

Effective Software Testing Approach for Priority Based Testing

Supriya¹ Manish Mahajan²

¹Research Scholar, Dept of Information Technology, CEC Landran.,

²Professor, Dept of Information Technology, CEC Landran.,

Abstract: Testing is an important process that is performed to support quality assurance. Testing activities support quality assurance by gathering information about the nature of the software being studied. Software test cases prioritization is the process used for providing priority to the test cases which are built by dividing the number of test cases in to number of modules. Selection of test cases is also a challenge in the testing and priority of testing for various test cases are also required. In this paper, we put more work on priority of the test cases and also shown the comparison over priority of the test cases and without priority of the test cases. 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

The software applications once developed need to be maintained and tested as they undergo regular or frequent modifications. Automation of software testing and tool support for testing, therefore, has been emerging as a very important technology to quality assurance of present software industry. In this paper we describe several techniques for prioritizing test cases and report our empirical results measuring the effectiveness of these techniques for improving rate of fault detection. The results provide insights into the tradeoffs among various techniques for

test case prioritization.

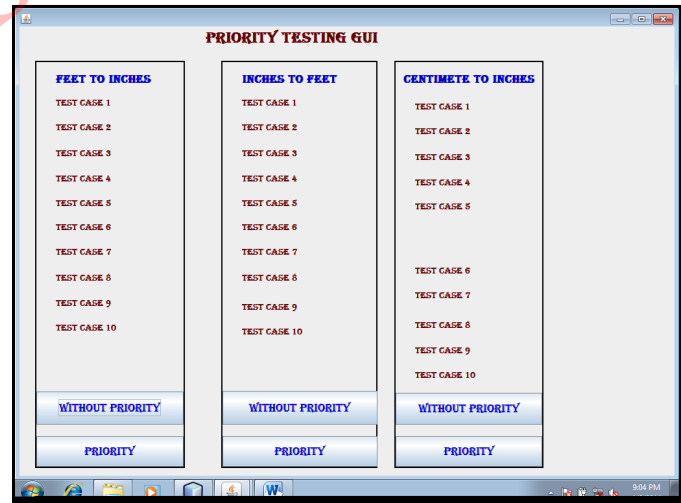


Figure 1: Graphical Interface for Module of the Priority Testing

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

2. AUTOMATED PRIORITY BASED TESTING

In our research we have focused on building automation tool for Graphical user interface testing in which testing framework generates user interface events such as keystrokes and mouse clicks, and observes the changes that result in the user interface, to validate that the observable behavior of the program is correct. Developing a model-based automated testing tool to ameliorate the testing situation at small software companies. 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. So, if first click on priority and then graph. The message will be displayed showing first click on without priority to get the graph. It will also save time for the user.

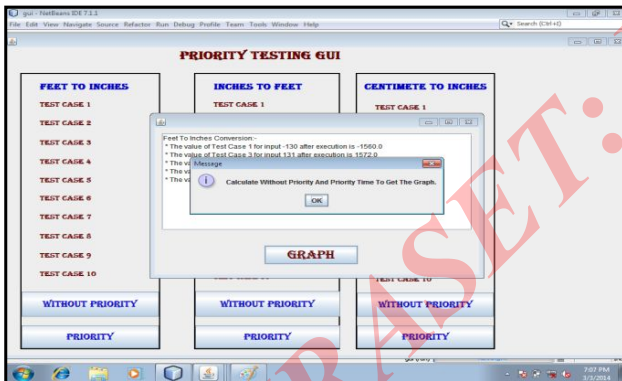


Figure 2: Module of the Non Priority Testing

3. PROBLEM DESCRIPTION

A process of executing a program with the goal of finding errors. So, testing means that one inspects behavior of a program

on a finite set of test cases. 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. Software applications once developed need to be maintained and tested as they undergo regular or frequent modifications. So automated testing becomes essential for priority testing as it saves resources while doing testing.

4. EXPERIMENTATION DONE WITH DISCUSSION OF RESULTS

In this work we put more work on testing both priority part and non priority of the testing module. In non priority part all the 10 test cases will execute. And in priority part of the test cases only 5 test cases will execute.

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

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

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. Accordingly test cases are performed by the user.

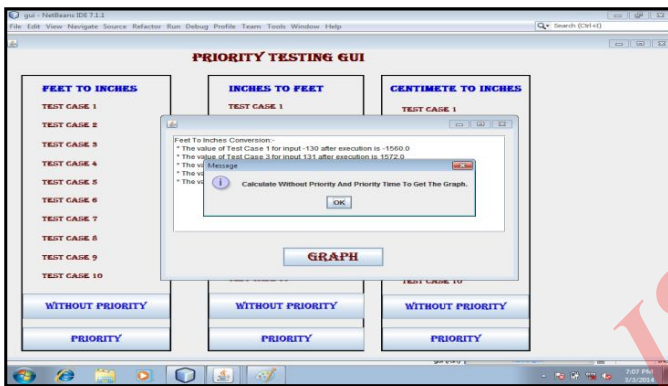


Figure 3: Output Module of the Priority Testing

The figure 3 shows the most important module of the software testing that is providing the Priority to the test cases. In this test cases are divided in to three parts, feet to inches, inches to feet and centimeter to inches. Also provided with two buttons at the bottom without priority and with priority. User has to click on any button and the output of the test cases will be displayed on the frame.

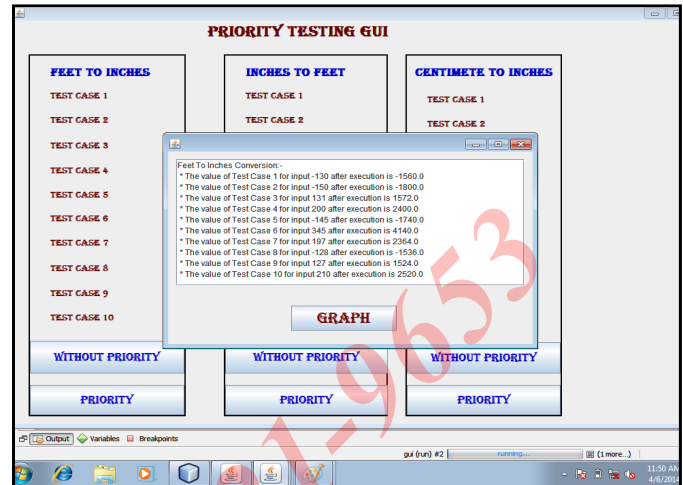


Figure 4: Output Module of the Non Priority Testing

Showing the output for non-priority base testing modules. In this output of all the 10 test cases will be displayed on the frame. In non priority testing user has to click on the without priority button at the bottom and the output for all the test cases will be displayed on 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 5 shows the comparison of the manual and automated testing in term of time taken

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

The Automation testing is the good solution for various application testing and provides better and faster testing results. Automation testing also requires less man power as comparison to the manual testing.

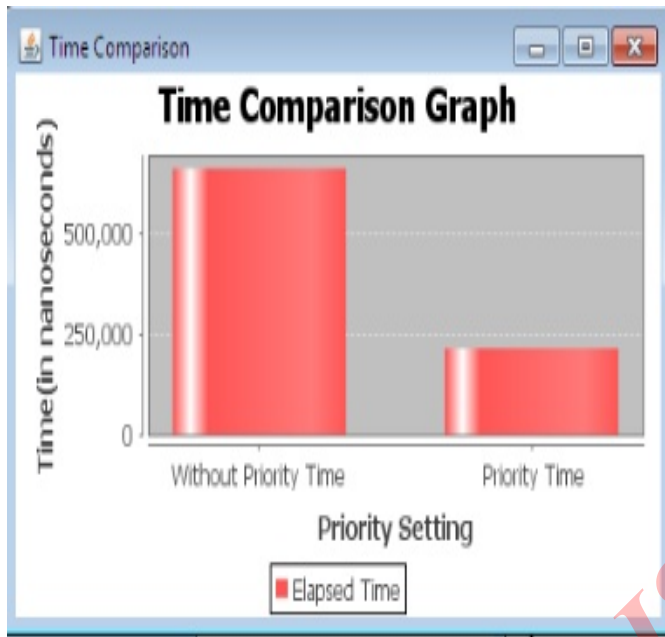


Figure 6: Time Comparison for Module of the Priority Testing showing the comparison of time consumption for testing in case of priority based testing and non-priority based testing. Clearly priority based testing have advantage over non priority module. In priority wise testing it requires less time to test the test cases and in non-priority testing it takes much more time to test the test cases. Testing on the priority based have major advantage. This clearly shows the less time in performing the priority based module.

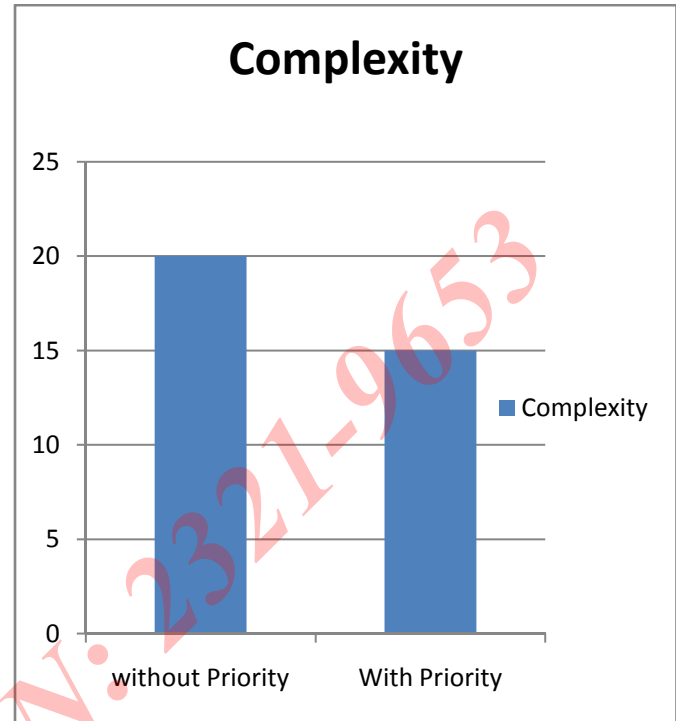


Figure 7: Complexity comparison of the software development testing.

Complexity of the testing is always huge parameters for the user which decides the overall performance, accuracy and output cost of the testing. Testing based on priority from users is less as compared to complexity occurred without priority from users. Complexity decides how our test cases are complex. Complexity covers number of nodes and number edges covered by the test case. In longer sense the testing process without priority from user id providing more complexity to overall process. Better complexity can be achieved in case of priority testing than non priority testing.

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

5. CONCLUSION

Here an internal perspective of the system, as well as programming skills, is used to design test cases. The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. The black-box approach is the method where the software under test has to be verified with a suitable studied set of inputs whose expected outputs are known only on the basis of the functional specifications. The proposed framework can be considered to be the black-box approach. In this work, we have focused on the software testing with the priority of the user so that the user need to run only some of the task so that the time in the execution of the rest of the test cases are with the testing of the various component of the GUI like checkbox, button, textbox.

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. 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.

REFERENCES

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

[2] Katja Karhu, Tiina Repo, Ossi Taipale, Kari Smolander, "Empirical Observations on Software Testing Automation",

IEEE International Conference on Software Testing Verification and Validation, pp.184-192, Vol.2, No.6, 2009.

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

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

[5] W.-K. Chen and Z.-W. Shen, "GUI test-case generation with macro-event contracts," in 2nd international Conference on Software Engineering and Data Mining (SEDM 2010), June 2010, pp. 145-151

[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] T. Mens and T. Tourwe, "A survey of software refactoring," Software Engineering, IEEE Transactions on, vol.30, no. 2, pp. 126-139, Feb 2004

[8]International Conference on Advanced Computing &Communication Technologies, IEEE Computer Society] Deep Kaur ,Kirandeep kaur, Vishal Arora, "QoS in WLAN using IEEE802.11e", 2012 Second, 2012.

[9] Mikhail Auguston, James Bret Michael, Man-Tak Shing, "Test Automation and Safety Assessment in Rapid Systems Prototyping", 16th IEEE International Workshop on Rapid System Prototyping (RSP'05), pp.14-17, October 2012.

INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERING TECHNOLOGY (IJRASET)

- [10] P. Kuppasam, 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.
- [13] W.-K. Chen, Z.-W. Shen, and T.-H. Tsai, "Integration of specification based core-based approaches for GUI testing," J. Inf. Sci. Eng., vol. 24, no. 5, pp. 1293-1307, 2008.
- [14] P. Nagarani, R. Venkata Ramana Chary, "A Tool Based Approach for Automation of GUI Applications", ICCCNT'12, pp. 34-36, Vol. 48, No. 3, 26 July 2012.
- [15] T. Mens and T. Tourwe, "A survey of software refactoring," Software Engineering, IEEE Transactions on, vol. 30, no. 2, pp. 126-139, Feb 2004.
- [16] G. Florin, C. Fraize, S. Natkin, "Stochastic Petri nets: Properties, Applications, and Tools," Microelectronic Reliability, Vol. 31, No. 4, pp. 669-697, 1991.
- [17] Mikhail Auguston, James Bret Michael, Man-Tak Shing, "Test Automation and Safety Assessment in Rapid Systems Prototyping", 16th IEEE International Workshop on Rapid System Prototyping (RSP'05), pp. 14-17, October 2012.
- [18] C. Y. Wong, T.S. Dillon, K.E. Forward, "Timed Places Petri nets with Stochastic Representation of Place Time," Proc. International Workshop on Timed Petri nets, Torino, 1985, pp. 96-103.