Implementation of Selenium Automation & Report Generation Using Selenium Web Driver & ATF

1st *Er* Khushboo Sawant
Department of Computer Science and Engineering
Lakshmi Narain College of Technology(R.G.P.V)
Indore, India
sawantkhushboo@gmail.com

2nd Reetu Tiwari
Department of Computer Science and Engineering
Lakshmi Narain College of Technology (R.G.P.V)
Indore, India
tiwarireetu1995@gmail.com

3rd Swapnil Vyas
Department of Computer Science and Engineering
Lakshmi Narain College of Technology (R.G.P.V)
Indore, India
Er,Swapnilvyas@rediffmail.com

4th *Pawan Sharma*
Sr. Software

Consultant
Profinch Solution Pvt Ltd
Pune, India
er.pa1.sharma@gmail.com

5th Aryan Anand
Department of Computer Science and Engineering
Lakshmi Narain College of Technology(R.G.P.V)
Indore, India
aryananand922@gmail.com

6th Shivani Soni
Department of Computer Science and Engineering
Lakshmi Narain College of Technology(R.G.P.V)
Indore, India
shivanisoni@gmail.com

***Abstract*—Among the most specific & forthcoming domain for exploration is testing using automation tool. There is numerous software systems that has been implemented as the browser applications which are complex to test. Automation and various Automation tools reduce the human intervention and iterative process of testing various web applications. There are various tools for automation like Junit, Chakram, and Selenium for performing Automation Testing.**

The proposed solution usage the resource monitoring technique the defected cases are detected. The proposed testing is implemented and tested in jerkin testing environment. In order to debugging the testing performance using two different testing scenarios the performance is compared. And the comparative performance study is performed in terms of debug ratio, throughput, and fail test cases ratio and memory consumption. The obtained results demonstrate the effective performance with respect to the traditional testing algorithm.

***Keywords*— Web-driver, J-unit, Test Suit, Automation based testing, selenium based web driver, Automation testing report.**

I. INTRODUCTION

Selenium IDE is an publicly available testing script generation tool that is used for report cases testing. Selenium tool is a set of many software modular Tools with a better mode to carry the scripts automation. Many browsers contain power to bridge all over laitance modular resources & completely. It boost many modular designing & coding languages, which integrate mostly, even so not limited to solely, Python, Perl, Java, Groovy, Ruby, C and PhP. Selenium web driver exist of many factors which accumulations three critical taps.[4] Everybody has a elect function in assisting the group action of scripts automation for a web based application.

Add on to software testing are teething oftentimes because of to many advantages of modular quality assurance over traditional tradition models used for testing the modules [11]. In this, agile software model are joined along-with a traditional approach for act each modules. Generally, agile methodologies may be performed in two or more techniques along scrum & crystal method with features driven development. Software quality can be improved by using extreme programming and also is defined by its own characteristics; it is planning, managing, coding, and designing where debugging info of test cases are dynamically created at run time. Because of the weak quality of the agile software testing & debugging it is yet an free Zone of latest research methodology & development. Only because of development to new tools of testing many critical problems are occurred in such type of research-technology i.e. benefit and protection [3].The target of our research is, mostly, to read on the performance-factors of previous testing and & striking on the failed test cases during testing & debugging process. And so, our ordinal aim is have to compare the outgrowth of new tools with previous one through which modules are easily tested. The analysis of this research-work can then be carry, in a near future-enhancement, as a direction to program newest web driver tool basically for debugging & testing.

978-1-7281-8586-6/21/$31.00 ©2021 IEEE

Selenium web driver RC is a server, drop a line in with Java programming that is very first automated web testing tool by accepting commands by using different protocol that permit clients to perform CRUD operation with selective web application based user interface related UI scripts in many more modular languages like , Perl, Java, Ruby C#, PHP, Python to deal with complex test scripts[5]. In selenium web driver like permits & allows for proper combination of Selenium web driver in previous unit test scripts frameworks at the same duration &then execute the updated test-cases in parallel way. Selenium web driver tool RC not finite to choosy gateway it hold up most of newly chrome web browsers thus client can execute many test suit along with own adaptable programming modular language this assist in adding to the performance of test-suit wrapping all the test suit operations and assist & fail methods. Selenium web driver RC create more modular & programming based languages. An objects of selenium RC server is help to update html test suit – which adapt that the port number may be unlike for every concurrent program that execute like Python, Perl, Ru

II. LITERATURE SURVEY

Test automation tools cannot be bringing in with-out very nice automation tools; as they find how automation tool is executed and whether the profits of automation tool can be delivered to its end user. Test script automation tools are pivotal parts in the DevOps in tool chain. The present test scripts automation direction have growing in put in deep learning, data Analyst & artificial intelligence and machine learning (AI/ML) to provide real world abilities for test cases enhancement optimization, test cases intelligent test generation, execution & reporting generation. It will be beneficial to recognize which automation tools are finest clam to grab lead of these current trendy.

In [2]paper author conclude & Compare many Web dependent many test generation based Tools & analyzed Selenium web driver is the good feasible open-source automation tool for web based petition among all that can be freely available. Testing can be performed by using some test cases, for that test script have been written for each module and tested separately. The proposed tools are based on test cases so that the profitable connections of web based request growing the significance of managing and upgrading its grade [1]. The focal edge of using automated testing tools is that we can ignore the guidebook try need to test each suit of our web based site by generating the test scripts. This paper used flowing preliminaries:

.

In this paper the authors have worked on the masked face detection from the video. The masked person is detected in this presented approach and mainly 4 steps are performed for the detection that are estimation of distance between camera and person, detection of eye line, detection of part of face and detection of eye. They have analyzed their algorithm on various video surveillance systems and achieved a fine accuracy [9].

In this paper author compared two testing tools in context of web based application. Testing tools have been chosen on the basis of complexity & functionality of an web application. For parallel execution of test cases selenium grid is used along with selenium tool for better efficiency. UFT is machining depended tools its mean it can run scripts on single machine at time. So it will degrade performance issue of the tool. Author also examine that UTF provide its own objects repository whereas selenium used HTML dom structure for recognition object. The impact of selenium tools is better than QTP under many complex scenarios. The inner working is determined hold on to degrade on the conditions of some of important points such as total outcome of test suit script run parallel, run duration & time, iteration process of test suit cases of, and the evolution of normal run test suit scripts [30].

In this paper [3] successfully testing of each modules of online web application by using automation selenium testing tool without any failure. User interaction with browsers has been recorded by tester on the basis of test script. In this research, author review the test-cases design & working of the web based selenium testing tool and need it to run some online web based application applications. In process to test an web based application, testing testers do not to learn the all selenium web driver tool completely. This tool is needful for a technical lead and tester; they can design their source code owing to review shot properties of framework IDE. It created the adapted test suit cases report to the testing tester. It is very well to maintain & update the fail-pass test suite case for latest version of the web based application by selenium tool. The documents name is in Rupess, in the Rs_Test( ) process, text connector operations such as above, capitalized, lower, sentence & alternate have been focused on the given keys & values. The editor based translator part of the online web based application has strongly converted the keys & values. There is no non-success in the web based site's application & this has been tested properly and hence proved by the web based on Selenium testing tool suit[32].

In paper [14] author conclude that liability is module testing is part of software quality and software assurance. To check the functionality of web page manual testing is performed .author also concludes that both the manual & automation testing flow same work procedure. In IDE testing have been used for generating test cases. Automation regression has been created once website is finalized. Manual testing is highly recommended as compared to automation testing. For conduct research on test Google tool have been used on manual & automation testing. Most of research has been done in previous decade on manual & automation testing.

In Paper [16] Conclude that multiple test scripts can be executed concurrently by using data driven automation. Various test cases have been proposed in conjunction with selenium test tool. In work authored summarized that manual dependency have been deduced as compared to automation. Continuous integration has been utilized to achieve the objective with respect to time, space & Complexity.

Extraordinary endeavor have been expert for testing the purpose of the software modules. Some testers defined the methods for test the script but bugs finding also can not able to handle the bugs. In the integration of modules some time create confusion [21] [22]. Build is the basically putting source code and provide precise results of integration modules which consist of compilation, deployment, inspection. Schedule of build has been decided on the basis of type of build.

III. Problem identification & Proposed approach

In this presented work a solution for report generation (executing the test scenarios) is introduced. While executing the code we get the reports with a bar graph designed with the specifications on failed and passed test cases but this is not in readable format once we are delivering the proposed work to client. Thus a UI friendly report format is required. This presented work demonstrates a mechanism to generate User friendly reports so that the deliverables can be easily approvable to clients.

Debugging in testing is processes of finding defects known as bugs are connected with test cases that can be identify error called bugs. Before testing begins effective testing is require for successful test case. Therefore for effective testing complex module is subdivided in to sub module and each module have significant advantage and execution steps. In code section by using if statement, error message can be printed so that we can get trackback information of bugs occurred during execution. These debugging steps are try to find an minimal solution in-between error & bugs. Therefore sanity checking is performed for early detection of bugs.

The testing for the application is done by the help of the designing tool called as testing. Whenever we are recording any of the test script using the testing class it is the focal & starting area of instances for executing suit tests in the TestNG IDE framework. Clients may make their own TestNG instance & call it in more different possible ways like testng.xml or command line or from Java for example we can see the most important file is TestNg.java. It doesn't matter where you are invoking testing because it can be invoked in many different ways. You can also conclude which instances to include & exclude, given parameters, etc. If we are planning to add Listener in our test the method by TestNG we need to use the respective listener which is nothing but an method accepting ITestNGListener interface variable. This method checks which instance is passed in your ITestNGListener variable and based on that decide what to implement in Test.testing.run() it is the final call to run the Testcases and Test Suite i.e performed by the run() method. TestNG call the method in which there are multiple calls to other method like sanityCheck(), runExecutionListeners(), List<ISuite> suiteRunners = runSuites(); etc. Likewise private void generateReports(List<ISuite> suiteRunners)generateReports() method is also very important method call from the run() method and this method receives a List of ISuite Interface which is used to create test case report at the end of the Suite excecute.Unfortunately Selenium tool Grid is a part of the Selenium Suite that overspecializes in executing many tests cases across different web browsers, different dependent operating systems, and machines in parallel execution.

. Therefore during the finding defects test cases are altered or modified for preparing long execution or misguiding the testing scenario. In addition of that the defected test case are making frequent connectivity from the entire modules in environment using false control message exchange. Due to these connected module test case methods replies the false request for connectivity and draining execution speed rapidly. Therefore in order to detect and prevent the test case in testing environment a new kind of scheme is required which monitor the IDE debugging activity and provide the decision for bugs occurring behaving test.

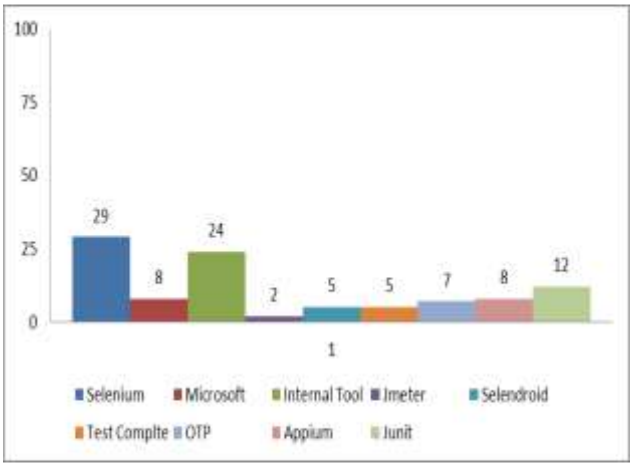


Fig 1: Performance Matrices for test automation

IV. IMPLEMENTATION

Selenium Driver makes at the first hands make calls to the web browser utilizing each browser's endemic assist for test script automation. There are so many web browsers and many technology there is require for common identification given by WebDriver API. TestNG is a testing IDE in the code up with test suit of JUnit and N-Unit, moreover it define some latest characteristics evolve that build it more strong & simple to utilize. In automated testing framework TestNG is an open source. TestNG is like to JUnit, unfortunate it's not a J-Unit extension. It's stimulated by J-Unit. Its plan to be preferable than J-Unit, mainly when testing unified classes. TestNG can create test reports ground on our Selenium test script outcomeAfter contrive. Now we have holdup down mail.jar for broadcasting the E-mail to the specific user. Mail.jar need may multiple protocols similar such as POP3, SMTP. After many test script suite we have support the mail report () method to mail the test cases suit.

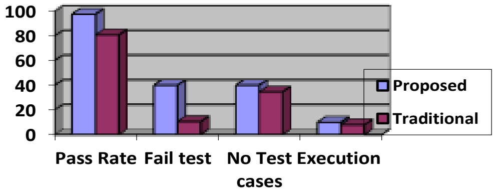


Figure 2: Proposed Scenario

After utilize of execute agile testing IDE, the testing perormance is mainly build on. Technical lead code the test script two times than traditional method of writing test suit script cases. This reduces the user system resource need to test the websites paging application. The updating price of suit script cases also overcome because of centralized system based modular repository. As the updated version of web based application challenges, we support to require change only object based repository for newly created elements. Many times test script cases are flaw down due to synchronization causes of selenium web driver not because of web application bugs defects. This framework overcomes the bug's rate of failing down the test script cases because of synchronization flaw. Eventually flow rate will grow. This manifest the correctness of proposed approach framework over old approach of testing. We have run test cases suite of 350 test script cases on data information web based application. After execution of run we got the following results in terms of overall pass –fail rate, run time etc. Passing rate represent that test script case gives us the best outcome as manual test script case gives. Sometimes web based application works exactly but test cases fail because of synchronization. Proposed approach framework synchronizes the test script cases systematically, so fail rate is overcome than old approach.
Report is customized on the report of organization require; this upgrade the testability of test cases report in Fig.3. After execution run is over customized test are sent to customer stakeholders. Stakeholders can analyse details pass- failure if possible. Screenshot provides an better way to tester to fix their bugs by clicking over screenshot web link in test script [22].

Figure 2 compares the pass fail loss over bugs environment of testing cases with fixed script cases with fixed number of rounds. IDE includes both legitimate and malicious bugs. Table 5.1 Conclude that pass-fail Loss over the proposed approached is 10% and remains tested pass is 90 %.whereas under bugs approached fail cases Loss is 35% & remains pass-fail is 65%. It decreases the pass; failed loss control over head by only 10 % with the presence of bugs [22].

After utilize of execute automation testing IDE, the testing efficiency is mainly build on. Tester write's the test script two times than traditional approach of writing test script cases. This reduce the human system] resource need to test the web based application. The updating price of test script cases also overcome because of centralized based repository. As the updated version of web based application challenges, we support to require change only object based repository for newly created elements. Many times test script cases are flaw down due to synchronization causes of selenium web driver not because of web application bugs defects. This framework overcomes the bug's rate of failing down the test script cases because of synchronization flaw. Eventually flow rate will grow.

This manifest the correctness of proposed approach framework over old approach of testing. We have run test cases suite of 350 test script cases on data information web based application. After execution of run we got the following results in terms of overall pass –fail rate, run time etc. Passing rate represent that test script case gives us the best outcome as manual test script case gives. Sometimes web based application works exactly but test cases fail because of synchronization. Proposed approach framework synchronizes the test script cases systematically, so fail rate is overcome than old approach. There are basically two methods used **@Before** & **@After.** They are typically require for control and tear & down of the native environment before & after each event.

@Before - Before execute before the first process of each event. This is mainly required for prerequisite process that has to be performed before the actual test case event.

- Features
  1. login.feature
  2. validateCRM.feature
- Grid
  1. chromedriver.exe
  2. geckodriver.exe
  3. hub.bat
  4. hub.json
  5. node1.bat
  6. node1.json
  7. node2.bat
  8. node2.json
  9. selenium-server-standalone-3.12.0.jar
- Logs
  1. applog.txt
- Properties
  1. Config.properties
  2. log4j.properties
  3. OR.properties
- Runner
- **1. Initialize a web driver**
- **2. Create Data Base connections**
- **3. To set-up test case data**
- **4. To set web browser cookies**
- **5. Propagate to default session page**
- **@After** - After method execute after the last process of each event, even in case of when

process are failed, , pending, undefined, skipped. This is commonly method require & used for process that have to be performed after the actual events gets properly run.

- **1. Stop the web driver:**
- **2. To close Data Base connections from pool:**
- **3. To flush the test case data & browser cookies:**
- **4. Session out from the all the web application**
- **5. Capture screenshots for fail & pass test cases scenarios:**

Algorithm finding bugs

A. CREATING THE Fail PASS SENARIO

Following code to create fail/pass by using store procedure & function.

```
@AfterClass
@BeforeClass
@Test
@AfterMethod
@BeforeMethod
set ns [new Environment]
#open the nam test file
set nf [open test.nam w]
$ns namtrace-all $nf
#define a 'finish' procedure
proc finish {}
{
   global ns nf
   $ns flush-test
   #close the test file
   close $nf
   #execute nam on the test file
   exec nam test.nam &
   exit 0
 }
#create test
set t0 [$ts test1]
set n=t1 [$ts test2]
```

Table 1: Performance Evloution

S. No.	Parameters	Proposed	Traditional
1	Maintenance Cost	Low	High
2	Pass Rate	High	Low
3	Fail Rate	Low	High
4	Execution Time	Low	High

This section, performance matrices Like Pass Ratio, End to End delay, Execution time and fail ratio are estimated in testing environment with testing Parameter, presented in tabular form.

V. RESULT ANALYSIS

This approach great for web page loads application like CONS: Doesn't require for AJAX based calls application. AJAX based calls application are easily & simply added to the overall web page load time & the method like getEvents() call is not execute in Mozilla Firefox for our application to recreate to manually basis calculate the ajax calling time.

After contrive of test script, it require to mail to individual power.

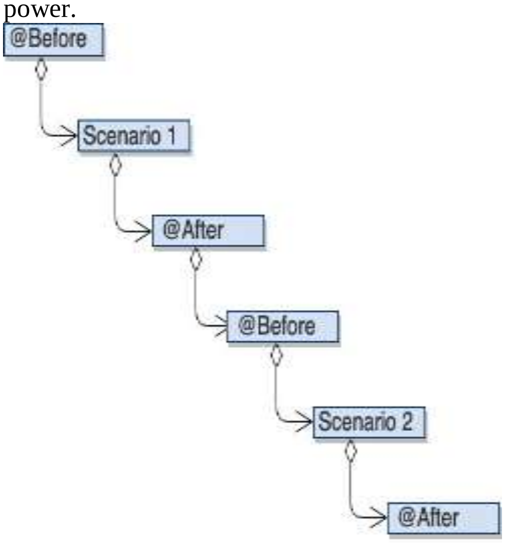
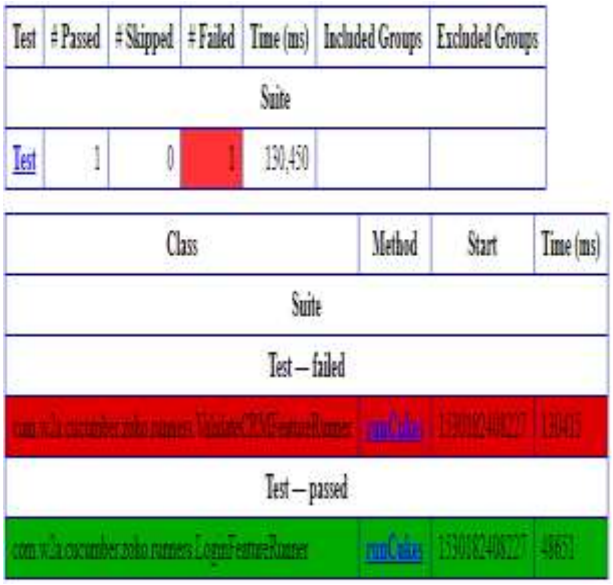


Figure 3. Running Scenario

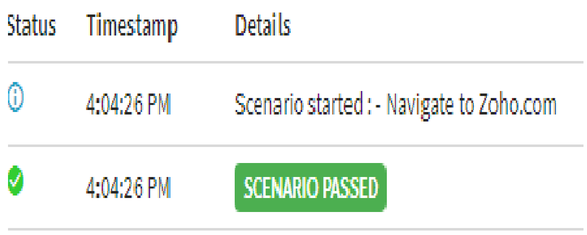
We have fortunately run our Proposed Approach inside testing environment for cucumber and then to analyses bugs through jerkin's & maven dependency with the use of various performance matrices Like speed control, rollup , Remaining pass-failed and Overall Throughput.



Test	# Passed	# Skipped	# Failed	Time (ms)	Included Groups	Excluded Groups
Suite						
Test	1	0	1	130,450		

Class	Method	Start	Time (ms)
Suite			
Test – failed			
com.w2a.cucumber.zoho.runners.ValidateCRMFeatureRunner	runCukes	1530182408227	130415
Test – passed			
com.w2a.cucumber.zoho.runners.LoginFeatureRunner	runCukes	1530182408227	49651

Figure 4. Testing Suit Scenario



Status	Timestamp	Details
ⓘ	4:04:26 PM	Scenario started : - Navigate to Zoho.com
✓	4:04:26 PM	SCENARIO PASSED

Figure 5. Running Scenario

VI. CONCLUSION

The proposed work is a security investigation and solution development of Test Cases in selenium Web driver & ATF. In this paper we have implemented an idea of using the new agile based automation suit testing report to run the web based paging applications based on selenium web driver which will surely enhance the readability to read the test suit cases & increase the pass- fail ration of test suit cases. It will further reduce overloaded workload of tester & lead. With the help this IDE- framework, we can create the customized doc-report and analyze the fail ratio using screenshots graphs and suit case figures. Tester can record the all information from one cloud. They are also helpful for automatically changing web paging applications. The agile based automation test cases suits are very understandable & easy to maintain using this IDE framework.

An Agile testing method is to give us to testing the software module web based solution. The main idea of an agile based automation testing is to cross down the price, reduce the efforts needed to run web based paging applications as well as web sites, grows software quality, minimize time-to-market and uses recoding test cases. Many agile based web automation tools like HP-QTP to Selenium were already exist. Our project article point on giving a suggestion on all the web test agile automation tools and so that it behave as better tools for future researchers in Selenium Enhancement

REFERENCES

[1] Monika Sharma, Rigzin Angmo, “Web Based Automation Testing and Tools”, IJCSIT Volume:5(1) , 908-912 , 2014

[2] Deepthi Wilson. R, Manjuprasad. B, “A Compressive Review on selenium Automation Testing Tools”, Department of Computer Science & Engineering, GSSSIETW, Mysuru,IJERT, ISSN: 2278-0181 2017.

[3] Satish Gojarea,*,Rahul Joshib,Dhanashree Gaigawarec, “Analysis and design of selenium web driver automation testing framework”, 2nd International Symposium on Big Data and Cloud Computing (ISBCC’15): 1877-0509 © 2015

[4] Prachi kunte , Prof Dhasrath mane, “Automation testing of web based application with selenium and HP UFT”, International Journal of Engineering & Technology, Volume 4, Issue 06, june 2018,pp-2395-0072

[5] Inderjit singh , Bindia Tarika, “Comparative analysis of open source software testing tools, selenium, sikuli and watir.” International Joural of information & Computation Technology”, ISSN 0974-2239 Volume 4, Number 15 (2014), pp. 1507-1518

[6] JyotiDevi, Kirti Bhatia,Rohini shrama, “A Study on Functioning of Selenium Automation Testing Structure”, International Journal of computer science and software engineering.and Knowledge Management, may 2017, Volume 7, issue-5 No. 2, pp. 855-862, ISSN: 2277 128X

[7] Sabastain Raju J, Vaidhehi V, “Design & Implementation of Hybrid test automation framework of web based application”, International Journal of Innovative Research in Advanced Engineering (IJIRAE) ISSN: 2349-2163;Issue 03, Volume 4 (March 2017)

[8] Srashti Lariya1 Dr. Sameer Shrivastava2 Er. Sumit Nema3, “Automation Testing using Selenium Web Driver & Behavior Driven Development (BDD)”, IJSRD - International Journal for Scientific Research & Development| Vol. 6, Issue 03, 2018 | ISSN (online): 2321-0613

[9] R. Anand* , M. Arulprakash, “Business driven automation testing framework”, International Journal of Engineering & Technology, 7 (2.8) (2018) 345-349

[10] Pratibha Singh, Puja Patel,, “Impact of agile software testing over traditional approach”,Internation journal of converging technology & management, VOL , NO 3, 2015