

Cox Proportional Hazard Regression of CR-fixing Duration in Testing of a Large Scale of Software System

Yi-Kyong Lim, Ho-Won Jung
College of Business Administration
Korea University
lik@kuba.korea.ac.kr, hwjung@korea.ac.kr

Chang-Shin Chung, Kyu-Ouk Lee
K-R Collaborative Development Team
Electronics and Telecommunications Research Institute
{cschung, kolee}@etri.re.kr

1. Introduction

The objective of this study is to identify CR characteristics associated with the CR-fixing duration in the testing phase. Our data set associated with CR-fixing duration includes censored data [1], which censoring arises by discarding the CR because it turns out not actually CR. Although CR-fixing duration of censored data is not known, censored data has a partial information that CR-fixing is not finished before censoring. Ignoring the censored data and only analyzing the subset can lead to misleading results. Conventional statistical methods cannot fully tackle the problem of censored data.

We employed the Cox proportional hazard regression in survival analysis to deal with the censoring data. In addition, we would like to note that this study is not for validation, but is for identifying the patterns of associations and interactions between CR-fixing duration and CR characteristics.

2. Case Study

2.1 Data source

Our data comes from a system test phase of a large telecommunication system. The data consists of 1566 CRs in software produced at 129 weeks system test. The software system consists of 139 functional blocks. The size of each functional block has a range of 2-10 KLOC with over 12 million lines of written code in the Chill language.

2.2 Measurement

The CRs covered in this study were generated in two cases during system test. The first case came from testing division. In this case, the CR-fixing duration begins with tester's generation of CR and ends with the pass of retest. The second case was from developers who were asked to accommodate new requirements (function enhancements or new functions). The CR-fixing duration of the second case was from receiving a formal request of new requirements to passing the system test. Thus, the CR-

fixing duration corresponds to the survival time of CR. If a CR was generated by misunderstanding or being withdrawn by new requirements request, the CR was classified as irrelevant and counted as a censored data. Kañichi and Kanoun[6] used the same procedure as the first case in collecting the data and handling the censored data in a reliability study of a commercial telecommunication system.

The CR-fixing duration was used as a metric in Hewlett-Packard [5]. Also Kan [6] addressed the CR-duration as a metric for software maintenance.

The CR-fixing process can be affected by several factors: software complexity [2], code size [4], human skills [12], development process, methodology [9][10][11], etc. However, this study only concerns CR characteristics such as the number of related functions, severity levels, source modification types, error injection phases, etc.

2.3 Research Method

Survival analysis is a statistical method for studying the occurrence and timing of events. Survival data is a measurement of the length of time until the occurrence of an event such as CR-fixing duration.

In many methods for analyzing survival data, this study utilized the Cox proportional hazard regression of the non-parametric model. The Cox proportional hazard regression is used to model failure time data in censored data. Its advantage is that it assumes any underlying distribution of CR-fixing duration.

2.4 Results

The results from Cox proportional hazard model are summarized in Table 1.

The value of the likelihood-ratio statistic, 52.95, reject the null hypothesis $H_0 : \mathbf{b}_1 = \mathbf{b}_2 = \dots = \mathbf{b}_k = 0$. Thus, at least one of the coefficients is not zero.

The signs of the coefficients state the direction of the relationship. The negative coefficients for "version 2 and

Table 1: Results from Cox proportional hazard model.

Covariates (factor variables)	Levels	Coefficient (b_j)	Standard error	Hazard rate	P -value
No of related functions	2 and over	0.166	0.078	1.180	0.033
Severity	Major	-0.050	0.175	0.951	0.774
	Minor	-0.198	0.129	0.820	0.123
Modification type	Function enhancement	0.039	0.088	1.039	0.660
	New function	0.010	0.087	1.010	0.906
Phase	Implementation	0.069	0.066	1.072	0.297
	System testing	0.218	0.117	1.243	0.062
Version	Version 2	-0.447	0.162	0.639	0.006
	Version 3	-0.446	0.131	0.641	< 0.001
	Version 4	0.524	0.111	1.689	< 0.001

3" indicate that CRs generated in Version 2 and 3 had longer time to be fixed than those generated in other versions. The positive coefficient of Version 4 indicates that CRs in Version 4 were associated with shorter times to be fixed. The coefficients are in the log relative hazard scale. Thus, for a coefficient b_k the value of e^{b_k} is the estimated coefficient called hazard ratio. For example, the coefficient of the number of related functions is $e^{0.166} = 1.180$ that is relative hazard ratio to just one function CR. CR-fixing duration with a large hazard ratio is longer than that with a small hazard ratio.

With p -values in Table 1, we can conclude that the number of related functions, error injection phase, and version are related to CR-fixing duration, while severity level and modification type are not.

The generalized R^2 [3][8] has a value of 0.034. The generalized $R^2 (= 1 - \exp\left(\frac{-G}{n}\right))$ is highly associated with the sample size. Since our data set includes a relatively high volume of observations, the generalized R^2 has relatively small value.

3. Final Remark

In this paper, we analyzed a case study involving 1556 CRs in a testing phase of a large-scale software system. This study utilized the Cox regression in accommodating censored data and assuming the underlying distribution of data.

Our results indicate that the number of related functions, error injection phase, and version are related to CR-fixing duration, while severity level and modification type are not. Since the number of observations are many, the 'generalized R^2 ' has relatively small value.

4. Reference

- [1] P.D. Allison, *Survival Analysis Using the SAS System: A Practical Guide* (SAS Institute, Inc., NC), 1995.
- [2] R.D. Banker, G.B. Davis, and S.A. Slaughter, Software Development Practices, Software Complexity, and Software Maintenance Performance: A field study. *Management science*, **44**, (4), 433-450, 1998.
- [3] D.R. Cox and E.J. Snell, *The Analysis of Binary Data*, 2nd ed. (London: Chapman & Hall), 1989.
- [4] K. El Emam, S. Benlarbi, N. Goel, W. Melo, H. Lounis, and S.N. Rai, The Optimal Class Size for Object-Oriented Software : A Replicated Study, National Research Council of Canada, March 2000
- [5] R.B. Grady and D.L. Caswell, *Software Metrics: Establishing A Company-Wide Program*, New Jersey: Prentice-Hall), 1986.
- [6] S.H. Kan, *Metrics and Models in Software Quality Engineering* (New York: Addison-Wesley Publishing Co.), 1995.
- [7] M. Kaânichi, and K. Kanoun, Reliability of a Commercial Telecommunications System, *Proceedings of the 7th Software Reliability Engineering Conference*, 207-212, 1996.
- [8] L. Magee, R^2 measures based on Wald and Likelihood ratio joint significance test, *The American Statistician*, **44**, 250-253, 1990.
- [9] D.E. Perry, N.A. Staudenmayer, and L.G. Votta, People, organizations, and process improvement. *IEEE Software*, 36-45, July 1994.
- [10] T.L. Robert, M.L. Gibson, K.T. Fields, and R.K. Rainer, Jr., Factors that impact implementing a system development methodology, *IEEE Trans. on Software Engineering*, **24**, (8), 640-649, 1998.
- [11] S. Sawyer and P.J. Guinam, Software development processes and performance, *IBM Systems Journal*. **37**, (4), 552-569, 1998.
- [12] J. Voas, Analyzing Software Sensitivity to Human Error, *Failure & Lessons Learned in Information Technology Management*, Vol. 2, 201-206, 1998.