

**AUTOMATED TESTING FRAMEWORK FOR RAPID WEB APPLICATION  
TESTING**Pooja Gadave<sup>1</sup>, Vrushali Shendge<sup>2</sup>, Shital Tule<sup>3</sup>, Nilam Tule<sup>4</sup><sup>1,2,3,4</sup>Computer, BSIOTR

**Abstract** — Automation Testing is an emerging field that draws maximum advantage with minimum effort. Web testing is the name given to software automation testing that focuses on web applications. Complete testing of a web-based system before going live can help address issues before the system is revealed to the public. Issues such as the security of the web application, the basic ability of the site, its accessibility to handicapped users and fully able users. Automated testing there are multiple test cases Automated testing framework for rapid application testing project will focus on creating a framework which will manage parallels or multiple execution of test cases for multiple computers as well as reporting results of executing test cases. Automated testing can more effectively. We can test much more complex scenarios and will be able to find issues in much deeper level. Automation testing should find more bugs. Automation testing should reduce testing staff and also reduce time. A test case in software engineering is a set of conditions or steps under which a tester will set whether an application software system or a one of its features is working as it was originally established for it to do. Main intention to implement an Automated Testing framework is reduced manual efforts required testing entire application. This will in the end make software testing task cost effective and without effort.

**Keywords-** WAT, GUI WebObjects, jsoup, Selenium, Cross-Platform, time-efficient, cost-reductive

**I. INTRODUCTION**

To automate the more expensive part of software development is our long term goal of project. A program having set of input parameters will generate a set of output automatically after execution of program. It is an old idea that automating test generation using program analysis. The advance and recent progress is dynamic test generation which is more generalize and more powerful than older static test generation. If we automate the testing process then it will significantly reduce cost of software development. The Web GUI applications are becoming more and more complex due to development of internet technology. Recently software development cycle are becoming shorter, which makes GUI application testing, especially regression testing more challenging. The manual approach fails to address these issues hence automation testing is the powerful solution.

The challenges faced during GUI Automation Testing:-

- 1) The most challenging requirement is to handle dynamically changing GUI application.
- 2) With the increasing versions of application, regression testing is key requirement [1].
- 3) Cross-platform implementation is also an important requirement, if the actual framework implementation is in Linux but want to run test case by launching browser in Windows.
- 4) There is need for multiple browser support, because some bugs are browser specific.

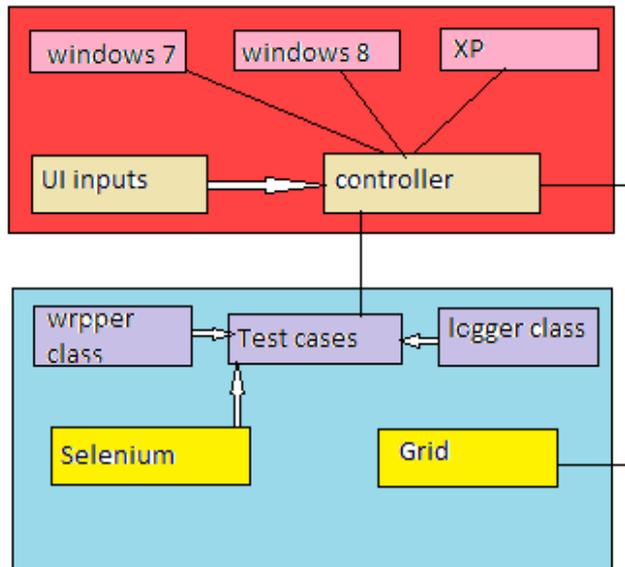
**II. LITERATURE REVIEW**

The number of tests carried out for verifying the correctness and accuracy of software application with respect to its performance. Testing is necessary for every software application before launching it in market. For this task, the QA means Quality Assurance Engineers perform this testing through manual testing. QA Engineer writes test cases by their own and test these test cases. After testing completed, they write result manually. Testing will be done at validation phase and it is costly and time consuming. Also it requires lots of programming knowledge and efforts so this disadvantageous task became easier by automating testing process. The dynamically changing nature of GUI application leads to automation becomes one challenge for testers. This automation was done by using automated tools like Selenium. There are many tools available, but Selenium is open source and works on web applications, so it was used for developing a framework [7]. The existing system is automation testing of web based application is one framework, which reads the HTML source program and generates GUI web objects. This framework contains a library, API and test cases suite for performing automation testing on any web base application. In the existing system Selenium was used to develop that framework [6] [8]. The existing system automated testing of web based application has some disadvantages like this can be run only on one server at a time and also it perform testing on only one platform or operating system at the same time. Because of this disadvantage testers does not get faster performance and also they have to write result manually. In recent years, the automation tool Fit and Fitness were used to make easy for automating functional testing of software applications. But there are some difficulties remain as it is because of to test multitier architecture, multiple browsers, and web technologies such as JavaScript. For solution on this problem some teams uses expensive tools like WinRunner or Rational Robot.

### III. SYSTEM DESIGN

In this automated testing, multiple test cases can be run on multiple operating systems or platform at the same time multiple. User interface communicates with the controller in automation. The controller is the basic main point of the system. Automation testing framework is for testing web based application, this will focus on creating a framework which will manage parallel execution of test cases on multiple computers as well as reporting of results of executed test cases .Main intention to implement this concept is to reduce manual efforts required for testing entire application .This will ultimately make software testing task cost effective and effortless .

#### ARCHITECTURE DIAGRAM:



**Figure 1:** Architecture Overview

#### ALGORITHM:

There are three modules and detailed implementation of the module is given below.

- A. FTIA (Framework and tool Integration algorithm)
- B. WOC (Web object creation Algorithm)
- C. TCA (Test Case Abstraction Algorithm)

#### A.FTIA

FTIA is integrates WAT framework and the external tool Selenium and also brings up the web application for testing , providing interface to interact with web application throughout testing and terminating it once testing is done. Presently it provides support for Windows and Linux operating system. There is also an option to specify the type of browser to be launched and the application specific URL. The algorithm is the one which actually interacts with the application and other modules uses it as interface to interact with the application. It also extracts the html source of the web application for further processing.

**INPUT :** webURL, web Browser ,operatingSystemName

**OUTPUT:** LaunchBrowser, //take commands/events from EGA and pass it to external tool selenium

#### ALGORITHM :

1. OperatingSystemName=System.getproprty(“os.name”)
2. getBrowserPath(filename)
  - a.parse configuration file to extract browser path
- 3.getMachineName(filename)
  - a.parse configuration file and extract machineName
4. launchBro wser(web BrowserPath,webURL,portNu mber)
  - a.webBrowserPath=getBrowserPath(configFile)
  - b.webURL=give URL for web application

```
c.portNumber=getRandomPortNumber()
5.executeCommand(String sCommand)
a.machineName=getMachineName(config File)
6. startSeleniumServer(portNumber)
a.executeCommand(“java -jar selenium-serever-standalone-2.1.0.jar -port “+)
7.startSeleniumClient()
if(operatingSystemName==’Windows’)
a.CreatesocketconnectionthroughW
b.startSeleniumServer(getRandomPortNumber ())
c.launchBrowser(getRandomPortNumber,get machineName(),
getBrowserPath, webURL)
d.Events from EGA passes through WA
else
a.startSeleniumServer(getRandomPortNumber ())
b.launchBrowser(getRando mPort Nu mber,get machineName(),
getBrowserPath, webURL) .
```

### **B. WOC(Web Object Creation)**

In Web applications different web objects are categorized based on the html tags associated with it in html source code. WOC algorithm takes html tag as an input and accordingly creates list of web objects. These web objects are then used by WSA for generating the locators.

**INPUT :**html tags

**OUTPUT :**list of web objects

#### **ALGORITHM :**

```
1.getHtmlTag()
a.returnobj.tagName()
2.getIdentifier()
a.returnobj.identifier()
3.if(htmlTag==obj.tag)
a. create list of list<obj>
```

### **C. TCA(Test case Abstraction Algorithm)**

TCA is provides abstraction for test case specific implementation. The Algorithm changes as per the language in which test case changes. For example we will discuss JUnit. JUnit supports annotations like @after and @before. With these annotations we can specify the steps to be executed before test case execution starts and steps to be executed after test case execution ends [9].

**INPUT :**type of language in which test case is required

**OUTPUT:** setup for writing test cases

#### **ALGORITHM :**

```
1.Initialize()
2.AppSpecificSetup()
3.EnvironmentalSetup()
4.tearDown()
5.Cleanup()
```

### **MATHEMATICAL MODEL:**

Finite State Machine:

I= {<browsers>,<os>,<TestCases>}

S= {<Accepting Inputs>,<Connecting Machines>,<Starting Browsers>,<Running tests>,<Producing Outputs>,<Sending Results>}

T= 1) GetInputs: Receives list of bowsers, os and Tests cases to run and changes state to ConnectMachines

2) ConnectMachines: Controller Connects all the Machines and starts Grid (Hub) and selenium nodes on all those machines and changes state to Start Browsers

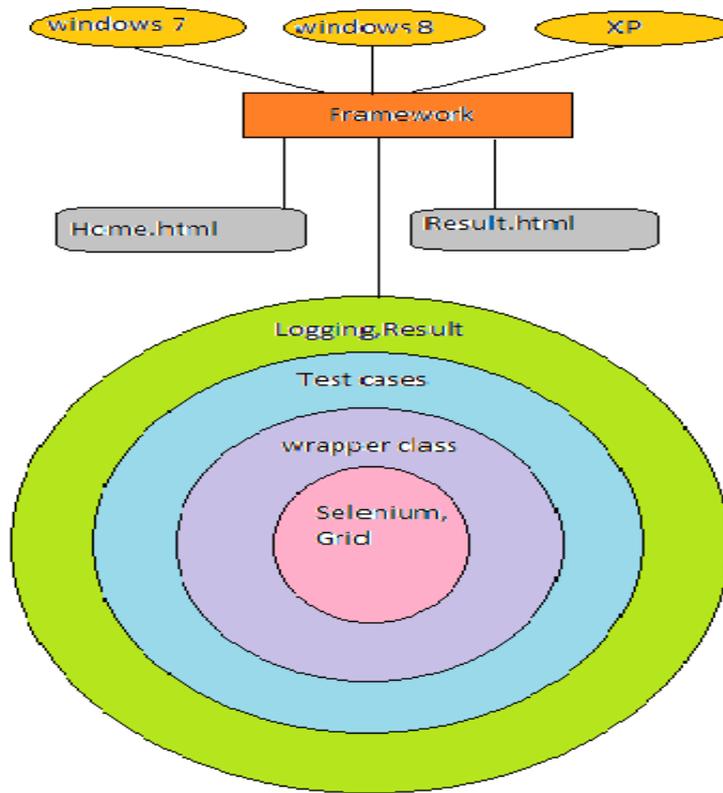
3) StartBrowsers: Controller starts Browsers on all the machines and changes mode to TestExecution

4) TestCaseExecution: All test cases will be distributed to browsers and browsers will run the tests

5) ResultGeneration: After execution of each test case result will be recorded and after completion of all the tests results summary is sent to respective.

6) StepFailed: At any stage if system fails then it will come in StepFailed step and reason for failure will be displayed.  
F= [Send Results]  
L=[AcceptInputs]

#### IV. PROJECT IMPLEMENTATION

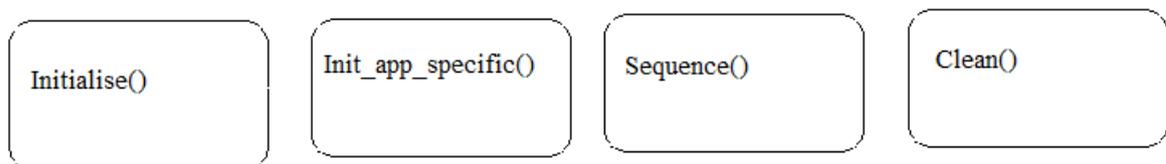


**Figure 2:** Frame work of selenium

In this project we are developing one selenium framework for automating testing of web applications. This selenium framework is common platform for automation. We have used selenium because of advantages over other automation tools. It is open source. The term selenium Grid is used as hub and selenium RC is used as node for connection between two or more computers. Selenium RC has two components server and client. Server is responsible for launching and killing the browser and also acts as HTTP proxy intercepting and verifying HTTP messages passed between and application under testing[10,11]. Client libraries provides interface between the programming language and Selenium RC server[2]. The Server receives commands from our test program using simple HTTP GET/POST requests. It means we can use any programming language that can send HTTP requests to automate Selenium tests on the browser. A Selenium client library provides a programming interface (API), i.e., a set of functions, which run Selenium commands from our own program[2]. The client library takes a Selenium command and passes it to the Selenium Server. The client library also receives the result of that command and passes it back to program. The program can receive the result and store it into a program variable and report it as a success or failure, or possibly take corrective action if it was an unexpected error.

#### A. Test Case Executer

Test Case Runner gives the test case execution sequence. As xpath[6] is created on runtime the dynamically occurring changes due to change in gui application development can also be handled.



Initialize()- Performs Initialization operation and initializes required components for testing. Init\_app\_specific()- Prerequisites for specific application are executed for example launching of browser. sequence() – Steps for executing specific test scenario are executed. clean() – Cleaning up after test execution so that the next test case will run correctly

#### B. WAT Execution Engine:

WAT framework is an object-oriented framework developed in java[4] and it also includes different action-events[3] need to perform on the objects which is necessary for performing functional testing[5].

WAT Framework contains the following modules:-

##### WebObjects :-

For GUI Application we are creating web objects of the web elements like button, dropdown, graph ,page etc so that test cases can use the object for generating the test scenario and performing the necessary operation.

##### JSoupParser:-

Each web objects has different html tags on the basis of the tag and the identifier , Jsoup parser helps us to parse the html source code and create the xpath during runtime.

##### External Interfaces and requirements:

##### 1. Java Development Kit:

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows. The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application.

##### 2. Eclipse:

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications.

##### 3. Selenium:

Selenium automates browsers. What you do with that power is entirely up to you.

This tool is used for automating web application testing and also automates boring web-based administration tasks.

##### 4. Selenium Grid:

Selenium-Grid allows you run your tests on different machines against different browsers in parallel. That is, running multiple tests at the same time against different machines running different browsers and operating systems. Essentially, Selenium-Grid support distributed test execution. It allows for running your tests in a *distributed test execution* environment.

#### Non Functional Requirements

In systems engineering and requirements engineering, a nonfunctional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours [9]. Non functional requirements can be divided into two main

Categories 1. Execution qualities, such as security and usability and Evolution qualities, such as testability, maintainability etc.

Core Requirements:

- Browse urls
- Render web pages
- Mimic User Actions
- Validate results
- Ajax/CSS support

Pragmatic requirements:

- Language choices
- Platforms (Browser/OS)
- JS toolkit support
- Distributed tests
- Scheduled execution
- Reporting
- Configurability
- Build Integration
- Test Creation Environment

### V. RESULT AND DISCUSSION

We performed few experiments which give following results shown in the Table-1 average execution time of automated test case exceeds 20 sec from average execution time of manual test case. Even though automation takes some extra time, but unlike manual validation it runs in background, so that tester intervention is not required throughout execution. It is important to consider the time to execute the checklist as shown in Table-1, because as and when new version of the application is released the tester has to execute and validate all the previous checklist, Automation provides the way to run the regression and saves lots of efforts and time of a tester.

TABLE 1:

Execution time to run checklist Time required	Manual Execution	Automated Execution
Create checklist	32 hrs	32hrs
Execute One checklist	67min	31min 30 sec
N checklist	N*67min	N*31 min 30 sec

## **VI CONCLUSION**

Automated testing is easy to switching in various types of testing for web applications. It supports multiple web browsers application and a variety of operating system. It can be commonly used in web application test automation. Parallel execution of test cases on multiple computers and reporting result of executed test cases. Automation Testing is more efficient. Automated software Testing Saves Times Money Be Cautious before choosing automation tool. Automation Tools are reliable in complex calculations & tasks. Automation Testing improves accuracy.

## **REFERENCES**

- [1] BOGDAN KOREL, MEMBER, Automation test generation IEEE TRANSACTIONS ON SOFTWARE ENGINEERING. VOL.16. NO. X. AUGUST 1990.
- [2] Mrs. Y.C. Kulkarni Mr. Ajeet A. Ghodeswar "AUTOMATING THE WEB APPLICATIONS USING THE SELENIUM RC".
- [3] Automating Software Testing Using Program Analysis, Patrice Godefroid, Peli de Halleux, Aditya V. Nori, Sriram K. Rajamani, Wolfram Schulte, and Nikolai Tillmann, Microsoft Research Michael Y. Levin, Microsoft Center for Software Excellence September/October 2008.
- [4] Antawan Holmes and Marc Kellogg Digital Focus "Automating Functional Tests Using Selenium" Proceedings of AGILE 2006 Conference.
- [5] Improving web application testing using testability measures nadia alshahwan<sup>1</sup>, mark harman<sup>1</sup>, alessandro marchetto<sup>2</sup> and paolo tonella<sup>2</sup>.
- [6] Web-Based Automation Testing Framework International Journal of Computer Applications (0975 – 8887) Volume 45– No.16, May 2012.
- [7] fNadia .Alshahwan, Mark.Harmang@kcl.ac.uk, fMarchetto, Tonellag@fbk.eu.
- [8] INCON13-IT-047
- [9] *International Journal of Computer Applications (0975 – 8887) Volume 45– No.16, May 2012 4*