



An Automated Testing Model Using Test Driven Development Approach

C.P. PATIDAR¹ and ARUN DEV DONGRE²

¹Assistant Professor, IT Department IET-DAVV, Indore (M.P.) India

²PG Student, Computer Engineering IET-DAVV, Indore (M.P.) India

Corresponding author e-mail: cpatidar@ietdavv.edu.in

<http://dx.doi.org/10.13005/ojcs/10.02.18>

Received: April 06, 2017; Accepted: April 14, 2017)

ABSTRACT

Today we live in the era of software and web applications. Software is used in every minor and major field. In defense, medical, education, research, government, administration and much other field software became a necessary part. Software also brings transparency in the systems. Software also makes people's life easy and comfortable. Software testing is a very important part of any software development process. Software testing requires approximately forty percent budget of any software development process. Like in an automobile industry every vehicle is tested before it goes to the customer. Also in software testing it is must to test the software before deployment. Because if software deployed without testing then user will face the bug and user will be unhappy with the software. In this paper we compare manual and automated testing and proposed an automated testing model with test driven development (TDD).

Keywords – Software testing, Test driven development (TDD) framework, test plan, test case, Application under test (AUT).

INTRODUCTION

Software testing is a method to find out the errors and defect of application under test and give a suitable solution for that. Now a day's software testing plays an important role in software development process. The main purpose of software testing is to ensure that the developed software or application has a good quality. Two

methods available for testing is manual testing and automated testing.

Manual testing is an old process. In this type of testing tester takes a place of stakeholder and ensure that the application behaves properly. For that tester creates a test plan, which has all the possible test cases and execute them in a manually manner and comparing results to the expected behavior.

Automated testing are required because of the development of web application. In this type of testing tester creates a test plan, test cases and execute them with the other test suites called framework. These test cases are automatically executed by the test suite and it gives the output¹.

Testing

Software testing is a method of assessing the functionality of a software or application. Testing ensure that the software is reliable, efficient and high in quality. There are many types of testing.

On the basis of test execution there are two types of testing

1. Manual Testing/ Traditional testing
2. Automated Testing

Manual Testing

Manual testing includes testing the software manually, without using any automated tool or any script. In this type of testing, tester takes over the role of an end-user and tests the software to identify any unexpected behavior or bug. It is an old and rigorous method of software testing. In manual testing, tester needs to possess a certain set of qualities like, to be observant, speculative, creative, innovative, and open-minded. So it is a laborious activity to do. Tester use test plans, test cases, or test scenarios to test the software to ensure the completeness of testing. Manual testing is an exploratory testing as tester explores the software to identify the errors or bugs in it. Explore a small software is easy but it is very hard with the large software or applications having large amount of data set coverage.

There are some problems with manual testing Time Consuming

Test cases are executed by the tester (human) which takes much time to execute the test.

Less Accuracy

Test cases are executed by tester so it is not performed accurately every time because of human errors.

Limited Scope

Test case scope is limited in manual

testing for example GUI object difference and color combination are not easy to find in it.

Can't Perform Non-Functional Testing

Non-functional testing like load testing and performance testing is not possible in manual testing.

Low Scalability

Manual testing is not suitable for large scale and time bonded projects.

Automated Testing

Automated testing is a part of both testing and development team which requires programming concepts as well as testing strategies. In automated testing tester creates test cases which is written in different scripting languages and executed by the computer. There is no need of any extra human efforts for execution of test case. Automated testing used specialized testing tool to control the execution of test case and compare the actual results with expected results. Testing tool not performed regression tests but help to automate data set up generation, product installation, defect logging etc. These testing tools are expensive so automated testing demands money but in the regression tests situation it is less expensive. Automation testing also decreases the human efforts and resources. Automated testing is much faster during the execution of test case. In the automated testing it is possible to retest again and again in less time as well as record the results of each time. Automated testing can perform functional as well as non-functional testing, like performance testing, security testing etc.

There are some benefits of automated testing Time Efficient

Test cases run by computers or other programs so it is faster in terms of execution.

Fewer Human Resources

Automated testing decreases human resource because there is no need of human effort during execution.

Reusability

Test scripts are reusable in any system environment and also allow the redo method without

forgetting any step.

More Reliable

Automated testing is performed by tool so there are no errors arising by human.

Wider Test Coverage

Automated testing has wider test coverage of application features. It can perform functional and non-functional testing.

Highly Scalable

In automated testing a large scale project can test in a certain time.

Cost Efficient

It reduces human efforts and resources. For a large scale project automated testing is very efficient.

Test Driven Development (TDD)

Test Driven Development (TDD) is a key practice of extreme programming (XP)³. TDD is a

practice which prescribes that the test code should be written before the application code. TDD is very strict about how the development should be done. It says that all the coding should be done in following phases:

1. Write test method and execute it, it should fail or even not compile as there is no application yet.
2. Write the simplest possible implementation for test to pass.
3. Refactoring is a disciplined way to restructure the code. Refactoring is use to remove all duplication and make application code clear.

Literature Review

Current research in the field of testing is focusing on making the testing reliable, cost efficient, fast and more intelligent. To achieve this purpose the automated testing getting high interest. There are many research papers and books are available to illustrate the pros and cons of manual

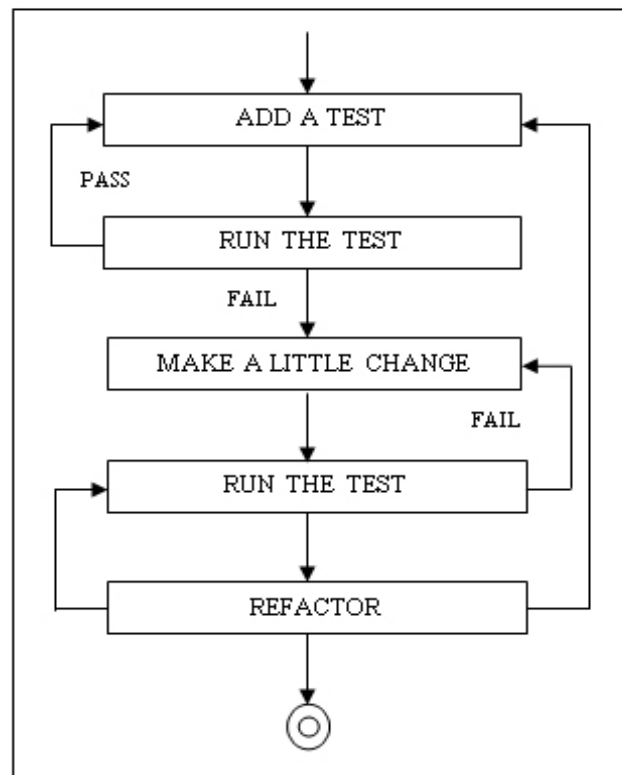


Fig.1: Flowchart of Tdd Process⁴

testing and automated testing. The various types of automated testing had been discussed in several research work:

Karhu et al.⁵ has discussed in his paper that "The cost of software testing is decreases if the automated testing is used. Automated testing takes less time and increases the quality of the software product. This paper also explains that automated testing needs extra efforts (Implementation, Execution, Maintenance, and Training). Automated testing system suffers from the same issues as any other systems.

Auri M. R. Vincenzi et al.⁶ published an article. In this article, manually generated test sets compare with automatically generated test sets on the basis of adequacy, effectiveness, and cost. This article also explained that the automatically generated test sets have higher statement coverage and mutation score as manual test sets.

Gerardo Canfora et al.⁶ also write on automated testing in his paper. In this paper the advantage of Test driven development (TDD) is described. The Testing after coding (TAC) is also illustrated. Differences between the Test driven development and Testing after coding and how they affect the software quality and productivity is explained in this paper.

Fei Wang and Wencai Du et al.⁸ published a paper. In this paper, authors have designed an automatic software testing framework for web applications based on the Selenium and JMeter. With the use of the software framework, we efficiently improve the extensibility and reusability of automated test. The results shows that the new software framework improves software product quality and also improve efficiency. This paper also illustrates how to design web-based test automation framework in details.

Tim A. Majchrzak and Andreas Simon⁸ explained Model-driven software development (MDSD) and test-driven development (TDD) in his paper. They also present a novel approach combining MDSD and TDD while using the Spring Roo Web development framework.

Tools Used in Automated Testing

These tools can be roughly categorized into few distinct groups⁹

Automated Web Testing Tool

These tools ensure that application don't gives errors. These tools are used to test the functionality of the application. User interface of an application might change regularly, because of the incompatibilities between browsers and server or client platforms. Some tools are given below which helps to make it easier to build and execute automated tests for web applications.

Selenium

SELENIUM is an open source and combination of different testing tools, each of them has different approaches. SELENIUM has capable to do the testing of web application of all types and helps to automate web browsers, across different platforms¹⁰.

Sahi

It is an open source tool for automation of web application. A commercial PRO edition of sahi is also available. Sahi allows that same script works on all browsers¹². Sahi PRO is used for cross browser testing of web application.

Watir

It uses Ruby libraries for automating web browser¹³. It allows writing tests that are easy to read and maintain. It is simple and flexible.

Automated GUI Testing Tools

Automated GUI testing tools are used to create and performed high quality GUI tests for different platforms and operating system⁹. Some GUI testing tools are listed below:

Test Complete

It supports various languages (C++, Python, Java) to create test. Also used to create robust GUI test for wide range of applications. It creates high quality test in affordable cost¹⁴.

Squish

Squish allows creating automated tests using different scripting languages such as Python, Java, and Ruby etc. It supports various platforms¹⁵.

Unit Testing Frameworks

Unit testing tools are used to test the specific functionality of application. Unit testing tools are dependent to programming language used to create test cases. Some of these are listed below:

Junit

It is an example of the xUnit architecture for unit testing. JUnit supports test driven development. It is simple framework for repeatable test written in java language¹⁶.

Nunit

It was initially ported from JUnit to NUnit. It supports all .Net languages and wide range of .Net platforms¹⁷.

PHPUnit

PHPUnit testing tools supports the project written in PHP language. It helps to write tests, execute them, and analyze results¹⁸.

Performance Testing Tools

Performance testing tools are used to create, execute and analyze the stress and load tests of web application. For load testing tools need to create all possible test cases as a user would performed in application then simulate many virtual users simultaneously. Some performance testing tools are listed below:

ApacheJmeter

It is an open source java based performance testing tool. It is able to test the performance of static and dynamic application. It can also analyze the performance in different load types by simulate a heavy load on server or group of servers¹⁹.

LoudUI

LoadUI is an API load testing solution. It supports all the standard protocols (REST, SOAP, AMF, JDBC, HTTP(S), and HTML). It gives complete test coverage of all types of performance testing²⁰.

CONCLUSION

Manual testing is appropriate for the small scale software project. For a large project, manual testing demands a huge investment of human resource and enough time to complete the project. Automated testing is a fast and reliable process. Manual testing can't perform non-functional testing and regression testing. Automated testing is able to execute all testing in a certain time with high accuracy. Automated testing is quite expensive because automated testing tools are expensive, but it is cost efficient for large and regressive project. Automated testing is better than manual testing in different parameters like reliability, time efficiency, accuracy and scalability.

REFERENCE

1. Yogesh Singh "Software Testing", Cambridge University Press, ISBN 978-1-107-01269-7, First edition 2012.
2. Vivek Kumar "Comparison of Manual and Automation Testing", International Journal of Research in Science and Technology (IJRST) 2012, Vol. No. 1, Issue No. V, Apr-Jun ISSN: 2249-0604.
3. Kent Beck, "Extreme Programming Explained ; Embrace change", Addison-Wesley, second (2nd) edition, 1999.
4. Flowchart of TDD process, <http://agiledata.org/essays/tdd.html>, 20 August 2016.
5. Karhu, Katja, et al. "Empirical observations on software testing automation", Software Testing Verification and Validation, 2009. ICST'09. International Conference on. IEEE, 2009.
6. Auri M. R. Vincenzi, Tiago Bachiga, Daniel G.de Oliveira, et al. Proceedings of the 7th International Workshop on Automating Test Case Design, Selection, and Evaluation. FSE (Foundation of Software Engineering) Conference, ACM New York, NY, USA, 2016
7. Gerardo Canfora, Aniello Cimitile, Felix Garcia et al. "Evaluating Advantages of TestDriven Development: a Controlled Experiment with

- Professionals", *International Symposium on Empirical Software Engineering (ISESE'06)*, ACM , September 21-22, 2006.
8. Fei Wang and Wencai Du et al. "A Test Automation Framework Based on WEB", *ACIS 11th International Conference on Computer and Information Science (ICIS)*, IEEE , 2012.
 9. Categories of automated testing tools, <http://www.testingtools.com/test-automation/>, 9 September 2016.
 10. SELENIUM FRAMWORK, http://www.seleniumhq.org/docs/01_introducing_selenium.jsp, 15 September 2016.
 11. Volume 3, No. 12, December 2012 Journal of Global Research in Computer Science RESEARCH PAPER Available Online at www.jgrcs.info © JGRCS 2010, All Rights Reserved 36 STUDY AND ANALYSIS OF AUTOMATION TESTING TECHNIQUES
 12. SAHI Tool, <http://sahipro.com/>, 25 September 2016.
 13. Watir Tool, <https://watir.com/>, 29 September 2016.
 14. TestComplete Tool, <https://smartbear.com/product/testcomplete/overview/>, 6 November 2016.
 15. Squish Tool, <https://www.froglogic.com/squish/>, 10 November 2016.
 16. JUnit Tool, <http://junit.org/junit4/>, 17 November 2016
 17. NUnit Tool, <http://www.nunit.org/>, 22 November 2016.
 18. PHPUnit Tool, <https://phpunit.de/>, 28 November 2016.
 19. Apache Jmeter Tool, <http://jmeter.apache.org/>, 7 December 2016.
 20. LoadUI Tool, <https://smartbear.com/product/ready-api/loadui/overview/>, 12 December 2016.