from Maven.

mainClass (*framework.initialise.Trigger*) The main class to execute.

classpathScope(*exec.classpathScope*, **Default:compile**) Defines the scope of the classpath passed to the plugin Set to compile, test, runtime or system.

cleanupDaemonThreads(*exec.cleanupDaemonThreads*) Whether to interrupt/join and possibly stop the daemon threads upon quitting. If this is false, maven does nothing about the daemon threads. When maven has no more work to do, the VM will normally terminate any remaining daemon threads. In certain cases, we might need to keep this enabled to make sure threads are properly cleaned up to ensure they don't interfere with subsequent activity.

Maven is based around the central concept of a **Build Life Cycles**. Inside each Build Life Cycles there are **Build Phases**, and inside each Build Phases there are **Build Goals**.

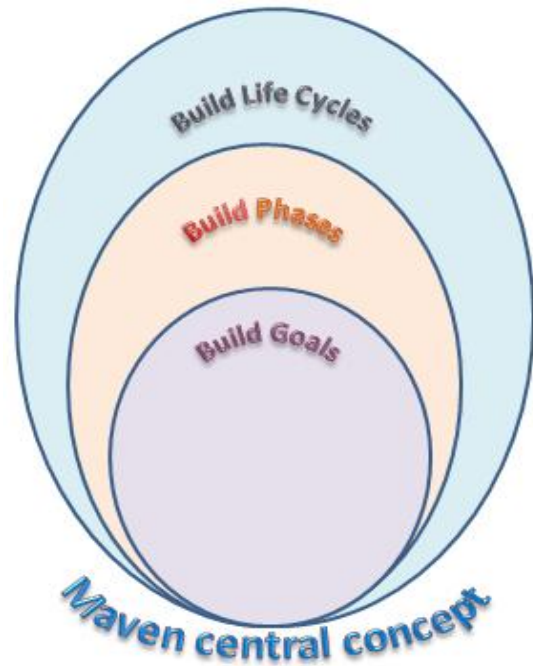


Figure 20. Maven is based around the central concept

When we creating a plugin execution in our Maven build file[10], we only have to specify the goal, after that it will bind that goal to a given default phase.

If we specify a phase then maven will run all phases up to the phase as we specified in order and for each phase it will run all goals attached to that respective phase.

E.g.: If we specify a package, then

- First it will run through the **compile phase**
- Then the **test phase**
- Finally the **package phase**

So finally user can able to get the test execution report from the LeanFT tool in the method of TestNG emailable-report.html report and Allure dashboard report.

For more details please refer with graph as mentioned in the sections III. D. and III. E.

IX. RESULT AND DISCUSSION OF LEANFT TOOL

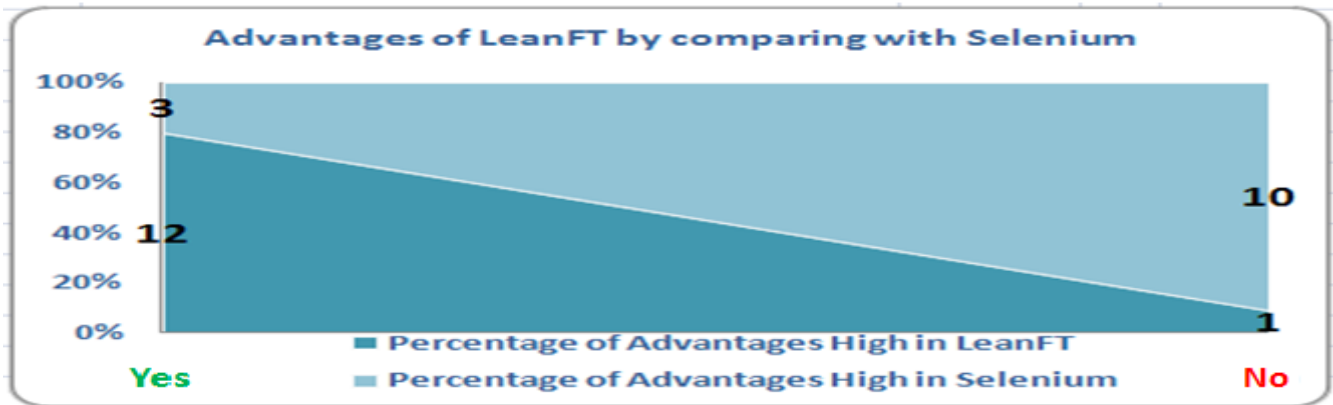
There are many advantages in the usability of the LeanFTv14.52 automation testing tool is mentioned by comparing with one of the popular Selenium automation testing tool.

As discussed from the beginning of this paper we have given the detailed justification on this LeanFTv14.52 tool is having around 80% of the advantages and technical benefits by comparing with the Selenium tool here.

Table- I: Experimented results in tabular form.

High Advantages are in	Yes	No
LeanFTv14.52 automation tool	12	1
Selenium automation tool	3	10

A. Experimented results in chart (High advantages in LeanFT by comparing with Selenium)



B. Experimented results in List (High advantages in LeanFT by comparing with Selenium)

#	LeanFTv14.52 automation testing tool	Advantage High in LeanFT	Selenium Automation testing tool	Advantage High in Selenium
1	LeanFT supports both browser based and window based or standalone applications.	Yes	Selenium support to test only browser based application and it will failed to handle window-based or standalone app	No
2	Supports to test Web, WPF (Windows Presentation Foundation), Mobile, mainframe, SAP, RPA, Windows standard, .NET Siebel UI, Windows Forms and IIR (Insight Image Recognition)	Yes	Supports to test only for Web applications	No
3	LeanFT is a licensed tool, user has to purchase this license for to use also can make use of the trailed version for time being.	No	Selenium is a open source tool and it is free to download and use anybody	Yes
4	LeanFT will be free of cost if it is user is already using the UFT tool	Yes	Selenium is completely freeware	Yes
5	Already has the automation framework that supports end-to-end automation of multiple technologies	Yes	Automation framework not existing, we have to create it from the scratch	No
6	Compatible with Hybrid automation framework, mobile automation framework and mainframe automation framework	Yes	Compatible with standard automation framework	No
7	Customization and compatibility support, technical help from Micro Focus customer care support	Yes	No support from technical help or customer care team	No
8	Support for multiple IDEs (Integrated development environment) are Eclipse, Visual Studio and JavaScript	Yes	Support IDE only for Selenium	No
9	It gives best solution for Browser based applications, Dev Ops and CI (Continuous Integration), Agile and continues testing apps	Yes	Gives best solution only for Browser based applications	No
10	LeanFT is very powerful support for cross browser automation testing	Yes	Need to be make setting this for cross browser testing	No
11	Supports for multiple coding languages are Java, C# and Java Script etc.	Yes	Supports for open source languages such as Java, Python, PHP, C#, Ruby and pure JavaScript etc.	Yes
12	Lean FT provides powerful tools like OIC (Object Identification Center) and AM (Application Models) inbuilt	Yes	No such tools availability in Selenium	No
13	LeanFT tool is having application secured and safety options due to licensed tool.	Yes	Selenium does not have any application secure and safety options due to an open source tool.	No



X. CONCLUSION AND FUTURE WORK

This LeanFT tool is introduced and maintaining by the Micro Focus <https://www.microfocus.com> after takeover from the HP company. This LeanFTv14.52 tool is been upgraded with the latest version for to make use and better supporting for the multiple applications, compatibility with the different test environment and framework. The conclusion is by using this LeanFTv14.52 software automation testing tool usability in the different test environment and testing the different software applications with obtained yield effective results are in web application, standalone application, mobile application and mainframe application including with financial and highly secured and confidential applications are can be tested including other supporting tools like Maven. TestNG Etc. This LeanFTv14.52 software automation testing tool features are supporting for to make easier and compatibility in testing the software applications and maintaining the better quality in the development of automation scripting and secured the software product using this licensed tool. By referring to this detailed mentioned in this paper and the above conclusion we shall be plan for to make some more detailed research work on to impalement the advanced and compatibility framework for all possible application testing and tool compatibility and application secured in the future work.

REFERENCE

1. T. Kanstrén."A Review of Domain-Specific Modelling. Software Testing."(The Eighth International Multi-Conference on Computing in the Global Information Technology). 2013.
2. S. H. Trivedi."Software Testing Techniques."(International Journal of Advanced Research in Computer Science and Software Engineering). vol. 2. no. 10. pp. 433-439. 2012.
3. V. Sangave."(Generic Test Automation." International Journal of Science and Research (ISJR)). volume. 4. no. 7. 2015
4. A. Holmes ; Digital Focus. "(Automating functional tests using Selenium)" Agile Conference. 2006
5. V. Sangave ."(A Review on Automating Test Automation.)" International Journal of Advance Research in Computer Science and Management Studies. vol. 2. no. 12. 2014.
6. Monier."(Evaluation of Automated Web Testing Tools.)" International Journal of Computer Applications Technology and Research. vol. 4. no. 5. pp. 405-408. 2015.
7. M. Sadiq "(A Method for the Selection of Software Testing Automation Framework using Analytic Hierarchy Process.)" International Journal of Computer Applications (0975 – 8887). 2014.
8. H. S. Gill. M. Kaur."(Automation Framework of Browser Based Testing Tool Watir.)" International Journal of Computers and Distributed Systems. vol. 1. no. 3. 2012.
9. Arjun Rana and Vinay Rishiwal "(An Automated Data Driven Continuous Testing Framework)" International Journal of Advanced Engineering Research and Applications (IJAERA) Vol. – 1. Issue – 3. July –2015 ISSN: 2454-2377
10. P. Yalla. L. S. and S. Reddy."(Framework for Testing Web Applications using Selenium Testing Tool with Respect to Integration Testing.)" International Journal of Computer Science and technology.

AUTHORS PROFILE



Reshma S. G., MCA, and pursuing (PhD) working as a Assistant Professor in the Department of Computer Science (MCA)from past one year at Maharani Cluster University Bengaluru, and worked as a Assistant Professor from past 2 years in the Department of Computer Science(MCA) at Bengaluru University, Karnataka, India,
Involving in the publications activities, Joining the PhD., and involving in research work, undergone the subject oriented and many technical training and achieved the JAVA certification and some more certification on technical domain.
Email ID: reshmasg.it@gmail.com.



Dr. Ashoka S. B., MCA PhD., Working as Assistant Professor in the Department of Computer Science from past 6 years (MCA) and taken responsibility as a PG-Coordinator in this same department at Maharani Cluster University Bengaluru, Karnataka, India.
Published more than 20 research papers and involved in publishing the technical books, Involved in research work and project guiding activities, undergone the subject oriented and many technical training, achieved many certification on technical domain and involved in getting the central government projects to the college and prepared the patent papers and articles.
Email ID: dr.ashoksbc@gmail.com.



Manu Ankanahalli Gowdaiah, M.Sc., Computer Science, working in the software engineering industries from past 12 years, Now working as a Test Manager, Software Application Automation Testing Team Head in the iFocus Systec India Pvt. Ltd. Bengaluru, Karnataka, India, Involving in the publications activities, Joining the PhD., and involving in research work, undergone the multiple technical training, Product IT Security awareness EN technology training and GRIPS training, iSDoc viewer and LC480 system application training, and achieved the Roche Diagnostics and cobas instruments Hardware Engineering certification at Switzerland, Scaled Agile Framework® (SAFe®) certification, Computer Networking certification, Got the successful ISO 13485 Medical Devices quality management system certification,
Email ID: manugowdaiah@gmail.com