Early Reliability Prediction: An Approach to Software Reliability Assessment in Open Software Adoption Stage

Wangbong Lee[†], Boo-Geum Jung[†], Jongmoon Baik[††]

[†]*Electronic Telecommunication Research Institute*
leewb,bgjung@etri.re.kr
[††]*Information and Communication University*
jbaik@icu.ac.kr

Abstract

Conventional software reliability models are not adequate to assess the reliability of software system in which OSS(Open Source Software) adopted as a new feature add-on because OSS can be modified while the inside of COTS(Commercial Off-The-Shelf) products cannot be changed. This paper presents an approach to software reliability assessment of OSS adopted software system in the early stage. We identified the software factors that affect the reliability of software system when a large software system adopts OSS and assess software reliability using those factors. They are code modularity and code maintainability in software modules related with system requirements. We used them to calculate the initial fault rate with weight index(correlated value between requirement and module) which represents the degree of code modification. We apply the proposed initial fault rate to reliability model to assess software reliability in the early stage of a software life cycle. Early software reliability assessment in OSS adoption helps to make an effective development and testing strategies for improving the reliability of the whole system.

1. Research Background

Nowadays, the use of OSS is increasing and the importance of its role is becoming high in many applications [1][4]. According to the Gartner's report, about 80 percent of all commercial software will include elements of open-source technology by 2012 [11]. Many commercial products use OSS in many fields such as embedded systems, web management systems, and software development systems. The studies of the reliability of OSS has been increasing in software engineering arena [2][3][4].

Among previous researches for OSS reliability, [9] is concerned on the software development environment of OSS for assessing software reliability. It proposed software reliability assessment methods for the concurrent distribution system development, an OSS development paradigm. It concluded that the logarithm Poisson execution time model fits better than the other SRGM's for the actual data set. Another related research, [2] proposed an empirical approach to build a general reliability model for OSS-based software development projects. It concluded that open source project shows the similar reliability growth pattern with that of the closed source projects. Also, it showed that general Weibull distribution offers a possible way to establish a reliability model.

Most previous works are interested in the reliability of OSS itself in this manner. However, many OSS are modified and adopted in software products to support high performance, reliability and functionality [5]. Previous proposed methods are not appropriate to estimate the reliability of OSS adopted systems because OSS has to be modified during the development of OSS adoption.

2. Approach

To assess the reliability of OSS adopted software systems in the early stage, we build a hypothesis; the requirements, the modularity and the maintainability of OSS will affect the reliability of OSS adopted software systems. To support and prove this hypothesis in assessing software reliability, the following steps are proposed.

1) Analyze requirements specification and find weight indexes by mappings between requirements and software modules which are linked with requirements.

A software requirements specification is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions. By some previous studies, a requirements specification is the basic element to make

978-0-7695-3266-0/08 $25.00 © 2008 IEEE
DOI 10.1109/SSIRI.2008.46

modify OSS modules and developed software modules [12]. Thus, we analyze requirements specification and get weight indexes which are based on number of relationship between requirements and modules for calculating the initial fault rate.

2) Measure product metrics of software modules such as number of lines of code, cyclomatic complexity, code maintainability, and code modularity.

We find and measure software metrics of OSS to measure the code maintainability and the code modularity. According to the study of Linux kernel development [8], code modularity lets many programmers extend the program by working on separate modules, without need to change or understand the core system, or interfere with each other's progress. This reduces the risk of the new bugs which can be introduced while changing other modules.

We use the Module Interaction Index(MII) to represent the code modularity which is a principal factor affecting software reliability. Another concerning factor is code maintainability. Code maintainability is the core quality issue in OSS development [7]. For this reason, we use the Maintainability Index(MI) to represent code maintainability. SEI already chose the Maintainability Index as the most suitable tool for measuring the maintainability of systems with high-quality requirements [10].

3) Calculate the initial fault rate from the collected metric values. Apply the initial fault rate to an appropriate model(one of the candidates is Weibull distribution model which is suggested by [2]) and assess the software reliability

In step 3, we calculate the initial failure rate of OSS adopted system using a modified method from [6] and apply the convention model with proposed initial failure rate and calibrate the assessment of software reliability. Following is the initial fault density prediction(δ) and the initial failure rate(λ) in a given module.

$\delta = A*D*(SS*SM*SMI)$

A(Application Type), D(Development environment), SS(Software Size), SM (Software Modularity), SMI (Software Maintainability).

$\lambda = \omega*F*K*(\delta$ *number of lines sources code)

ω is the weight index

F is the linear execution frequency of the program

K is the fault exposure ratio

4. Verification

The proposed method to assess software reliability has to be verified by case study analyses. We collect the failure data from the development of OSS adoption in a large embedded system. This embedded system is the QoS Router system which consists of more than 50 software components including OSS, commercial software and in-house built software. To verify the proposed approach, we compare early predicted values by proposed method with the actual values. Also, we should compare values from the proposed method with values from the widely used prediction methods.

5. References

[1] Armand Marchesin, "Using Linux for Real Time Applications," IEEE Software Vol. 21, Issue 5, September 2004

[2] Ying Zhou, Joseph Davis, "Open source software reliability model: an empirical approach," Proceedings of the 5th WOSSE, pp. 1-6, 2005

[3] A. Mockus, R. Fielding, and J. Herbsleb, "Two case studies of open source software development: Apache and Mozilla," ACM TOSEM, Vol.11, Issue 3, July 2002

[4] Paul Luo Li, Jim Herbsleb, Mary Shaw, "Forecasting Field Defect Rates Using a Combined Time-Based and Metrics-Based Approach: A Case Study of OpenBSD," ISSRE'05, pp. 193-202, 2005

[5] David A. Wheeler, "Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!," http://www.dwheeler.com/oss_fs_why.html

[6] Rome Laboratory (RL), Methodology for Software Reliability Prediction and Assessment, Technical Report RL-TR-92-52, volumes 1 and 2, 1992.

[7] Ioannis Samoladas, et al, "Open Source Software Development Should Strive for Even Greater Code Mainatainability," Communications of the ACM, Oct 2004, Vol.47, No.10

[8] Lee, G.K. and Cole, R.E. The Linux Kernel Development As A Model of Open Source Knowledge Creation, Haas School of Business, University of California, Berkeley, Berkeley, CA, 2000.

[9] Yoshinobu Tamura, Shigeru Yamada, "Comparison of Software Reliability Assessment Methods for Open Source Software," Proceedings. 11th ICPDS 2005 Vol 2, 20-22 July 2005 pp:488 - 492 Vol. 2

[10] Maintainability Index Technique for Measuring Program Maintainability Software Technology Review, SEI; http://www.sei.cmu.edu/str/descriptions/mitmpm.html

[11] http://arstechnica.com/news.ars/post/20080205-gartner-80-percent-of-commercial-software-programs-will-include-open-source-by-2012.html

[12] C. Smidts , R. W. Stoddard , M. Stutzke, "Software reliability models: an approach to early reliability prediction," ISSRE, p.132, October 30-November 02, 1996