Graphical Testing of Software Development Model for Regression Testing

Supriya[1], Manish Mahajan[2]
[1] Research Scholar, Department of Information Technology, CEC Landran.,
[2] Associate Professor, Department of Information Technology, CEC Landran.,

ABSTRACT: Due to increased complexities in the software development, there is huge need of testing process to be carry on in better way. Also as the computer systems are significant to our society in every daily life and are performing an increasing number of critical tasks, so more work in software testing and analysis has become of great importance. Testing is having easy process when we process it through graphical user end. In this paper, we have provide view of an optimized approach which use automation of software testing process using graphical Interface tool. Automation of the test plays a significant role in testing activity, as it saves time and provides better utilization of resources. The test automation has many situations to deal such as mapping of user specifications to identification of the test cases, test case generation, maintenance of test cases and test scripts. Graphical user interface software testing is an important process to utilize the easy process for testing. A process of executing a program with the goal of finding errors is known as testing process. As the size and complexity of software is continually growing, manual testing becomes very difficult and tedious task. In this paper we have done experimentation based on the testing cases prioritization process with graphical user interface testing process.

Keywords: *Software Development Life Cycle, Software testing, Test case priority, Graphical Interface*

1. INTRODUCTION

Test automation is a process of writing a computer program to do testing that would otherwise need to be done manually. Once tests have been automated, they can be run quickly and repeatedly. This is often the most cost effective method for software products that have a long maintenance life, because even minor patches over the lifetime of the application can cause features to break which were working at an earlier point in time.

In testing a software application, a developer may wish to apply different tests to various code regions. The tests are also often applied with different cover- age criteria. Our testing tool includes a graphical user interface for specifying the tests to apply, where to apply them, and under what conditions. Our tool for branch coverage testing includes the capability to select code regions using the GUI interface. Coverage criteria can also be specified for each region. The structure of initial testing tool is given below in figure 1.

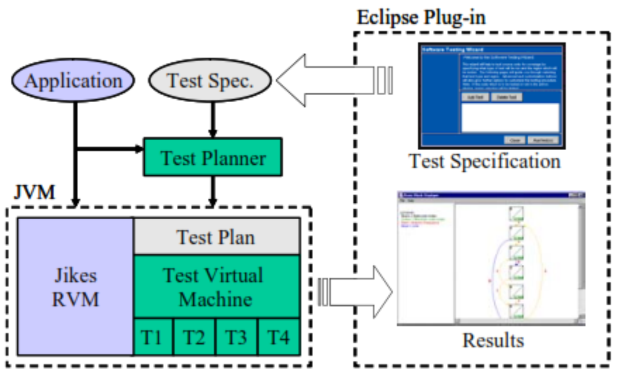


Figure 1: Basic architecture for graphical tool with testing framework of Java

2. AUTOMATED GRAPHICAL FRAMEWORK

We are developing Graphical testing based tool as a complete framework for testing of Java software. The components in the framework, including a test specified a test planner, a test virtual machine (TVM), and a test analyzer. One component is a language, test spec, for specifying a software test process. The specification includes the relevant parts of the program to be tested and the actions needed in the testing process. Testers can either write a specification in test spec or, better, use the GUI, which automatically generates a specification in test spec. A test planner consumes the test spec specification and determines a plan for testing the program given the specification.

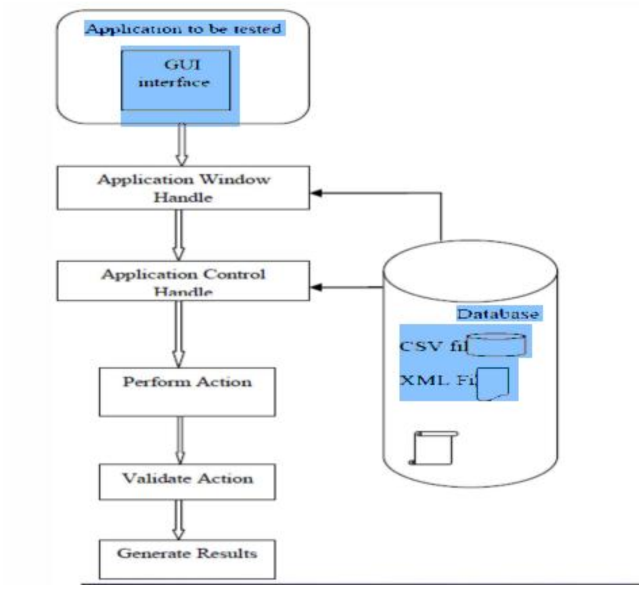


Figure 2: The Process Flow chart [3]

3. PROBLEM DESCRIPTION

Our Test cases design is store in the database that is the Excel with the number of the input and the actual value of the output. whichever the value is going to enter by the are going to match from that database the which test case is supposed to be passed will be shown in the graphical user interface. Except from the only the number of the selected input the user can also give the any number of the input it could range from 10 to 1000 we are providing the module of the automate testing in which the user need to upload the file with any number of the input and all the test cases that are pass and fail will be shown in the new interface. The Priority of the test case is also taken as the major module of our work for showing the priority to the test cases the user is provided with the Interface with the number of the test cases shown in the checkbox the user has to select only the number of the test cases that is user want to execute first once the user click on the desired checkbox the .The Particular Test case is executed and the execution of the test case is done as according the user click the checkbox.

4. EXPERIMENTATION DONE WITH DISCUSSION OF RESULTS

In this work we put more work on making a part of the code in which it is tested for the both testing the GUI Interface and testing of the source code also with providing the priority to the test cases that will be executed according to the choice of the user.

Attributes	Values
Language used	JAVA
Development Tools	Eclipse
Number of test cases	12
Classes used	Manual by user
Database	Mysql 5.0

Table 1: Attributes used for research

This research has focused on automated testing of the software development process by enhancing some term of the testing process. One of the most important parts of any software testing approach is on save resources while doing testing in any software. Software test cases prioritization is the process used for providing priority to the test cases which are built by dividing total code into various small parts. Selection of test cases is also a challenge in the testing and priority of testing for various test cases are also required.

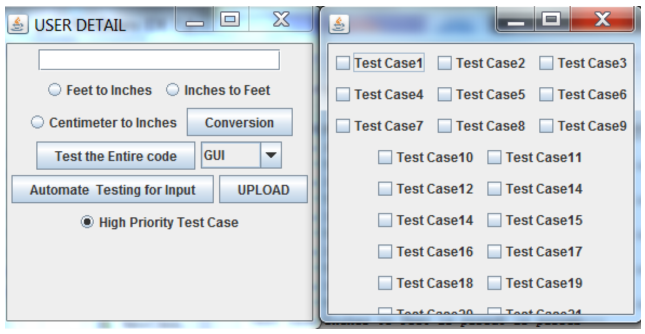
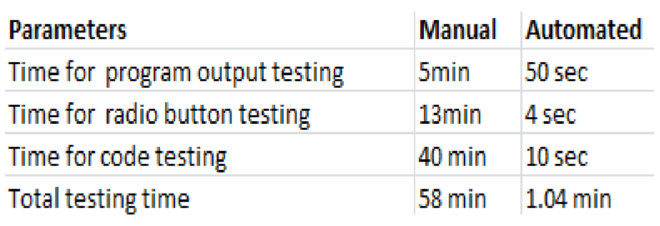


Figure 3: Proposed Module for Test Cases Priority

The figure 3 shows the most important module of the software testing that is providing the Priority to the user Test cases that is user is interested in executing. The Major Module of the Project includes the Priority of the Test cases in this the user can only test those cases which the user feeling is might be important as compare to the rest of the test cases.so that he can skip the rest of the test case. When the user is going to click on the below displayed radio button with the name high priority a new frame window is appeared this frame window consist of all the test cases to which we are going to provide the priority. The user just has to click on the Test which user wants to execute. After that click event has occur the result of the user Proffered test case are displayed in the frame.



Parameters	Manual	Automated
Time for program output testing	5min	50 sec
Time for radio button testing	13min	4 sec
Time for code testing	40 min	10 sec
Total testing time	58 min	1.04 min

Figure 4: Graphical User Interfaces for the complete software testing module

Figure 4 shows the comparison of the manual and automated testing in term of time taken. The Automation testing is the good solution for various application testing and provides better and faster testing results.

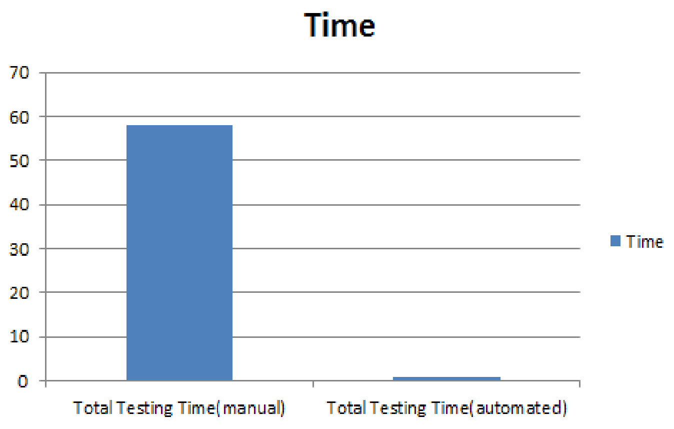


Figure 5: Comparison of Time Taken

The time is one of the essential parameter to consider while testing the software application. As suggested in figure 5, testing time taken by automated testing is 2 seconds with 12 test cases as compared to manual testing which process same test cases in 57 seconds.

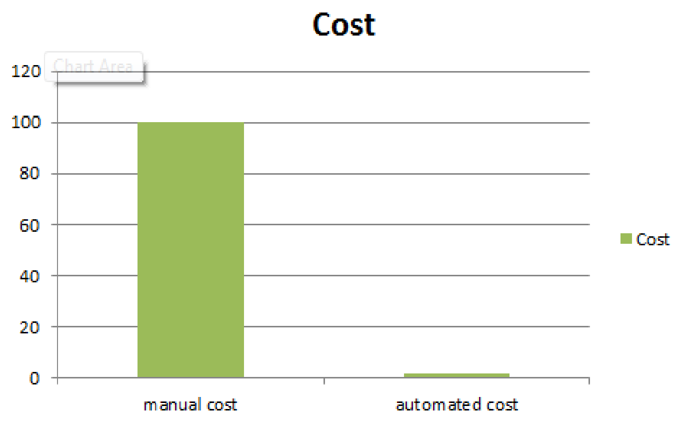


Figure 6: Comparison of Cost

Cost of the testing is the most concerning area of the software testing. From this research, it is found that the proposed automated testing provide cost of 2 units for processing 12 test cases testing as compared to manual testing which used 99 units for same 12 test cases testing.

5. CONCLUSION

In this work we put more work on making a part of the code in which it is tested for the both testing the GUI Interface and testing of the source code also with providing the priority to the test cases that will be executed according to the choice of the user. Our approach is different from other technique because it follows the hypothesis of testing screen transitions of a web application through its functional testing and the different functionalities of the application can lie at any layer except the user interface layer and not only on database layer. Also our approach is targeting the limited time a tester has in order to obtain a certain level of confidence about correctness in Testing. The idea is to extract function signatures to acquire domain knowledge. This domain knowledge will be used to identify equivalence classes.

Equivalence classes will be helpful in generating sufficient number of prioritized test cases to be executed. We have divided testing coverage into different confidence levels achieved by the tester or programmer based on size of test suite and sufficiency of testing technique used. We believe that this step wise approach will facilitate the tester to execute high priority test cases very quickly and to achieve certain level of confidence about the correctness of the application. Due to minimal amount of manual effort involved, the learning curve of our proposed approach will be very low. This will solve the problems of human resource retention and less usage of available automated testing tools in small software industry or any similar type of industry. This approach also deals well with strict time constraints and tight deliverable deadlines that the software industry has. The priority based feature of our approach will also be very useful for developers who want to test their recently developed code very quickly and with a certain degree of confidence.

REFERENCES

[1]http://www.webopedia.com/TERM/S/static_routing.html, date last viewed: 2012-10-11.

[2] Routing protocols and concepts, CCNA exploration companion guide. ‘‘Introduction to dynamic routing protocols’’. Chapter three pages 148 to 177.

[3] http://www.ietf.org/rfc/rfc2501.txt , date last viewed: 2012-10-11.

[4] Jing Xie, Luis Girons Quesada and Yuming Jiang, ‘‘A Threshold-based Hybrid Routing Protocol for MANET’’. Department of Telematics, Norwegian University of Science and Technology.

[5] S. Gowrishankar, T.G. Basavaraju, M. Singh, Subir Kumar Sarkar, ‘‘Scenario based Performance Analysis of AODV and OLSR in Mobile Ad hoc Networks’’, Jadavpur University, Acharya Institute of Technology India.

[6] Lijuan Cao Kashif Sharif Yu Wang Teresa Dahlberg, ‘‘Adaptive Multiple Metrics Routing Protocols for heterogeneous Multi-Hop Wireless Networks’’, Department of Computer Science, University of North Carolina at Charlotte, Charlotte, USA.

[7] Ammar Zahary and Aladdin Ayesh, ‘‘Analytical Study to Detect Threshold Number of Efficient Routes in Multipath AODV Extensions’’, Faculty of Computing Sciences and Engineering De Montfort University Leicester, LE1 9BH, UK.

[8] Deep Kaur ,Kirandeep kaur, Vishal Arora, “QoS in WLAN using IEEE802.11e”, 2012 Second International Conference on Advanced Computing &

Communication Technologies, IEEE Computer Society, 2012.

[9] Seyed Ali Hosseini, Hamid Farrokhi, PhD," The Impacts of Network Size on the Performance of Routing Protocols in Mobile Ad-Hoc Networks", Second Pacific-Asia Conference on Circuits, Communications and System (PACCS), pp-18-22, volume-1,IEEE, 2010.

[10] P.Kuppusam, Dr.K.Thirunavukkarasu, Dr.B.Kalaavathi, "A Study and Comparison of OLSR, AODV and TORA Routing Protocols in Ad Hoc Networks", pp-143 – 147, volume-5, ICECT, IEEE, 2011.

[11] Sung-Hee Lee, Young-Bae Ko, Youg-Geun Hong, and Hyoung-Jun Kim, ''A New MIMC Routing Protocol Compatible with IEEE 802.11s based WLAN Mesh Networks'', pp-126-131, ICOIN, IEEE, 2011.

[12] B.Soujanya, T.Sitamahalakshmi," Study Of Routing Protocols In Mobile Ad-Hoc Networks", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 4, pp 2622-2631 April 2011.