Role of Automated Testing Tools in Software Testing

V. Neethidevan
Department of Computer Applications
Mepco Schlenk Engineering College (Autonomous), Sivakasi

Abstract

In order to test software effectively use of automated tools is the best choice. Manual testing is a time consuming one and boring task. Earlier testing is done by manually with the input data. Now more complex applications developed and it needs a thorough testing. Testing phase of a software system consumes more time. So testing the software systems with automated tools means, more time can be saved and it improves the confidence in the software systems. This paper discussed about how to perform automated testing using Jmeter and how to do nonfunctional testing like performance testing, load testing and stress testing. It is a open source tool and it is used to perform various form of web based testing. The performance testing is ensure performance of our web site, when more number of concurrent users are accessing your system. With the performance testing, system throughput can be easily measured. [1]

Keywords- Software Testing, Automation Tools, Automated Testing, Manual Testing

I. INTRODUCTION

A. What is Automation Testing?

Each and every module of a software system is made error free by performing unit testing by the module owner. During developing the module, he has to prepare the test cases and then performing manual testing by executing the program and proving various inputs manually. It is a time consuming process and it takes more time to perform. To avoid this one, automation testing is proposed. It is nothing but using any one automation tool to perform the testing Manual Testing is performed by a human repeatedly executing test cases. It is a time consuming process. So there is a need for automated testing. For example, xUnit architecture is used to perform unit testing. NUnit tool is used to perform unit testing on programs developed using .Net platform. JUnit is a unit testing framework to perform unit testing and it is the base for all other levels of testing like integration testing and system testing. The author [4] described about NUnit framework that is for unit testing for windows applications.

Using unit testing framework as Junit, automated testing is possible by comparing actual value with expected value. Using various assert statements various test cases can be checked for successful or failure one.

In recent years many more software applications were developed for various purposes and the major concern is software reliability and security. The paid version of static program-analysis tools are now regularly used in many software development organizations. [2] These tools are more popular because they find many software bugs, and the three main ingredients: they’re automatic, they’re scalable, and they check many properties. Naturally, using any of one of tool we can automatically check millions of lines of code against the various coding conventions followed in software industry. Once the software is tested, reliability increases and the increases the confidence.

In the following sections discussed about Literature review, Evaluation of study and conclusion and future work.

II. LITERATURE REVIEW

The fundamental test unit is the unit testing to ensure there is no errors in the module so that quality and reliability of software is improved.[3] This unit testing is performed by manual process or automated process in which testing tools are used to perform unit testing. Junit is one of the unit testing framework to do unit testing and it is the base for all other levels of testing like integration testing and system testing. The author [4] described about NUnit framework that is for unit testing for windows applications.

In [5], authors discussed about performance testing concepts, objectives, goals, types and tools that support performance testing. Also this type of testing is used to determine the response time, throughput, reliability, and scalability of a software application under a given workload.

In [6], the authors talked about automating the regression testing process in which modified program is tested one more time to gain more confidence. They proposed a test generation method to automatically generate test data to test modified program.
III. Evaluation of Study

A. JUnit - Testing Framework for Java

It is an open source testing frameworks for Java programs. The application developed using Java can be tested for Unit testing. The current version is Junit4.[7] Two classes are needed to perform unit testing. One is tested class and another one is test case class.

The org.junit package contains many interfaces and classes for junit testing such as Assert, Test, Before, After etc.

Annotations for JUnit testing

The Junit 4.x framework is annotation based, so let's see the annotations that can be used while writing the test cases.

@Test annotation specifies that method is the test method.

@Test (timeout=1000) annotation states that method will be failed if it takes longer than 1000 milliseconds (1 second).

@Before Class annotation states that method will be invoked only once, before starting all the tests.

@Before annotation specifies that method will be invoked before each test.

@After annotation specifies that method will be invoked after each test.

@After Class annotation specifies that method will be invoked only once, after concluding all the tests.[8]

Example,

```java
package Myunit;
import org.junit.Test;
import static org.junit.Assert.*;
public class MyUniTest {
    @Test
    public void testConcatenate() {
        Myunit myUnit = new Myunit();
        String result = myUnit.concatenate("one", "two");
        assertEquals("onetwo", result);
    }
}
```

To test the above code

```java
package Myunit;
public class Myunit {
    public String concatenate(String one, String two) {
        return one + two;
    }
}
```

B. Test Suites

In JUnit, test suite allows the testers to cumulative all test cases from multiple classes in one place and run it together.

To run the suite test, the following annotations are used

1) @RunWith(Suite.class)
2) @SuiteClasses(test1.class,test2.class……) or
@Suite.SuiteClasses ({test1.class, test2.class……})

With above annotations, all the test classes in the suite will start executing one by one.
import org.junit.runner.RunWith;
import org.junit.runners.Suite;
@RunWith(Suite.class)
@Suite.SuiteClasses({
    TestJunit1.class,
    TestJunit2.class
})
public class TestSuite {

C. Jmeter
Jmeter is a popular open source performance testing tool. The Apache JMeter is an open source based on Java, developed by Stefano Mazzocchi of the Apache Software Foundation, used to load test functional behavior and measure performance. With this, to analyze and measure the performance of web application or variety of services. Performance Testing means testing a web application against heavy load, multiple and concurrent user traffic. JMeter originally is used for testing Web Application or FTP application. The following screen shots explain about performance testing done on www.google.com/calender application, when simulated by 100 users concurrently and returns statistics information through graphical diagrams, table forms and tree based output.

The above screen shows various steps involved in creating test plan.

D. Summary Report
In this we can specify various elements like, Name, read result from file, log and display only , save table data.
E. Tabular Results

F. Graph Results
Graphical results, used to analyze the performance of the web server under test, and focused on 2 parameters
- Throughput
- Deviation
The Throughput ability of the server to handle heavy load. The higher the Throughput is, the better is the server performance.
In this test, the throughput of Google server is 1,491.193/minute. It means Google server can handle 1,491.193 requests per minute. This value is quiet high so we can conclude that Google server has good performance.[9]
The deviation is shown in red - it indicates the deviation from the average. The smaller the better.
IV. CONCLUSION AND FUTURE WORK

This paper described about how to do automated testing using various software testing tools such as Junit testing frameworks and JMeter used to perform performance testing. Using Junit testing framework we can able to test unit testing developed using Java and also multiple test cases can be in a single batch using test suites [10]. Using JMeter, various nonfunctional testing like performance testing, load testing and stress testing can be performed. The other measurements such as throughput of our application and deviation can also be noted. Thus using automated tools more time and more resources can be saved thus automated tools play a vital role in software testing. The tester must have the proficient knowledge in various testing tools available and he has to choose optimized tool for the current software system under test. The future work may be performed for Mobile testing framework which involves most of the applications developed today are for mobile devices only.

REFERENCES