Scientific Journal of Impact Factor (SJIF): 4.72
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 7, July -2017

Proof of concept (Poc) selenium web driver based Automation framework

B AJITH KUMAR

Master of science (Information Technology)
Department of Mathematics
COLLEGE OF ENGINNERING GUINDY (CEG)
Anna university

ABSTRACT: *To control test execution time. Software testing is a process of executing a program or application with the intent of finding the software bugs. It can also be stated as the process of validating and verifyingthat a software program or application or product: Meets the business and technical requirements that guided it's design and development. Works as expected.*

KEY WORDS: *SOFTWARE TESTING .AUTOMATION TESTING ,SELENIUM, SELENIUM WEBDRIVER ,AGILE TESTING , TESTNG*

Test automation :

In software testing, test automation is the use of special software (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. Test automation can automate some repetitive but necessary tasks in a formalized testing process already in place, or add additional testing that would be difficult to perform manually.

AGILE TESTING :

A software testing practice that follows the principles of agile software development is called Agile Testing. Agile is an iterative development methodology, where requirements evolve through collaboration between the customer and self-organizing teams and agile aligns development with customer needs.

Selenium automation tool

Selenium has the support of some of the largest browser vendors who have taken (or are taking) steps to make Selenium a native part of their browser. It is also the core technology in countless other browser automation tools, APIs and frameworks.

1.1.1 Selenium WebDriver

• •

• create robust, browser-based regression automation suites and tests
scale and distribute scripts across many environments

Selenium WebDriver

The biggest change in Selenium recently has been the inclusion of the WebDriver API. Driving a browser natively *as a user would* either locally or on a remote machine using the Selenium Server it marks a leap forward in terms of browser automation.

Selenium WebDriver fits in the same role as RC did, and has incorporated the original 1.x bindings. It refers to both the language bindings and the implementations of the individual browser controlling code. This is commonly referred to as just "WebDriver" or sometimes as Selenium 2.

Selenium 1.0 + WebDriver = Selenium 2.0

• WebDriver is designed in a simpler and more concise programming interface along with addressing some limitations in the Selenium-RC API.

- WebDriver is a compact Object Oriented API when compared to Selenium1.0
- It drives the browser much more effectively and overcomes the limitations of Selenium 1.x which affected our functional test coverage, like the file upload or download, pop-ups and dialogs barrier
- WebDriver overcomes the limitation of Selenium RC'

TESTNG
TestNG is testing framework inspired from most popular JUnit framework used for the Java programming language. The TestNG framework is introduced to overcome the limitations of JUnit framework. Most of the automation users are using this framework because of its advantages & more supported features. Until we have executed selenium test script but not even generated test reports. So using this TestNG framework we will learn how to generate test reports.
TestNG is a testing framework inspired from JUnit and NUnit but introducing some new functionalities that make it more powerful and easier to use, such as :

- Annotations.
- Run your tests in arbitrarily big thread pools with various policies available (all methods in their own thread, one thread per test class, etc...).
- Test that your code is multithread safe.
- Flexible test configuration.
- Support for data-driven testing (with @DataProvider).
- Support for parameters.
- Powerful execution model (no more TestSuite).
- Supported by a variety of tools and plug-ins (Eclipse, IDEA, Maven, etc...).
- Embeds BeanShell for further flexibility.
- Default JDK functions for runtime and logging (no dependencies).
- Dependent methods for application server testing.
  TestNG is designed to cover all categories of tests: unit, functional, end-to-end, integration, etc...

Proof of concept(poc)

(poc) selenium web driver based Test Automation framework
Every project need a basic poc to understand the tool suitability to the overall project along with strength and weakness implementing an incorrect tool or best tool at incorrect time may affect project expense and cause failure to the test automation projects
so we preparing proof of concept for selenium web driver based automation framework
introduction
selenium web driver based automation framework consists of set of tools and process to get implemented in web based testing cycles carried over agile project this can be done by downloading the selenium web driver based java jar files from selenium HQ
selenium 2.o version has been identified to construct the automation framework and the it eventually support Firefox 33.o browser alone with chrome and IE browser for Poc.

Automation Poc approach
As a prerequisite to the automation project, subset of manual test cases are identified and selected for converting them to automation tests.
Poc Consideration
java based selenium automation framework has been considered for the poc and other framework such as c# ,phyton,ruby and java script based frameworks are not included in the Poc similarly selenium 2.o is identified to be used for the tests rather than using latest IDES Such as intellij etc

Assumption and dependencies

assuming that the java has been installed in the target machine and java home and path updated when the java project (automation framework is imported in latest version of eclipse, the project may get updated with latest eclipse combability scripts and it may not get imported in older version of eclipse

Risks association

re installation of browsers and additional add-ins to the browser and change of operating system to the target machine are critical factors to work in latest test environments if selenium 3.o is implemented to the framework ,gecko driver or marionette drivers as an 28aug2016

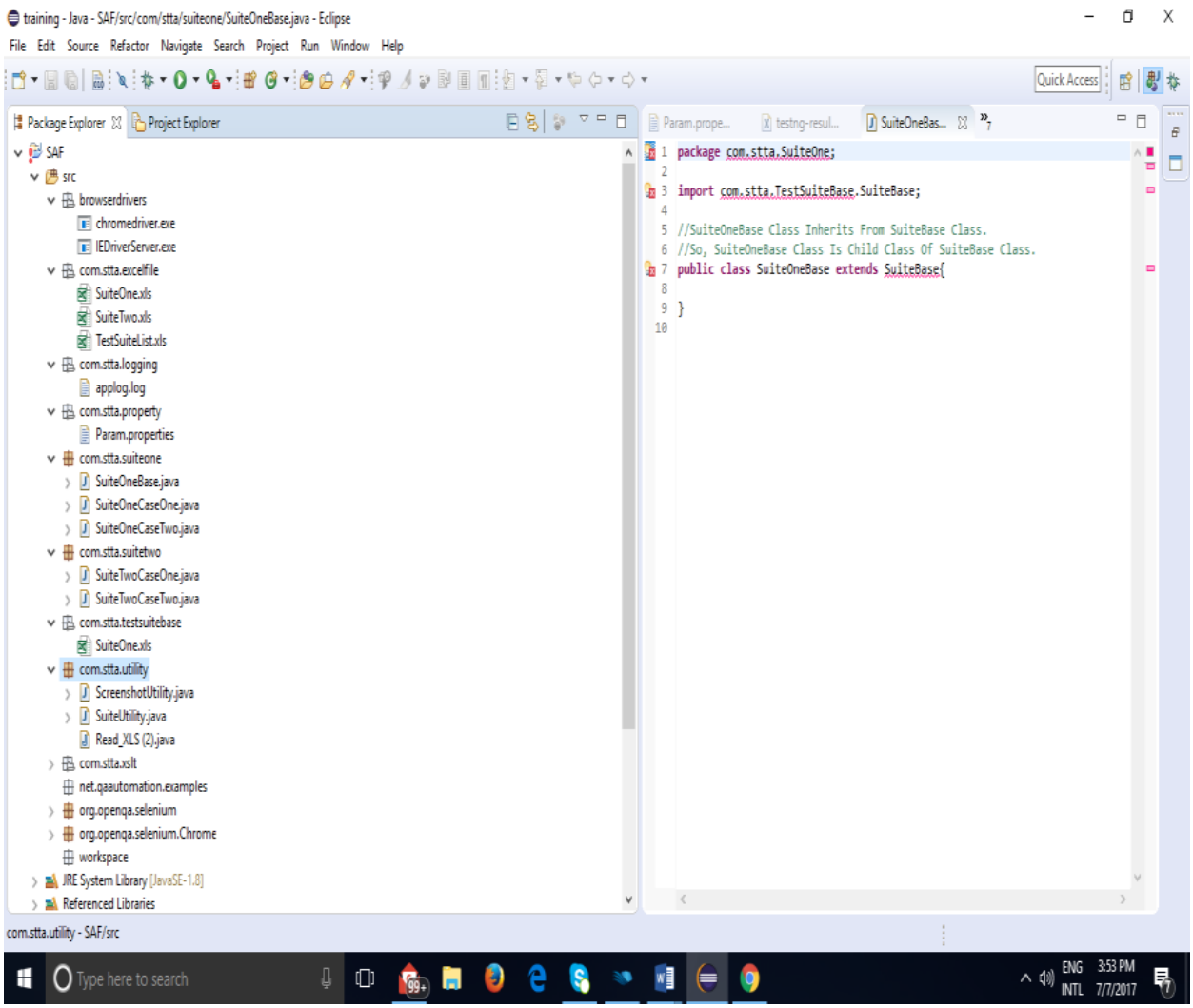
if the browsers are upgraded to latest versions for Firefox (example 47 and above) which will not be compatible to the automation framework constructed as part of this Poc .similarly change in operating system from win 2007 to latest version may impact the current automation framework built for the Poc .

Problems in automation

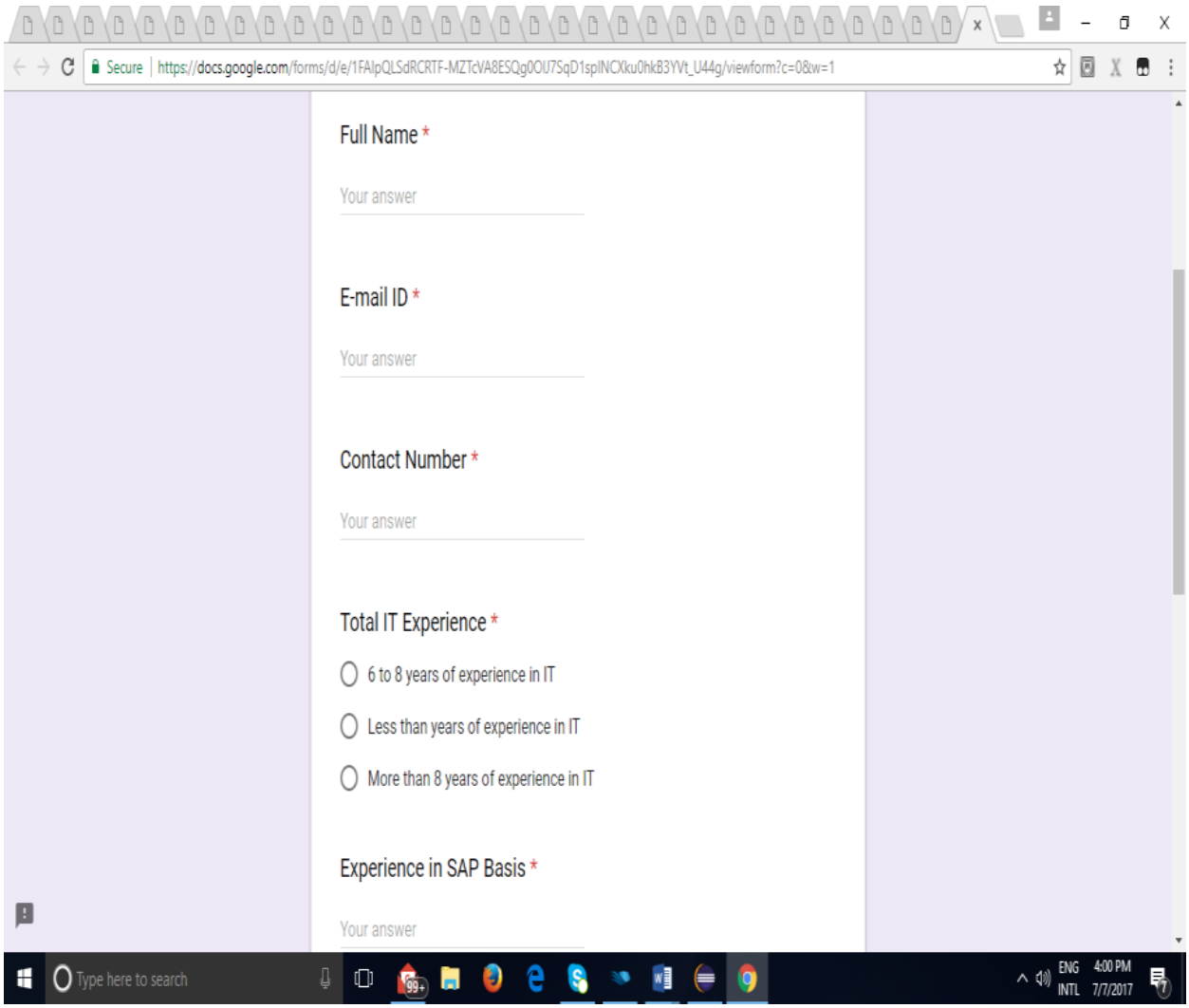
It is not possible to automate the application other then web pages in selected selenium based automation framework .

Methodologies for Poc approach

Download the eclipse and download the jar file FROM THE
SELENIUMHQ WEBSITE and build into eclipse LIKE this below image



The form create from google driver used to automation proof of concept
https://docs.google.com/forms/d/e/1FAIpQLSdRCRTF-MZTcVA8ESQg0OlJ7SqD1spINCXku0hkB3YVt_U44g/viewform?c=0&w=1



Following line of codes are written to automate the target web page:

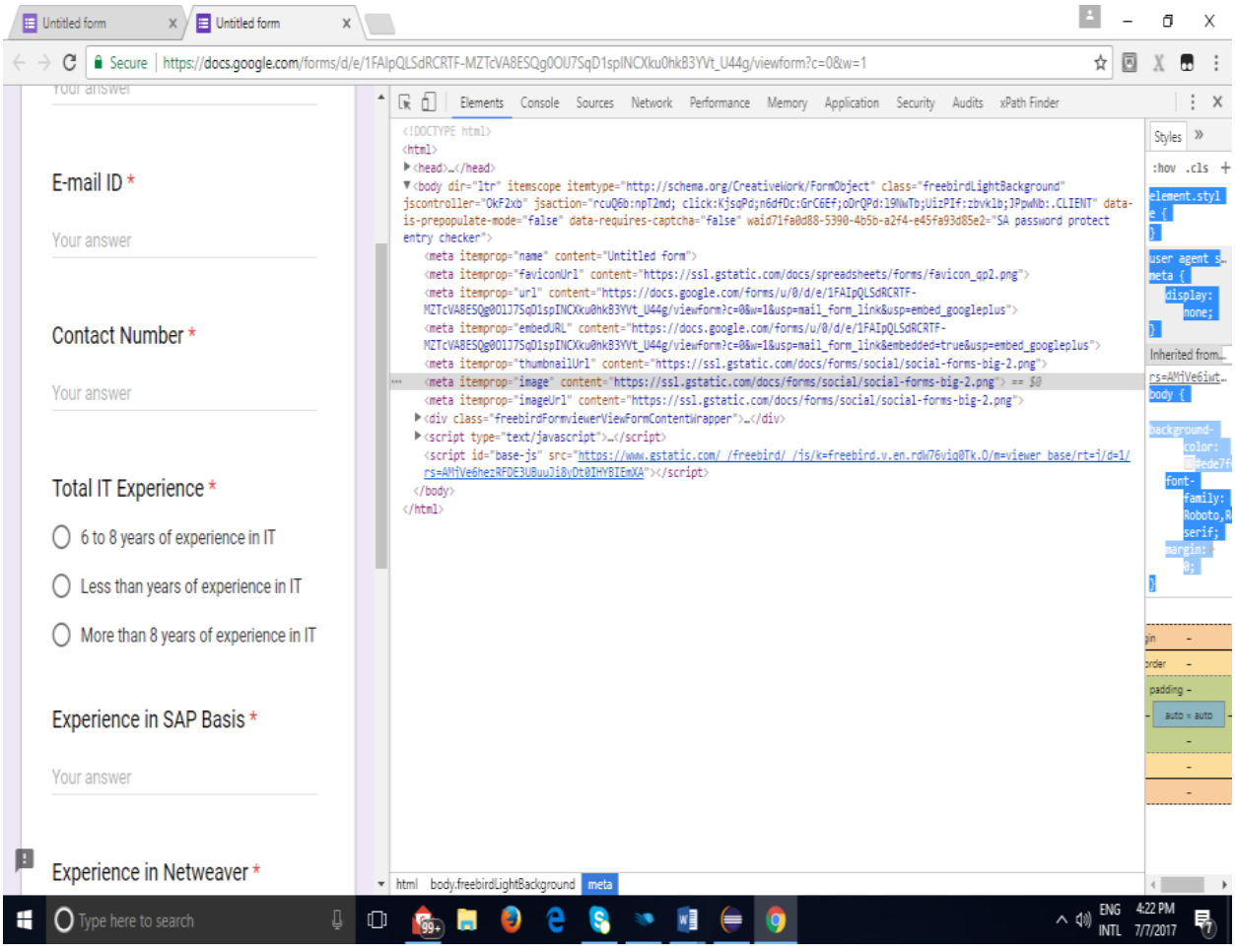
```
package com.stta.SuiteOne;
import java.io.IOException;
import org.openqa.selenium.By;
import org.testng.SkipException;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeTest;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;
import com.stta.utility.Read_XLS;
import com.stta.utility.SuiteUtility;
//SuiteOneCaseOne Class Inherits From SuiteOneBase Class
//So, SuiteOneCaseOne Class is Child Class of SuiteOneBase Class and SuiteBase Class.
public class SuiteOnceCaseOne extends SuiteOneBase{

Read_XLS FilePath=null;
String SheetName=null;
String TestCaseName=null;
String ToRunColumnNameTestCase=null;
String ToRunColumnNameTestData=null;
String TestDataToRun[]=null;
static boolean TestCasePass=true;
static int DataSet=-1;
static boolean Testskip=false;
static boolean Testfail=false;
SoftAssert s_assert=null;
```

```
@BeforeTest
public void checCaseToRun() throws IOException{
//Called init() function from SuiteBase class to Initialize .xls Files
init();

//To set SuiteOne.xls file's path In FilePath Variable.
FilePath=TestCase.ListExcelOne;
TestCaseName=this.getClass().getSimpleName();
//SheetName to check CaseToRun flag against test case.
SheetName="TestCasesList";
//Name of column in TestCasesList Excel sheet.
ToRunColmnNameTestCase="CaseToRun";
//Name of column In Test Case Data sheets.
ToRunColumnNameTestData="DataToRun";
//Below given syntax will Insert log In applog.log file.
Add_Log.info(TestCaseName+":Execution started.");
//To check test case's CaseToRun=Y or N In related excel sheet.
//If CaseToRun=N or blank,Test case will skip execution.Else It will be executed.
if(!SuiteUtility.checkToRunUtility(FilePath,SheetName,ToRunColumnNameTestCase,TestCaseName)){
Add_Log.info(TestCaseName+":CaseToRun=N for So Skipping Execution");
//To report result as skip for test cases In TestCAsesList sheet.

SuiteUtiity.WriteResultUtility(FilePath,SheetName,"pass/Fail/Skip",TestCaseName,"SKIP");
//To throw skip exception for this test case.
throw new SkipException(TestCaseName+"'s CaseToRun Flag is 'N' Or Blank. So Skipping Execution
Of"+TestCaseName);
}
//To retrieve DataToRun flags of all data set lines from related test data sheet.
TestDataToRun=SuiteUtility.checkToRunUtilityOfData(FilePath,TestCaseName,ToRunColumnNameTestData);
}
//Accepts 4 column's String data in every iteration.
@Test(dataProvider="SuiteOneCaseOneData")
Public void SuiteOneCaseOneTest(String DataCol1, String DataCol2, String DataCol3, String DataCol4,
String DataCol5, String DataCol6, String DataCol7, String DataCol8,String ExpectedResult)
{
DataSet++;
//Created object of TestNG SoftAssert class.
S_assert=new SoftAssert();
//If found DataToRun="N" for data set then execution will be skipped for that data set.
If(!TestDataToRun[DataSet].equalsIgnoreCase("Y")){
Add_Log.info(TestCaseName+":DataToRun=N for data set line"+(DataSet+1)+"So skipping its
execution.");
//If DataToRun="N",Set Testskip=true.
Testskip=true;
Throw new SkipException("DataToRun for row number"+DataSet+":No or Blank.So skipping its
execution.");
}
//If found DataToRun="Y" for data set then below given lines will be executed.
//To convert data from String to integer

//int ValueOne=Integer.parseInt(DataCol1);

//int ValueTwo=Integer.parseInt(DataCol2);
//int ValueThree=Integer.parseInt(DataCol3);
//int ExpectedResultInt=Integer.parseInt(ExpectedResult);
//To initialize Firefox browser.
loadWebBrowser();
driver.get(Param.getProperty("siteURL");
driver.findElement(By.xpath("id('entry_1000000')")).sendKeys(DataCol1);
//getElementByName("Laevistrombus canarium").click();
driver.findElement(By.xpath("id('entry_1000013')")).sendKeys(DataCol2);
```

```
driver.findElement(By.xpath("id('entry_1000014')")).sendKeys(DataCol3);
driver.findElement(By.xpath("id('group_1000001_2')")).click();
driver.findElement(By.xpath("id('group_1000001_2')")).click();
driver.findElement(By.xpath("id('group_1000001_3')")).click();
driver.findElement(By.xpath("id('entry_1000002')")).sendKeys(DataCol4);
driver.findElement(By.xpath("id('entry_1000003')")).sendKeys(DataCol5);
driver.findElement(By.xpath("id('entry_1000004')")).sendKeys(DataCol6);
driver.findElement(By.xpath("id('entry_1000005')")).sendKeys(DataCol7);
driver.findElement(By.xpath("id('group_1000006_1')")).click();
driver.findElement(By.xpath("id('group_1000006_2')")).click();

driver.findElement(By.xpath("id('group_1000006_3')")).click();
driver.findElement(By.xpath("id('group_1000007_1')")).click();
driver.findElement(By.xpath("id('group_1000007_2')")).click();
driver.findElement(By.xpath("id('group_1000008_1')")).click();
driver.findElement(By.xpath("id('group_1000008_2')")).click();
driver.findElement(By.xpath("id('entry_1000009')")).sendKeys(DataCol8);
driver.findElement(By.xpath("id('group_1000011_1')")).click();
driver.findElement(By.xpath("id('group_1000011_2')")).click();
driver.findElement(By.xpath("id('group_1000011_3')")).click();

driver.findElement(By.xpath("id('group_1000011_4')")).click();
driver.findElement(By.xpath("id('group_1000011_5')")).click();
driver.findElement(By.xpath("id('group_1000011_6')")).click();
driver.findElement(By.xpath("id('group_1000011_7')")).click();
driver.findElement(By.xpath("id('group_1000011_8')")).click();
driver.findElement(By.xpath("id('group_1000011_9')")).click();
driver.findElement(By.xpath("id('group_1000011_10')")).click();
driver.findElement(By.xpath("id('group_1000011_11')")).click();
driver.findElement(By.xpath("id('group_1000012_1')")).click();

driver.findElement(By.xpath("id('group_1000012_2')")).click();driver.findElement(By.xpath("id('group_1000012_3')")
).click();
driver.findElement(By.xpath("id('group_1000012_4')")).click();
driver.findElement(By.xpath("id('group_1000012_5')")).click();
driver.findElement(By.xpath("id('ss-submit')")).click();
String Result=getElementByName("txt_Result").getAttribute("value");
//To convert data from string to integer
Int ActualResultInt=Integer.parseInt(Result);
//Compare actual and expected values.

If(Testfail){
//At last, test data assertion failure will be reported In testing reports. It will mark your test data,test
case and test suite as fail.
S_assert.assertAll();
}
}
//@AfterMethod method will be executed after execution of @Test method every time.
@AfterMethod

Public void reporterDataResults(){
If(Testskip){
Add_Log.info(TestCaseName+":Reporting test data set line"+(DataSet+1)+"as SKIP In excel.");
//If found Testskip=true,result will be reported as SKIP against data set line in excel sheet.
SuiteUtility.WriteResultUtility(FilePath,TestCaseName,"Pass/Fail/Skip",DataSet+1,"SKIP");
}
else if(TestFail){
Add_Log.info(TestCaseName+":Reporting test data set line "+(DataSet+1)+"as FAIL in excel.");

//To make object reference null after reporting in report.
S_assert=null;
//Set TestCasePass=false to report test case as fail in excel sheet.
```


TestCasePass=false; //If found TestFail=true,Result will be reported as FAIL against data set line in excel sheet.
SuiteUtility.WriteResultUtility(FilePath,TestCaseName,"Pass/Fail/Skip",DataSet+1,"FAIL");
}else{
Add_Log.info(TestCaseName+": Reporting test data set line"+(DataSet+1)+"as PASS In excel.");
//If found TestSkip=false and TestFail=false,Result will be reported as PASS

against data set line in excel
sheet.
SuiteUtility.WriteResultUtility(FilePath,TestCaseName,"Pass/Fail/Skip",DataSet+1,"PASS");
}
//At last make both flags as false for next data set.
Testskip=false;
Testfail=false;



Data driven framework methods are used to design the automation framework and web Elements and capture using xpath methodologies

advantages of Poc
performing poc provides a better understanding on automation framework implementation creating implementation creating the framework during Poc phase helps in estimating the return on investment calculations with the following inputs

Testing Activity	Before PoC	Aftet PoC
writing a test case	30 minutes	40 minutes
Test execution	20 minutes	0.3 minutes
Test amndments	10 minutes	10 minutes

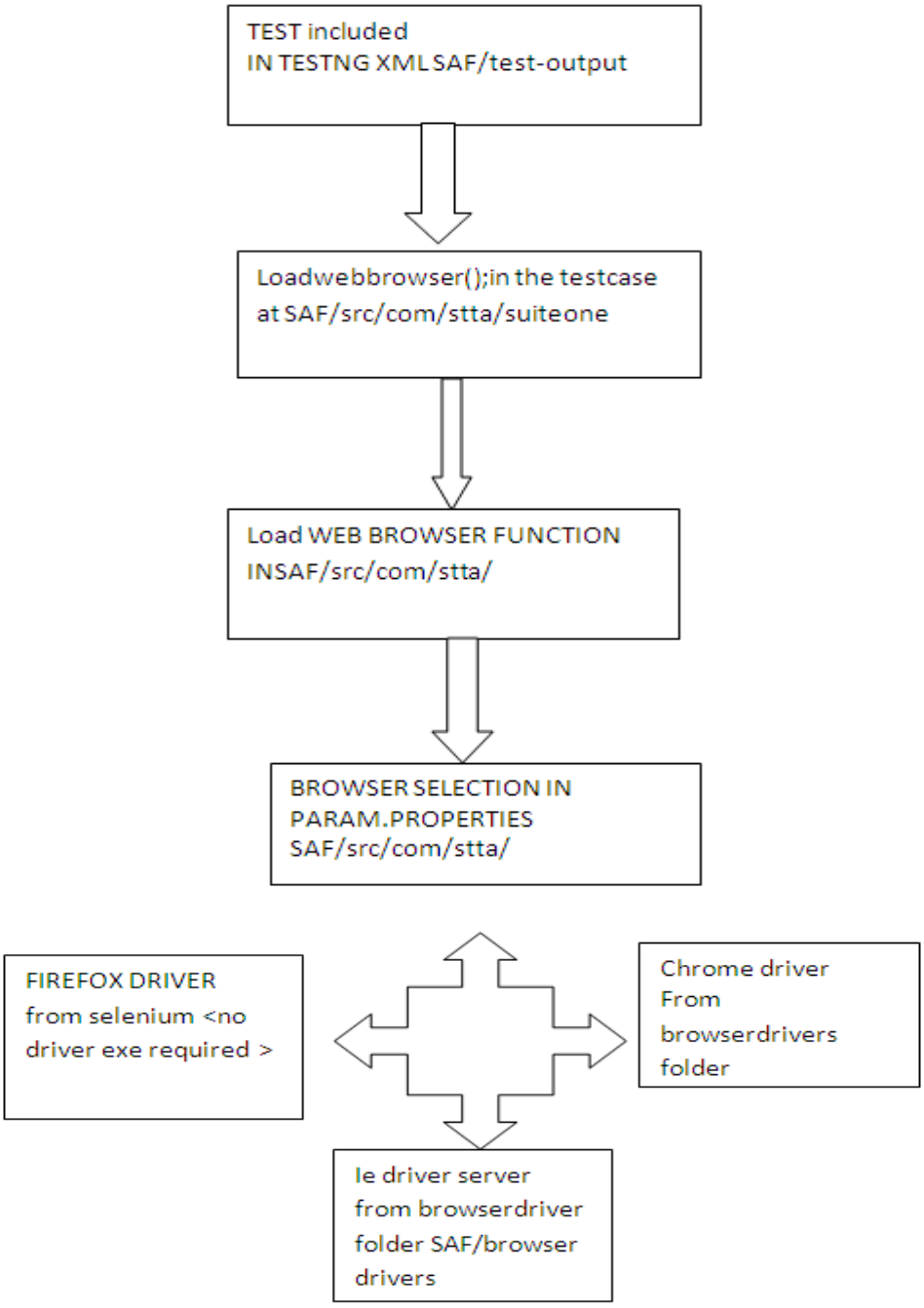
Entry criteria
Objects of web elements should be made available from dev team test environment should be available to perform Poc

Exit criteria
Once PoC is designed and presented without open defects within automation framework successfully , the Poc phase of scripting
Test architecture

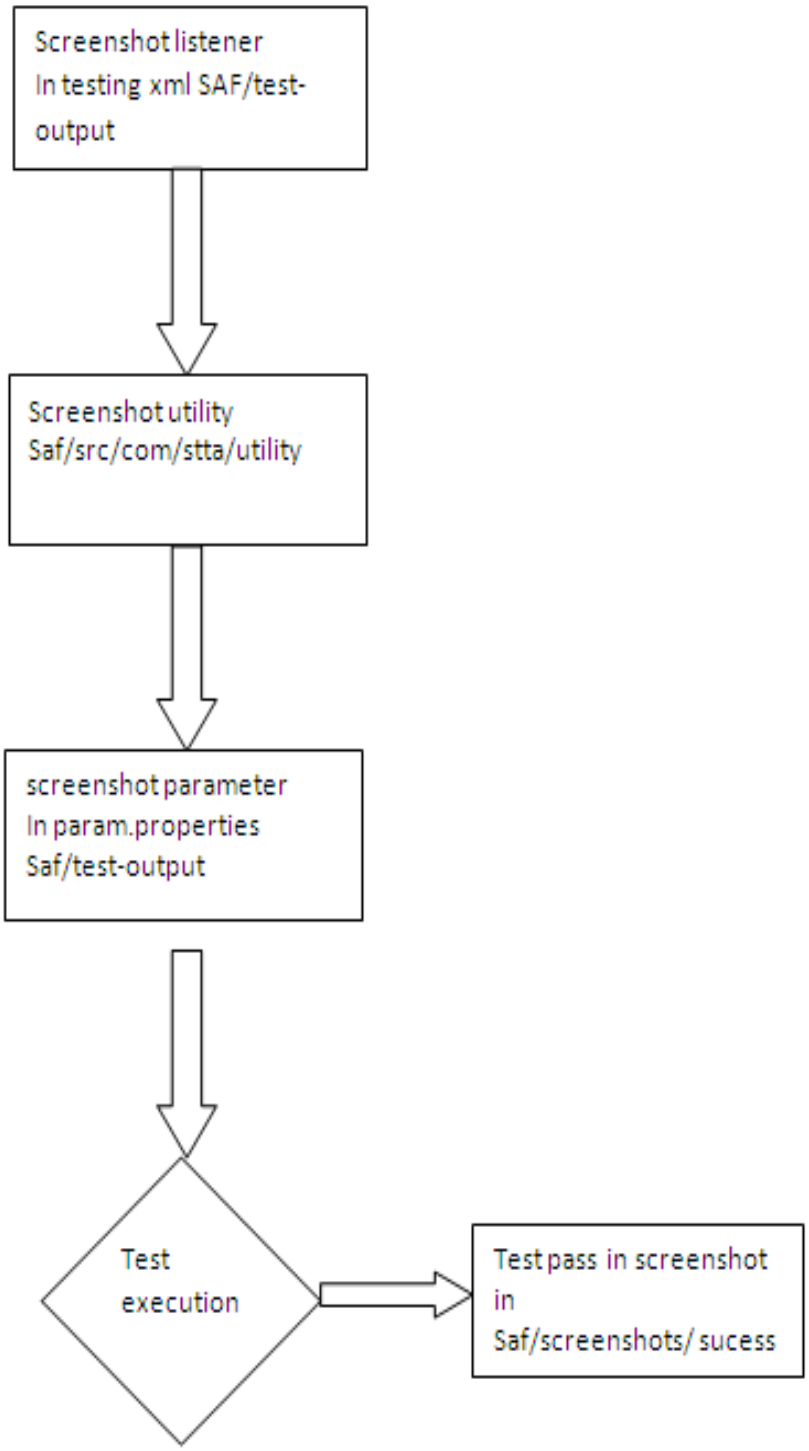
- • To design a system for coders you must understand how to code (and be coding)
- • You can't design a system without being aware of what is happening at ground level

Architecture is not just about broad stroke design but about adapting to changing needs at the code level

TEST Architecture : cross browse compatibility

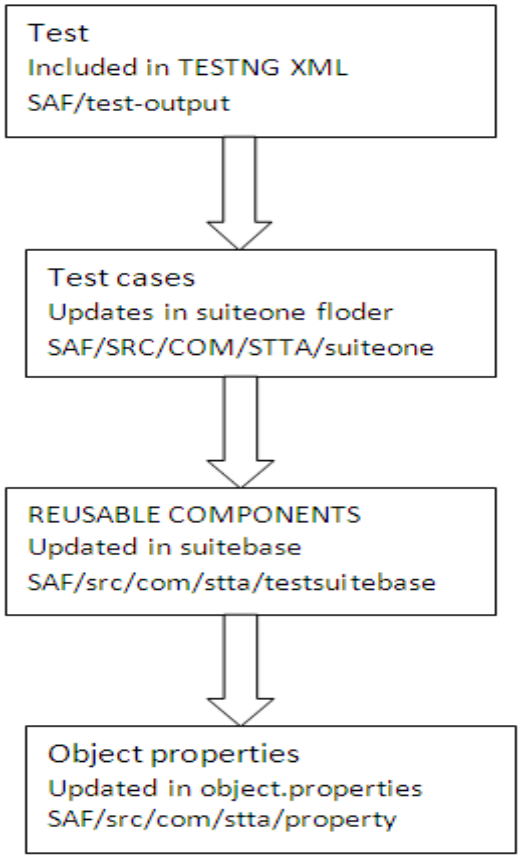


Screenshot capability of automation framework

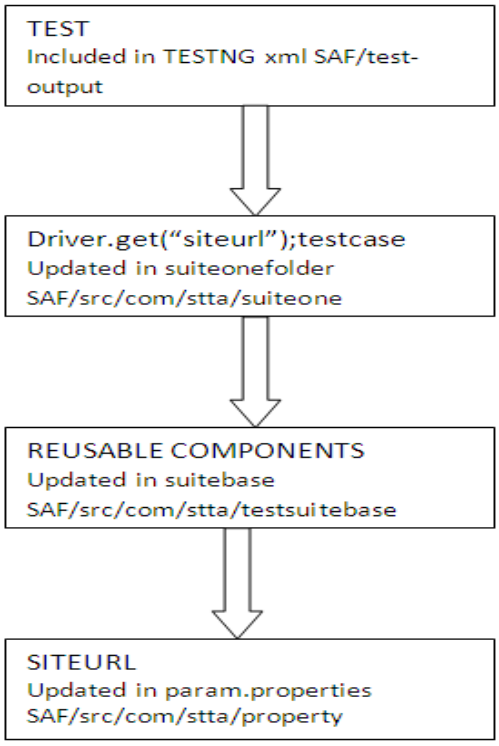


CBJECT REPOSITORY CAPABILITY automation framework

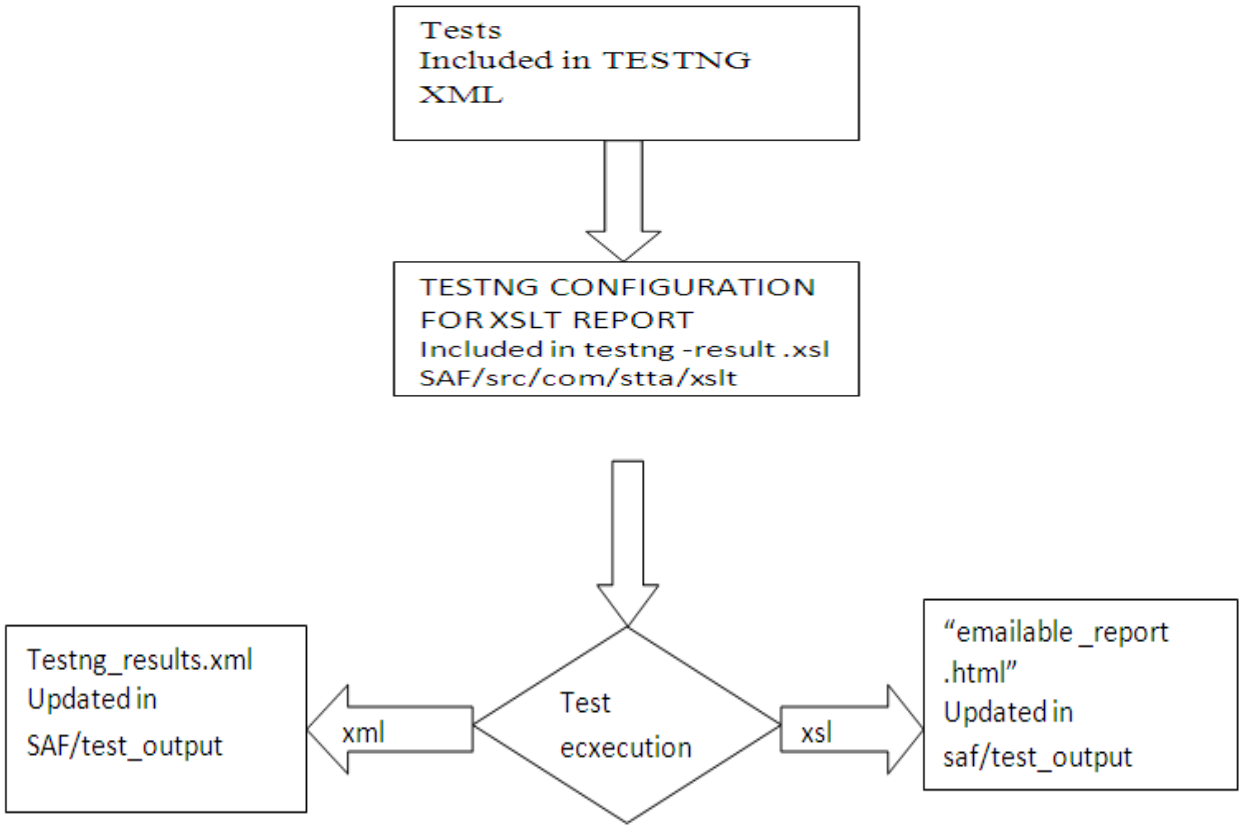
<test name =”suiteonecaseone”>
<classes>
<class name =”com.stta.suiteone.suiteonecaseone”></classes>
</test>



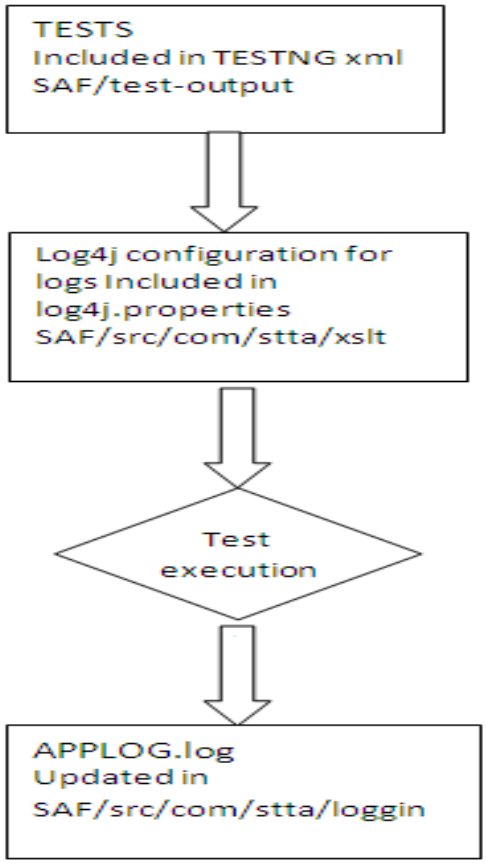
SITEURL MANAGEMENT USING PARAMETER <test name =”suiteonecaseone”><classes><class name =”com.stta.suiteone.suiteonecaseone”></classes></test>



XSLT reporting capability



Logging capability automation framework



Type of feasibility	Possible ?
Screenshot capture	Yes
Log4j integration	Yes
Parameterisation	YES
Core component framework development	YES
Cross browser compatibilty (ie ,chrome,firefox)	YES
XSLT reporting	YES
Data input from excel files	YES
Resuable function library	YES
Test suitehierarchy	YES
TESTNG EXECUTION	YES
ANT BUILD	YES
MAVEN BUILD	YES
JENKINS JOB SCHEDULING	YES
Web automation Standalone automation	YES
Android automation	NO
Standalone automation	NO

DESIGN FOR SUTAINABILITY

SOFTWARE	SUPPORTED	NOT SUPPORTED
SELENIUM	2.45.0	3.0 BETA
FIREFOX	33.0 TO 46.0	47+
IE	7,8,9,10	6,11
CHROME	2.0	LATEST BETA VERSION

PROVE OF THE CONCEPT:
Since web automation in firefox 46b,ie8 and c webdriver 2.45.o version using java 1.8 (jdk) implementation

Advantage of poc

Performing poc provides a better understanding on automation framework implementation creating the framework during poc phase helps in estimating the return on investment.

REFERENCES:

Website: http://docs.seleniumhq.org/download/

Book: Foundations of Software Testing by Rex Black, Erik Van Veenendaal, Darothy Graham (third Edition)

