Testing of Software development by Graphical Interface Tool

Supriya[1], Dr.Navdeep Kaur[2]
[1] Research Scholar, Dept of Computer Science, CEC Landran.,
[2] Professor, Dept of Computer Science, CEC Landran.,

ABSTRACT: Software testing is one of the essential parts of the software development life cycle testing for acceptance of user. In these days size and complexity are most essential parts of hardware and software to take care of while doing testing. Most of the system crashed due to inappropriate hardware testing. Manual testing for hardware becomes more difficult due to limitation of time and resources. One solution for improving the effectiveness of hardware and software testing is to perform automation testing. In this approach, testers can focus on critical software features or more complex cases, leaving repetitive tasks to the test automation system for hardware testing. Some of the Open source automation tools that are present are AutoIT (for Windows Application), Selenium (for Web-based applications), WatiR, WatiN etc which are mainly used for software testing only. In our research we will focus on building tool for Graphical user interface testing in which testing framework generates user interface events. Main focus will be on decreasing the execution time of testing and testing on priority part of the particular software. Priority of the hardware for testing will be decided by the user. Parity test will be the primary focused part. In related study, testing tool is platform dependent and we will make it platform independent with java platform.

Keywords: *Software Development Life Cycle, Software testing, regression testing, Graphical Interface*

1. STRESS TESTING AND REGRESSION TESTING

Software testing is often used in association with the terms verification and validation. Verification is the checking or testing of items, including software, for conformance and consistency with an associated specification [9]. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted [10]. Software testing is a set of activities conducted with the intent of finding errors in software

Stress testing is the type of testing in which, full capacity of particular hardware or software will be judged based on number of resources, number of users and numbers of transitions per second. In Normal environment stress testing is only recommended for software or live servers but it is very important to consider the hardware part also. Due to lack of stress testing in hardware, performance of software use to go down or perhaps sometime software process got crashed. Stress test required live machinery or live databases for testing purposes.

Regression testing is any type of software testing that seeks to uncover software errors by retesting a modified program. The intent of regression testing is to provide a general assurance that no additional errors were introduced in the process of fixing other problems or modifications of software [2]. Regression test suites are often simply test cases that software engineers have previously developed, and that have been saved so that they can be used later to perform regression testing [3].

2. HARDWARE AND SOFTWARE TESTING

© 2013 IJAIR. ALL RIGHTS RESERVED

Testing is important if we are seeking quality in overall system development. Most of testing methods only focus on software testing partially to save resources and only little further focus on hardware testing. Software testing is incomplete without testing the capacity of the hardware.

Hardware testing is much more important in military and any sensitive operation in which failure of any component can trigger break down to whole system. In base paper, author emphasize on the hardware fault finding by testing the hardware modules along with minor software testing. In our research we will also focus on both hardware testing and software testing so that we can detect issues in overall system. One solution for hardware testing is to simulate the hardware and apply tests on it virtually or to find fault by hardware parity testing. Other mechanism is to connect to live real time machines and work on them for further testing of hardware. The purpose of our proposed work is to describe a model-based test methodology that incorporates hardware design and characteristics into test software and enhances the test accuracy in terms of hardware fault isolation. The test model will be considered throughout the system development life cycle starting from the first phase of establishing the test model

3. BENEFITS OF TESTING

Test Cost Reduction - Perhaps one of the most important reasons for using testing is its improved efficiency and cost savings. It has been repeatedly documented in [11] that using this technique improves the quality of the software [12], while also reducing the amount of effort. Defects are spotted in earlier stages of the process, making them not as expensive to fix as if they were found later [13][14].

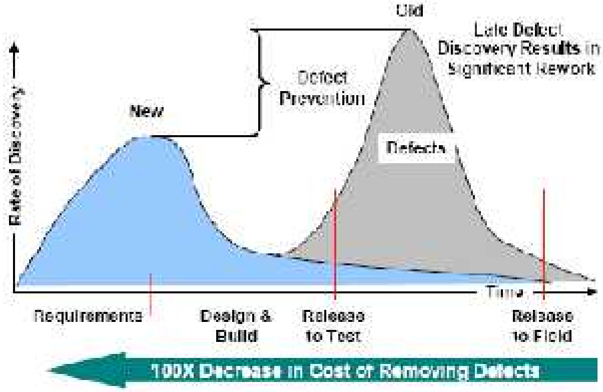

Figure 1: Defects are found earlier, reducing costs [11]

Figure 3 in [11] shows a comparison on cost reduction while using model-based testing. The gray curve shows the cost in a conventional test process, and the blue curve presents the cost in a model-based testing process [14].

Good documentation - One advantage of model-based testing is its easy production for detailed documents. According to [13], these documents can be used as a training tool for new testers in a team because they can learn and understand properties and behaviors of the under test product by reviewing its models [15]. In addition, the model-based artifacts are very useful for testers to understand the model-based generated tests and test scripts because most of them are lack of detailed comments.

4. PROBLEM DEFINITION

Market trends with demands for faster time-to-market and higher quality software continue to pose challenges for software companies that often work with manual test practices that cannot keep up with increasing market demands. Companies are also challenged by their own systems that are often Graphical User Interface (GUI) intensive and therefore complex and expensive to test [16], especially since software is prone to changing requirements, maintenance, refactoring, etc., which requires extensive regression testing [17].

In our proposed work, we will focus on automation tool which will test the hardware and software for better

© 2013 IJAIR. ALL RIGHTS RESERVED

accuracy and performance of overall system. The testing tool will be developed based on the concept of finding basic faults in hardware by detecting misbehavior of hardware (such as slow processing) and software by doing stress and regression testing on total Software development life cycle including hardware and complete testing.

5. OBJECTIVES OF THE STUDY

To fulfill our research properties we have following objectives:

- ✓ To analyze the existing stress and regression testing techniques for testing of code.
- ✓ To find optimized results for hardware testing and software testing.
- ✓ To develop and implement the tool for testing based on various parameters.

6. RESEARCH METHODOLOGY

Our research will start with study of stress testing and regression testing. Java language will be used to develop a testing tool which will automatically judge the faults in hardware and software. Developed tool will have many properties such as managing 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 and further will detect the hardware anomalies. Automation tool will be based on java development and will be capable of performing various operations like Efficiency in Recording, Ability to generate scripts, Data Driven Testing, Reporting of test results, Script Reusability, Playback of the scripts and partial code testing. For core hardware testing we will fetch information based on behavior of hardware and if some observation will be found then we will start stress testing on hardware. Software testing with regression testing in mind will be carried out for complete results.

7. CONCLUSION

The proposed work will bring stability in software development life cycle y introducing testing at good level for both hardware and software testing. In our ongoing research we have also done some experimentation for testing module. Final regression testing is ongoing with graphical interface testing automation tool.

REFERENCES

[1] John Wu, ‘‘Stress Testing Software to determine Fault Tolerance for Hardware Failure and Anomalies’’, IEEE Transactions, Vol.34, Issue.6, pp.345-356, June 2012.

[2] R. Savenkov,” How to become a software tester”, Roman Savenkov Consulting, 2004.

[3] Alexey G. Malishevsky, Joseph R. Ruthruff, Gregg Rothermel, Sebastian Elbaum, “Cost-cognizant Test Case Prioritization,” Technical Report TR-UNL-CSE-2006-004, Department of Computer Science and Engineering, University of Nebraska–Lincoln, Lincoln, Nebraska, U.S.A., March 2006.

[4] Aaron Marback, Hyunsook Do, and Nathan Ehresmann,” An Effective Regression Testing Approach for PHP Web Applications”, IEEE Fifth International Conference on Software Testing, Verification and Validation, 2012.

[5] Tony Cowling,” Stages in Teaching Software Testing”, IEEE Conference on Software Education, Zurich, Switzerland Software Engineering Education, ICSE 2012.

© 2013 IJAIR. ALL RIGHTS RESERVED

[6] Huang,Chin-Yu, Lin,Chu-Ti,”Software reliability analysis by considering fault dependency and debugging time lag”, IEEE Transactions, vol. 55(3), 2006, pp. 436-450.

[7] Xiao Qu, Mithun Acharya, Brian Robinson,” Configuration Selection Using Code Change Impact Analysis for Regression Testing”, 28th IEEE International Conference on Software Maintenance (ICSM), 2012.

[8] Christ of Budnik, Siemens,” Software Testing, Software Quality and Trust in Software-Based Systems”, Siemens.com.

[9] N. Prakash, T. R. Rangaswamy,“ Multiple Criteria Based Test Case Prioritization for Regression Testing”, European Journal of Scientific Research, Vol. 84, No.1, February 2012, pp.36 - 45.

[10] Mohd. Ehmer Khan,” Different Forms of Software Testing Techniques for Finding Errors”, IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 1, May 2010.

[11] Blackburn, M., Busser, R., and Nauman, A. "Why Model-Based Test Automation is Different and What You Should Know to Get Started", In International Conference on Practical Software Quality, 2004.

[12] L.Apfelbaum and J. Doyle, “Model-Based Testing”, Software Quality Week Conference. May 1997.

[13] A. Pretschner, W. Prenninger, S. Wagner, C. Kühnel, M. Baumgartner, B. Sostawa, R. Zölch, and T. Stauner, “One Evaluation of Model-Based Testing and Its Automation,” Proceedings of the 27th International Conference on Software Engineering, St. Louis, Missouri, ACM, New York, pp. 392–401, May 15–21, 2005.

[14] M. Blackburn, R. Busser, A. Nauman, and R. Chandramouli, “Model-Based Approach to Security Test Automation.”, Proceedings of Quality Week 2001. June 2001.

[15] I. K. El-Far and J.A. Whittaker. Model-based software testing. In John J. Marciniak, editor, Encyclopedia of Software Engineering, Vol.1, pp.825–837, Wiley-InterScience, 2002.

[16] Emil Borjesson and Robert Feldt,” Automated System Testing using Visual GUI Testing Tools: A Comparative Study in Industry”, IEEE Conference, 2012

[17] P. Li, T. Huynh, M. Reformat, and J. Miller, “A practical approach to testing gui systems,” Empirical Software Engineering, Vol.12, No.4, pp.331–357, 2007.

© 2013 IJAIR. ALL RIGHTS RESERVED