Advanced Science and Technology Letters
Vol.136 (ITCS 2016), pp.116-119
http://dx.doi.org/10.14257/astl.2016.136.28

Test Framework Development for Software Reliability Test using Formal Method

[1]Jeong-In Park, [2]Sung-Hun Kim, [3]Hansol Lee, [4]Jin-Tak Choi

EJSOFT.Corp.[1], Incheon National University, Korea[2,3,4]
jipark@ejsoft.co.kr[1], pcmania@inu.ac.kr[2], sol_dream@naver.com[3], choi@inu.ac.kr[4]

Abstract. This paper introduced formal method as a new method to evaluate the reliability of software that has been developed. By using a mathematical notation called formal specification, this evaluation method describes the characteristics and functions of system that includes software or hardware and proves what can be proved using a proving technique. It can give a clearer answer than other techniques because it guarantees completeness, correctness, and consistency, which are the general characteristics of math, and evaluates reliability by mathematical technique. Since it is difficult to express a mathematical notation in a UI, a framework was developed by using it to show the measured results on a screen like other general software. This framework was named FMTF (Formal Method Test Framework) and was made to test the program of general web development. For the performance experiment of FMTF system, the parallel test, loop test, conditional branch perform test, time control test, etc. of web program were experimented. According to the measurement result, the reliability was low in the beginning of project due to many factors such as analysis design but gradually improved by error correction and debugging. By suggesting a guide on how to correct what errors, the paper could also provide a method to improve reliability.

Keywords: Software Reliability, Formal Method, Software Test, Test Framework

1 Introduction

Many processes besides coding are needed to develop software and improve its reliability. That is, while creating a program is coding, there are many other important tasks to be done before and after coding which are rather much more difficult and important than coding. In fact, as in Figure 1, the distribution of manpower and time put in for software development is much more concentrated on the activity of previous stage than the programming stage[1].

Another way to improve reliability is planning on what product to make, i.e., making an important user requirement analysis. This process is also an important element in reliability improvement and is sometimes much more difficult than the programming activity.

Recently, many researches have been conducted on formal method as a technological alternative to improve the reliability of software. Formal method states

ISSN: 2287-1233 ASTL

the characteristics and functions of system that includes software or hardware using a mathematical notation called formal specifications [2][3], and can be defined as a technique to prove what can be proved. Therefore, this method must be able to prove the mathematical completeness, correctness, and consistency on specifications. If the formal method is more excellent than other methods, it can give a clearer answer on an application to prove reliability and safety than any other existing methods.

2 Background

After handing over the software that finished development and analyzing the errors reported for a year, we could learn that more than 2/3 of errors occurred from the requirement analysis stage, the initial stage of system development[4][5]. Also, if an effort to correct an error found in a system requirement definition process is assumed to be 1, it is known that 1.5~1.6 times of effort is needed in the development stage and 60~100 times of effort is needed in the maintenance stage. Such series of statistics show how important it is to clearly find the user requirements of what users want in system development rather than coding itself.

Requirement specifications are used as a means of communication among people with different perspectives. If the information expressed in the specifications is unclear and vague, it will be interpreted randomly depending on who understands it. Thus, information can change greatly from its original intention when it is delivered from a user to a system analyzer or from a designer to a programmer.

As a method to alleviate such problem, many techniques such as structural analysis technique, object-oriented design technique, entity relation model, etc. have been suggested [6][7]. These method express a system using figures and accompany all kinds of design guidelines and support tools, but none of these can provide a special solution if the verification of software is required. Therefore, a formal method that applied mathematical notation was adopted and software reliability can improve using this mathematical technique.

3 Development of Test Automation Framework

FMTF is a specifications writing tool and test automation tool that was developed by applying the functions of test automation tool called FitNesse and STAF.

FMTF maximized its efficiency by combining the reusability, expandability, and the distributed environment support functions of STAF and the visible test case design and report functions of FitNesse. The operating principle of FMTF is as in Figure 3: a user writes a test in a table form on a random test page and click on the ‘Run the test’ button, which then performs the test through the FIT server. The user can perform the test of distributed environment using the service of STAFF or directly use and develop the functions s/he needs as a service. The final test result is shown to the user in a table form. The test is shown in green if it succeeds and in red if it fails.

To perform parallel test in the distributed environment, process delay for 1 second through the ‘DELAY’ service of STAF in 3 servers (server 1, server 2, server 3) was

written in three tables respectively and the block was designated in a 'parallel' statement to perform the command in three table form. The 'parallel' statement will create threads as many as the number of tables regardless of the order they are written and run the threads at the same time. Their performance time is compared to determine whether all three works performed at the same time or sequentially. The value of test result symbol variable '$FMduration$' was 953 When the 'parallel' statement was changed to a 'sequence' statement to perform sequentially, the perform time was 3112 (may vary depending on the performance of computer that conducts the test). Therefore, the performance time of 'parallel' statement is about 1/3 of the processing time of 'sequence' statement, showing that there is parallel processing.

4 Future work & Conclusion

There are various methods to measure software reliability, but this paper tested reliability by applying a mathematical technique called formal method and developing a framework that can measure reliability. The test performed parallel test, loop test, conditional branch test, time control test, etc. to experiment how the reliability of web-based software was measured.

As a result, the reliability was low in the beginning but gradually improved by error correction and debugging and then it improved considerably by suggesting a guide on how to correct what errors. The future research must be on the commercialization of FMTF to be used as an application.

Acknowledgement. This research was supported by a grant (12-TI-C01) from Advanced Water Management Research Program funded by Ministry of Land, Infrastructure and Transport of Korean government.

References

1. Hong, Y.-G., "The Research Trends of Early Defect Prediction for Improving of Software Reliability", The Korean Institute of Industrial Engineers, Fall Conference, pp. 288-298 (2007.11)
2. Lee, M.-J., Kim, I.-H., Choi, J.-S.: "A proposal for method improving reliability of BIM-based Quality Checking software", The Architectural Institute of Korea, Fall Conference, pp.567-568, (2015.10)
3. Joo, U.-G., Lee, C.-H., Kim, J.-B.: "A Survey Formal Verification Methods", The Korean Operations Research and Management Science Society, Fall Conference, pp.297-300, (2003.11)
4. Kim, Y.-J., Kang, M.-Y., Choi, J.-Y.: "Verification of Security Properties for M2M Protocol using Formal Methods", Korea Institute of Information Scientists And Engineers, Fall Conference Vol.38, No.2, pp.164-167, (2011.11)
5. Cho, H.-J., Hwang, J.-G., Yoon, Y.-G.: "Applying Methodology for the Safety-Critical S/W Development of Railway Signaling with the Z and Statechart Formal Method", KIEE, Vol.57P, No.2, 65-71, (2008.6)

6. Kim, H.-C.: "NHPP Software Reliability Model based on Generalized Gamma Distribution", The Korea Society of Computer & Information, Vol.10, No.6, pp.27-35, (2005.12)
7. Kim, D.-H., Nam, K.-H.: "A Study on Software Reliability Assessment Model of Superposition NHPP", Journal of the Korean society for quality management, Vol.36, Issue 1, pp.89-95, (2008)

