Comparing Fault Prediction Models Using Change Request Data for a Telecommunication System

Young Sik Park[a], Byeong-Nam Yoon, and Jae-Hak Lim

Many studies in the software reliability have attempted to develop a model for predicting the faults of a software module because the application of good prediction models provides the optimal resource allocation during the development period. In this paper, we consider the change request data collected from the field test of a large-scale software system and develop statistical models of the software module that incorporate a functional relation between the faults and some software metrics. To this end, we discuss the general aspect of regression method, the problem of multicollinearity and the measures of model evaluation. We consider four possible regression models including two stepwise regression models and two nonlinear models. Four developed models are evaluated with respect to the predictive quality.

I. INTRODUCTION

When a software system is developed, the majority of faults are found in a few of its modules [1], [13]. In the case of a telecommunication system, 55 % of faults exist within 20 % of source code as shown on Figure 1 [6]. It is, therefore, much of interest to find out fault-prone software modules at early stage of a project.

Many authors have studied on the relation between the inherent faults and various software metrics. In general, it is believed that the program size strongly related to the inherent faults. Inherent faults mean the faults that are associated with a software product as originally written or modified [10]. Typical coefficient of determination relating the program size and inherent faults ranges between 0.5 and 0.6 [1], [5], [10].

Takahashi and Kamayachi [19] have studied the effect of other factors on improving inherent faults prediction, based on data taken for 30 projects and find three significant additional factors, which are specification change activity, average programming skill and thoroughness of design documentation. In another study, authors have tried to find some factors that impact software reliability or software development performance. Sawyer and Guinan [17] show the effect on software development performance due to production methods of software development and social processes of how software developer work together. Robert, Jr. *et al.* [14] conduct an empirical study and identify five important factors for implementing system development methodology (SDM). Zhang and Pham [20] perform a survey in order to find some factors important to the software reliability.

In the area of the identification of fault-prone modules using software metrics, authors have considered two broad modeling approaches that are the estimative approach and the classification approach. In the estimative approach, regression models

Manuscript received September 1, 1998; revised August 27, 1999.

[a] Electronic mail: ysp@etri.re.kr.

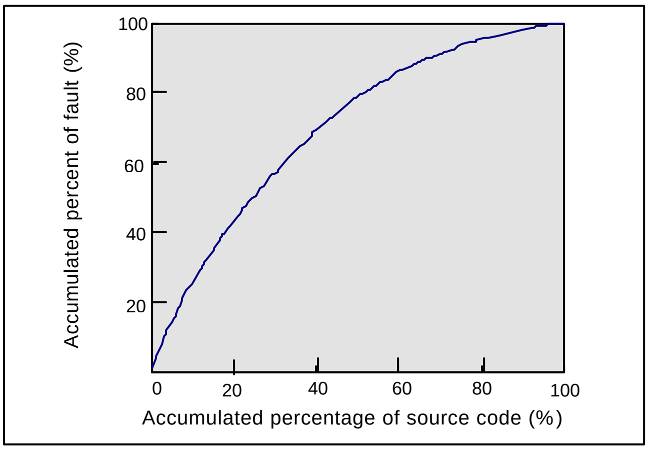


Fig. 1. The relation between the number of faults and source size.

are used to predict the actual number of faults that will be disclosed during testing period [2], [7], [8], [9], [18]. In the classification approach, software modules are categorized into two or more classes. And then classification methods such as discriminant analysis have been used to develop a classification model [9], [15]. Gaffney [4] develops a model which explains the relation between the number of faults and the number of lines of code.

Most of studies are, however, focusing on a small size system whose total sum of source code statements is far below from 1 million lines. And those studies need some detailed data of a software system such as the complexity of the software modules, control flow documents, the amount of modified or reused program and programmers skill and so on [9], [10], [15]. But it is hard to get the above detailed data for a large-scale software system, because of the vast program size, the lack of detailed documents and the incessant change of programmers during the development progress.

In this paper, we consider the change request data (hereafter, CR data) collected from the field test of a large scale switching system and, based on this data, develop a statistical model for identifying fault-prone blocks of a software system. In Section II, the functional structure of the large scale switching system and the CR data is briefly described. And Section III is devoted to summarize the basic statistical issues of regression analysis including the problem of multicollinearity and several methods for evaluating the alternative models. In Section IV, the results of our analysis are presented and some statistical models for fault prediction are recommended. All computational results are obtained by using SAS [16].

II. THE CR DATA OF TDX-10 ISDN SWITCHING SYSTEM

The data considered in this paper is obtained from the field test of TDX-10 ISDN Switching System. This data is called the CR data. The software system of TDX-10 ISDN Switch System developed by Electronics & Telecommunications Research Institute (ETRI) consists of 1.3 million source code statements and 140 software blocks. This software system consists of the following five functional modules:

- Call Processing (CP)
- Data Handling (DH)
- Administration (Ad)
- Operation & Maintenance (OM)
- Kernel (K)

The blocks in call processing (CP) module are mainly related to the job of providing PSTN, ISDN, and packet service. The function of data handling (DH) module includes the collection of data from call processing and insertion, modification and deletion of the data. administration (Ad) module conducts a set of functions related to system administration such as charging and statistics of the subscribers' or trunks' numbers. operation and maintenance (OM) module monitors the state of the system while it is operating. The blocks in kernel (K) module are related to jobs of tasks scheduling, the memory management, and the communication control between processors. The number of blocks and the number of source code in each module are summarized in Table 1.

Table 1. The number of blocks and size of modules.

Categories	Number of Blocks	Size (KLOC)
CP	45	390.7
DH	14	171.9
Ad	44	315.3
OM	29	240.7
K	8	217.5
Total	140	1336.5

When an error occurs during the testing period, the causes of the error are discovered and appropriate correction actions are taken. And a change request (CR) form is reported to manage efficiently the system development and to keep the historical record of the development. The CR data simply contains the description of the error, the action for correcting the error and other related information of the error.

Figure 2 shows the brief flow of collecting CR statements. Once a failure occurs while the software is being tested, a number of CR statements related to the failure are reported. A CR statement simply consists of a problem description part and a summarized correction part. In the review meeting, some of statements are selected to be solved. After the appropriate corrections and tests are conducted, it is decided whether to replace the failed block by the corrected block or not in Configu-

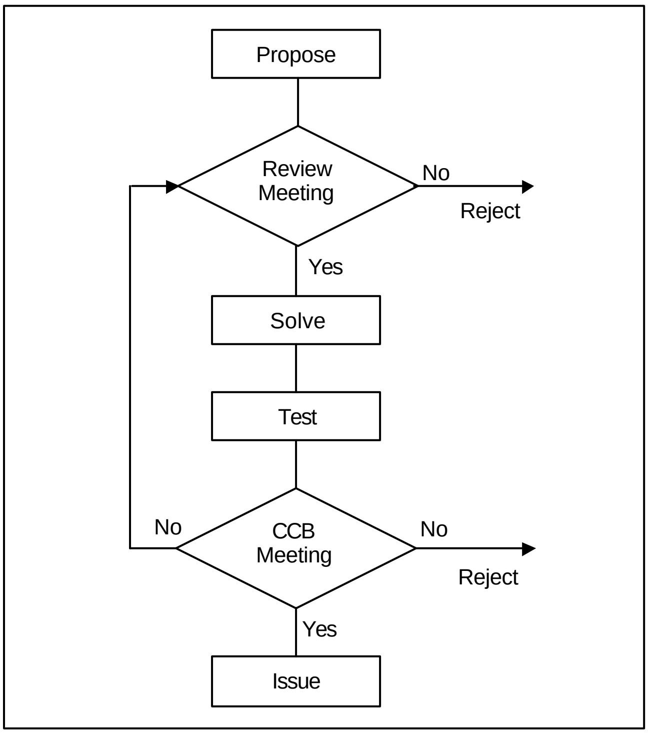


Fig. 2. The flow of collection of CR statements.

ration Control Board (CCB) meeting. For 93 weeks, we obtain 1500 CR statements including CRs due to specification changes.

There are many factors that cause software faults. Most of studies conclude that the size of a software block is strongly related to the faults. But the size factor alone is not enough to explain the faultiness of a software block. Even if several software complexity metrics can be obtained by analyzing the structure of the software, this would be a difficult task for a large-scale software system. The number of messages between software blocks would be another good metric which indicates the complexity of a software block. This means that any block including more messages is apt to have more faults. Also, the number of conditional statements is adopted as a metric because of the close relation between the number of conditional statements and the McCabe' s cyclomatic complexity. The developer's career would naturally be accepted as the other metric; the more experienced the developer is, the fewer faults the developer makes. Since a large-scale system as a telecommunications system has a long-term development and maintenance life, some new software blocks may be added by supplementary requirements from users from time to time. Considering this situation, we consider one more metric, the level of newness of a block. Finally, we consider the number of processes, which might have great effect on the faults.

The sizes of blocks are measured by the number of source code lines including comment lines, and the number of message is the number of statements related to exchange messages between blocks. The developer's career of a module is measured by computing arithmetic average career years of all developers who have been responsible for the module. The levels of the newness of blocks are marked as 1, 2, and 3 according to the development year of the blocks. The blocks of level 1 are considered at the design phase while the blocks of level 3 are added recently.

Software metrics used in this study and their abbreviations are summarized as follows.

LOC	the number of source code including comment lines
DC	the arithmetic average of developers' career (in years) who had been responsible for a block
M	the number of external messages related to a block
N	the level of newness of a block
P	the number of processes in a block
C	the number of conditional statements in a block

III. STATISTICAL METHOD FOR DEVELOPMENT OF FAULT PREDICTION MODEL

Regression analysis is a statistical tool that utilizes the relation between a dependent variable and several independent variables so that the dependent variable can be predicted by independent variables. The model development is to select independent variables from a set of variables so that the most amount of variance in a dependent variables is explained by the selected independent variables. The coefficients for the independent variables are computed by a least square fit of these variables to sample data.

1. The Problem of Multicollinearity

In most regression application, independent variables are sometimes correlated among themselves and with other variables that are related to the dependent variable but are not included in the model. Especially, in software development, software metrics are so strongly correlated that the regression results are ambiguous [7].

Such cases are referred to as the problem of multicollinearity and have following effects on the regression results [12].

(i) Adding or deleting independent variables greatly change the regression coefficients.

(ii) The extra sum of squares associated with an independent variable varies, depending upon which independent variable already is included in the model.

(iii) The estimated regression coefficients individually may not be statistically significant even though a definite statistical relation exists between the dependent variable and the set of independent variables.

A variety of informal or formal methods have been developed for detecting the presence of serious multicollinearity. The presence of multicollinearity can be detected by investigating some diagnostics from regression results. The diagnostics are well summarized in [12]. One formal method of detecting the presence of multicollinearity is by means of variance inflation factors (VIF). It means how much the variances of the estimated regression coefficients are inflated as compared to when the independent variables are not linearly related. Also, the tolerance limit, 1/VIF, is frequently used in many computer programs.

Several approaches to remedy the problems of multicollinearity have been developed. These include regression with principle components, where the independent variables a linear combination of the original independent variables, and Bayesian regression, where prior information about the regression coefficients is incorporated into the estimation procedures. Also Ridge regression is proposed by modifying method of least squares to allow biased estimators of the regression coefficients.

2. Selection of Regression Model

The most widely used method for selection of best regression model is to examine iteratively some potential independent variables. That is referred to as a stepwise regression procedure. The details of this procedure are excellently discussed in [3], [12]. An initial model is formed by selecting a variable with the highest correlation with the dependent variable. In subsequent iterations, the order is determined by considering the partial correlation coefficient with variables in the model as a measure of the importance of variable not yet in the model.

Variables in this model may be removed from the regression equation when they no longer contribute significantly to the explained variance. There must be a priori level of significance chosen for the inclusion or deletion of variables from the model.

3. Evaluation of Regression Models

The effort of developing the prediction model produces more than one possible model. The model evaluation process is excellently discussed in Myers [11]. These are several statistical measures of the performance of a regression model. They are the coefficient of multiple determination $R^2{}_p$, mallow C_p statistic and $PRESS$ statistic.

Traditionally, the $R^2{}_p$ statistic is used almost exclusively empirical studies in software engineering. $R^2{}_p$ is defined a ratio of sum of squares: $R^2{}_p = SSRp/SSTp$, where p represents the number parameters in the regression equation, SSR is the regression sum of squares for the fitted subset regression model with p parameters, and $SSTp$ is the total sum squares.

There is a generic problem associated with the use of $R^2{}_p$. That is, $R^2{}_p$ only increases as independent variables are added to a regression equation even if $R^2{}_p$ does take account of the variance of a dependent variable in the model. A modification of $R^2{}_p$ statistic is the adjusted coefficient of multiple determination, which does not attempt to correct for the parameters in the regression model. This adjusted coefficient of multiple determination, denoted by $adjR^2$, is defined

$$adjR^2 = 1 - \left(\frac{n-1}{n-p}\right)\frac{SSEp}{SSTp},$$

where n is the number of observations and SSEp is the error sum of squares for the fitted subset regression model with p parameters. The larger both $R^2{}_p$ and $adjR^2$ are, the more the model explains the variation of a dependent variable.

The statistic, C_p, is concerned with the total mean squared, error of the n fitted values for each of the various subset regression. The C_p is defined as follows:

$$C_p = \frac{SSEp}{MSE(X_1,\cdots,X_{p-1})} - (n-2p)\ ,$$

where $MSE(X_1,\cdots,X_{p-1})$ is an unbiased estimator of σ^2 and σ^2 is a variance of error term in regression model. The choice of regression model is determined by selecting a model with either the smallest value of C_p or the value of C_p closed to p.

The predicted residual sum of squares (PRESS) statistic is based on residual analysis. This statistic is defined as follows:

$$PRESS = \sum_{i=1}^{n}(\tilde{y}_i - y_i)^2\ ,$$

where y_i is the i-th observed value of dependent variable, $\tilde{y}_i$ is the prediction value from a regression model formed with all observations except the i-th observation and h_i is the leverage of the i-th observation. The smaller the value of PRESS statistic is, the better the regression model predicts. Thus one should choose a model with lowest value of the statistic.

IV. ANALYSIS RESULTS OF CR DATA

1. Preliminary Analysis

The first step to explore the relation between faults and the selected software metrics would be make a plot with the metrics and the faults considered and to compute correlation coefficients, which represent the strength of linear relationship. Table 2, 3, 4 and 5 show the correlation coefficients and their corresponding p-value for module CP, DH, Ad and OM, respectively. We note that the module K is excluded from our analysis since small number of observations might lead us to distorted results.

The correlation structures of four modules are not exactly the same but have some common aspect. For example, LOC, P and C are highly related to the faults for most of modules. It is

Table 2. Correlation coefficient for module CP.

	LOC	M	DC	N	P	C
Faults	0.5961 (0.0001)	0.4293 (0.0033)	0.0714 (0.6412)	0.0709 (0.6435)	0.5368 (0.0002)	0.7873 (0.0001)
LOC		0.3966 (0.0070)	−0.0565 (0.7126)	0.1202 (0.4317)	0.4370 (0.0034)	0.8208 0.0001)
M			−0.1103 (0.4705)	0.0225 (0.8835)	0.5611 (0.0001)	0.5592 (0.0004)
DC				0.0300 (0.8449)	0.1133 (0.4696)	−0.0226) (0.8943)
N					0.2496 (0.1065)	0.1503 (0.3815)
P						0.6662 (0.0001)

Table 3. Correlation coefficient for module DH.

	LOC	M	DC	N	P	C
Faults	0.8983 (0.0001)	0.6164 (0.0189)	0.5277 (0.0524)	0.0728 (0.8046)	0.5475 (0.0528)	0.1735 (0.5708)
LOC		0.6721 (0.0085)	0.6147 (0.0193)	−0.0886 (0.7711)	0.6414 (0.0181)	0.4062 (0.1684)
M			0.2759 (0.3396)	−0.2058 (0.4804)	0.8283 (0.0005)	0.1134 (0.7122)
DC				−0.0564 (0.8481)	0.4116 (0.1623)	0.1807 (0.5546)
N					0.2367 (0.4363)	0.3112 (0.3006)
P						0.3996 (0.1761)

Table 4. Correlation coefficient for module Ad.

	LOC	M	DC	N	P	C
Faults	0.6579 (0.0001)	0.1173 (0.4624)	−0.2976 (0.0498)	0.2466 (0.1066)	0.6179 (0.0001)	0.5689 (0.0002)
LOC		0.0917 (0.5538)	−0.0829 (0.5924)	0.1092 (0.4804)	0.8511 (0.0001)	0.9753 (0.0001)
M			0.1636 (0.2866)	−0.1414 (0.3597)	0.0940 (0.5489)	0.0636 (0.7045)
DC				−0.1032 (0.5050)	−0.0688 (0.6612)	−0.0816 (0.6264)
N					0.0977 (0.5330)	0.1248 (0.4553)
P						0.7889 (0.0001)

Table 5. Correlation coefficient for module OM.

	LOC	M	DC	N	P	C
Faults	0.7654 (0.0001)	0.3170 (0.0938)	0.0319 (0.8695)	0.4750 (0.0092)	0.6838 (0.0001)	0.0635 (0.7629)
LOC		0.3593 (0.0556)	0.2527 (0.1860)	0.1988 (0.3011)	0.6788 (0.0001)	0.3096 (0.1321)
M			−0.1323 (0.4937)	0.2659 (0.1632)	0.5038 (0.0063)	−0.0027 (0.9898)
DC				0.0455 (0.8146)	0.0676 (0.7324)	0.3605 (0.0767)
N					0.4548 (0.0150)	−0.0120 (0.9544)
P						0.2934 (0.1546)

noted from Table 2 ~ 5 that the correlation coefficient between the faults and the number of messages (M) is highly significant for modules CP and DH while it is not for the other modules. That is natural since the blocks related to the job of call processing or data handling need to have more messages to be exchanged. Since the number of messages of a block depends on the specified behavior of the module, the blocks in module having such a behavior would be carefully tested in testing phase. It is also noted that the correlation coefficients between the number of faults and the developer's career do not come up to our expectation. We review the CR raw data again, and conclude the following two possible major reasons for it. One is the existence of well-defined development methodology, which makes the programmer's skill to be independent of the implementation of the product. Another is that the developer's career over a certain degree hardly affects the number of faults.

An inspection of Table 2 ~ 5 shows that the correlation coefficients between some of metrics considered are greater than 0.7. For example, LOC is strongly correlated with the number of processes (P) and the number of conditional statements (C) in Table 3. This reflects the existence of multicollinearity among those metrics. There is also a relatively high correlation between the fault and some of individual metrics. Thus several regression models could be formed that incorporate the software metrics as independent variables and the fault as a dependent variable.

As the remedy of possible multicollinearity in explanatory variables, we conduct a factor analysis with faults as a variable in the analysis. Table 6 shows the resulting factor pattern after factor rotation. It is noted that the dimension of independent variables has been reduced to two factors which are denoted by Factor 1 and Factor 2 in Table 6. Factor 1 has associated with those metrics mostly related to the size dimension such as LOC and P. That is an expected result since the number of processes

Table 6. Factor pattern with fault.

	Module CP		Module DH		Module AD		Module OM	
	Factor 1	Factor 2	Factor 1	Factor 2	Factor 1	Factor 2	Factor 1	Factor 2
LOC	**0.9290**	0.1320	**0.8923**	0.0962	**0.9862**	–0.0373	**0.6411**	0.5440
M	**0.7438**	–0.3962	**0.8704**	–0.1329	0.1372	**0.6951**	**0.7482**	–0.2186
DC	–0.0350	**0.6064**	**0.6426**	0.0283	–0.0788	**0.5948**	–0.2083	**0.7856**
N	0.1298	**0.7609**	–0.1745	**0.8774**	0.0949	**–0.7204**	**0.6270**	–0.0776
P	**0.8650**	0.1357	**0.8400**	0.3474	**0.9187**	0.0198	**0.8554**	0.2751
C	**0.8938**	0.0558	0.3282	**0.7247**	**0.9619**	-0.0606	0.1161	**0.7754**

Table 7. Regression ANOVA for module CP.

Model	Source	d.f.	S.S.	M.S.	F-value
1	Regression Error Corrected Total	1 34 35	23.0872 15.8304 38.9183	23.0879 0.4656	49.59
2	Regression Error Corrected Total	3 32 35	9030.8546 2307.8954 11338.75	3010.2849 72.1217	41.74
3	Regression Error Corrected Total	3 42 45	20871.6849 8201.3151 29073.0000	6957.2283 195.2694	35.63
4	Regression Error Corrected Total	2 43 45	20640.3464 8432.6537 29073.0000	10320.1732 196.1082	52.62

are positively related to LOC. Factor 2 consists of the environmental metrics for developing a software. The values in Table 6, which are referred to as factor loadings, represent the correlations of variables with factors. For example, for the Module CP, the correlation of LOC with Factor 1 is 0.9290 while the correlation with Factor 2 is 0.1320.

From the factor pattern presented in Table 6, factor scores are computed for each observation vector so that the problem of multicollinearity in regression modeling is eliminated. We note that standardized data are used to compute the factor scores.

2. Regression Models for CR Data

In this subsection, we explore the potential use of software metrics considered for their predictive value in terms of factor. To this end, we conduct a stepwise regression with software metrics as independent variables and fault a dependent variable, Also we run another stepwise regression with two factors as independent variable and the fault as a dependent variable.

For the purpose of investigating the more detailed relations, we develop two nonlinear regression models. The close investigation of the relationship between LOC and faults leads the following two nonlinear regression model with only LOC as an explanatory variable. One of nonlinear models is as follows: $y = b_0 + b_1 (LOC)^{b_2}$ where b_0, b_1, b_2 are determined by interactive method. A special case of this model is Gaffney's model in which b_2 is a constant 4/3. Another nonlinear regression model is referred to as Poisson regression. This model is summarized as follows: $y = Exp[a + b\, ln(LOC)]$, where a and b are determined empirically.

The models considered are summarized as follows:

Model 1 A stepwise regression model with the two factors produced by the factor analysis

Model 2 A stepwise regression model with the software metrics as independent variables

Model 3 A generalized model of Gaffeny's model with LOC as an independent variable

Model 4 A Poisson regression model with LOC as an independent variable

Table 7, 8, 9, and 10 show the results of the regression analysis of variance for the four models for each module. We note

Table 8. Regression ANOVA for Module DH.

Model	Source	d.f.	S.S.	M.S.	F-value
1	Regression Error Corrected Total	1 11 12	6.8336 5.0129 11.8465	6.8336 0.4557	15
2	Regression Error Corrected Total	3 9 12	7456.5837 820.6471 8277.2308	2485.5279 91.1830	27.26
3	Regression Error Corrected Total	3 11 14	19000.2524 1340.7476 20341.0000	6333.4175 121.8861	51.9628
4	Regression Error Corrected Total	2 12 14	18874.5500 1466.4500 20341.0000	9437.2750 122.2042	77.2255

Table 9. Regression ANOVA for module Ad.

Model	Source	d.f.	S.S.	M.S.	F-value
1	Regression Error Corrected Total	2 35 37	18.0903 20.4280 38.5182	9.0451 0.5837	15.5
2	Regression Error Corrected Total	2 34 37	1858.1094 982.7591 2840.8684	619.3698 28.9046	21.43
3	Regression Error Corrected Total	3 41 44	4961.6805 1962.3195 6653.0000	1653.8935 41.2761	40.07
4	Regression Error Corrected Total	2 42 44	4961.0932 1692.9068 6654.0000	2480.5466 40.3073	61.55

Table 10. Regression ANOVA for module OM.

Model	Source	d.f.	S.S.	M.S.	F-value
1	Regression Error Corrected Total	1 23 24	14.3717 11.6974 26.0691	14.3717 0.5086	28.26
2	Regression Error Corrected Total	3 21 24	7138.8014 1764.5586 8903.3600	2379.6005 84.0266	28.32
3	Regression Error Corrected Total	3 26 29	15123.0485 3942.9515 19066.0000	5041.0162 151.6520	33.24
4	Regression Error Corrected Total	2 27 29	15122.7918 3943.2082 19066.0000	7561.3959 146.0448	51.77

Table 11. Model predictive quality.

Module	Model	R_2	$adjR^2{}_p$	PRESS	C(P)
CP	1	0.5932	0.5813	5500.7660	1.0107
	2	0.7965	0.7774	3111.9200	1.9575
	3	0.7179	0.6976	-	-
	4	0.7099	0.6965	-	-
DH	1	0.5768	0.5384	5984.8639	1.0002
	2	0.9009	0.8678	2364.7900	4.9617
	3	0.9341	0.9161	-	-
	4	0.9279	0.9159	-	-
AD	1	0.4696	0.4394	1978.1602	3.0000
	2	0.6541	0.6241	3610.4381	4.1914
	3	0.7457	0.7271	-	-
	4	0.7456	0.7335	-	-
OM	1	0.5513	0.5318	5674.7230	1.7009
	2	0.8018	0.7735	3458.6369	2.8735
	3	0.7312	0.7693	-	-
	4	0.7932	0.7779	-	-

that the sum of squares of Model 1 in each table is much smaller than the other models since the standardized data is used to develop Model 1. Consequently, so is the mean square of Model 1.

There are four basic models for the prediction of faults which are all significant. In order to identify the best model for each module, we investigate the predictive quality of each model. The statistics associated with the predictive quality are $R^2{}_p$, $adjR^2{}_p$, PRESS and C_p. And they are summarized in Table 11. It is shown from Table 11 that, in regards to $R^2{}_p$ or $adjR^2{}_p$, Model 2 is slightly superior for Module C_p and OM while Model 2 and 4 are mildly better for Module DH and Ad. The predictive quality of models based on PRESS statistic is applied to the comparison of Model 1 and Model 2 because of the difficulty of computation for other models. We note that the value of PRESS statistic of Model 1 is obtained by multiplying the computed value by the variance of the faults since the computed value is based on the standardized data. The comparison of PRESS statistic results in Model 2 is better than Model 1 for Module C_p, DH and OM.

In using the C_p criterion, one seeks to identify regression model for which (i) the C_p value is small or (ii) the C_p value is near to p, where p represents the number of parameters to be estimated by the model. Because there is only one independent variable in the two nonlinear regression models, computation of C_p for these models is not possible. In the sense of C_p, Model 1 is the best model for Module C_p and DH and Model 2 is superior for Module OM. And the value of C_p for Module Ad does not recommend any of two linear models.

Table 12. Parameter estimate and its standard error of selected models.

Module	Model	Parameter	Estimate	std Error
CP	2	Intercept	3.3624	2.7759
		LOC	0.0027	0.0005
		P	−0.1815	0.0618
		C	0.0177	0.0080
DH	2	Intercept	−8.4615	9.0235
		LOC	0.0022	0.0002
		N	10.2292	4.2338
		C	−0.0513	0.0187
	3	b0	−13.7412	18.8113
		b1	0.2693	0.7304
		b2	0.5480	0.2408
	4	a	−3.6919	1.2083
		b	0.7582	0.1190
Ad	3	b0	0.5423	4.6386
		b1	0.0380	0.0979
		b2	0.6164	0.2275
	4	a	−2.9971	0.7742
		b	0.5928	0.0793
OM	2	Intercept	0.9724	8.0573
		LOC	0.0025	0.0003
		DC	−1.4040	0.6510
		N	7.3433	0.33585
	3	b0	0.3960	9.87582
		b1	0.0029	0.0122
		b2	0.9722	0.4020
	4	a	−5.6892	1.4122
		b	0.9576	0.1454

The inspection of the model predictive quality concludes that different models are recommended for different modules as follows: Model 2 would be recommended for Module C_p, Model 2 or nonlinear models are seemed to be the best model for Module DH and OM, and nonlinear models would be suggested for Module Ad.

Table 12 shows estimates of parameters and their corresponding standard errors of recommended models for four modules. The final regression model for Model 2 consists of three metrics which are different for Module CP, DH, and OM. It is noted that LOC is commonly included in the model for all three modules. The frequently used methods for searching the solution of the nonlinear model are Gauss-Newton method, Marquardt algorithm, and DUD method among which the first two methods need partial derivatives of parameters while DUD method does not. In this study, we examine all three methods and obtain similar results. The values in Table 12 are from Gauss-Newton method.

V. CONCLUSIONS

This study is devoted to investigate some aspect of the relationship between software metrics and the faults and to develop a statistical model for the prediction of faults. To this end, we adopt regression method with several software metrics as independent variables and the number of faults as a dependent variable. We consider four regression models which are a stepwise regression model with two factors produced by factor analysis, a stepwise regression model with software metrics, a generalized model of Gaffney's model with LOC, and Poisson regression model with LOC.

Although, the predictive model shows promising results, its accuracy need further improvement. And those recommended models could be changed under the development environments of various software development projects. Finally we have the following conclusive remarks. The comparison of predictive quality shows that nonlinear models are useful for all modules. Hence the nonlinear model should be considered in new project. The software metrics considered are easily obtained at the early stage of the development, this predictive procedure can be used to control the software system development. And owing to the standardization of software development environments such as CASE tools, there are not much difference of the total amount of source codes and messages in similar projects. Therefore, our predictive model can still be useful in any similar new project.

REFERENCES

[1] V. R. Basili and D. H. Hutchens, "An Empirical Study of a Syntatic Complexity Family," *IEEE Tr. on Software Engineering*, Vol. 9, No. 6, 1983, pp. 664–672.

[2] S. Craw, A. McIntosh and D. Pregibon, "An Analysis of Static Metrics and Faults in C Software," *The Journal of Systems and Software*, Vol. 5, 1985, pp. 37–48.

[3] N. Draper and H. Smith, *Applied Regression Analysis*, 2nd Ed., John Wiley & Sons, New York, 1981.

[4] J. E. Gaffney, Jr., "Estimating the Number of Fault in Code," *IEEE Trans. Software Eng.*, Vol. 10, 1984, pp. 459–464.

[5] L. L. Gremillion, "Determinants of Program Repair Maintenance Requirements," *Communications ACM*, Vol. 27, No. 8, 1984, pp. 826–832.

[6] O. Johansson and C. Nord, "Using Predictions Improve Software Reliability," *Ericsson Review*, No. 1, 1995, pp. 30–35.

[7] M. Khoshgoftaar and J. C. Munson, "Predicting Software Development Error Using Software Complexity Metrics," *IEEE Journal of Selected Area in Communication*, Vol. 8, No.5 , 1990, pp. 253–261.

[8] M. R. Lyu, J. Yu, E. Keramidas, and S. Dalal, "ARMOR: Analyzer for Reducing Module Operational Risk," *Proceedings of the 25th International Symposium on Fault Prone Tolerant Computing (FTCS-25)*, Pasadena, Calif., 1995, pp. 137–142.

[9] Munson and T. M. Khoshgoftaar, "The Detection of Fault-Prone Programs," *IEEE Transaction on Software Engineering*, Vol. 18, No.5, 1992, pp. 423–433.

[10] Musa, A. Iannino and K Okumoto, *Software Reliability; Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.

[11] H. Myers, *Classical and Modern Regression with Applications*, Duxbury, Boston, MA, 1986.

[12] J. Neter, W. Wasserman, and M. H. Kutner, *Applied Linear Statistical Models*, IRWIN, Homewood, 1985.

[13] A. A. Porter and R W. Selby, "Empirically Guided Software Development Using Metric-Based Classification Trees," *IEEE Software*, 1990, pp. 46–54.

[14] T. L. Robert Jr., M. L. Gibson, K. T. Fields, and R. K. Rainer, Jr., "Factors that Impact Implementing a System Development Methodology," *IEEE Tr. On Software Engineering*, Vol. 24, No. 8, 1998, pp. 640–649.

[15] V. Rodriguez and W. Tsai, "A Tool for Discriminant Analysis and Classification of Software Metrics," *Information and Software Technology*, Vol. 29, No. 3, 1987, pp. 137–149.

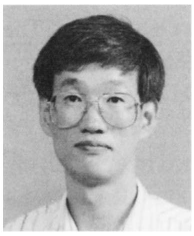
[16] *SAS/Procedure and State Guide for PC*, Ver. 6th Ed., SAS Institute Inc., 1990, pp. 1354–1456.

[17] S. Sawyer and P. J. Guinan, "Software Development: Processes and Performance," *IBM System Journal*, Vol. 37, No. 4, 1988, pp. 552–569.

[18] V. Shen, T. Yu, S. Thebaut, and L. Paulsen, "Identifying Error-Prone Software: An Empirical Study," *IEEE Tr. on Software Engineering*, Vol. SE-11, No. 4, 1985, pp. 317–323.

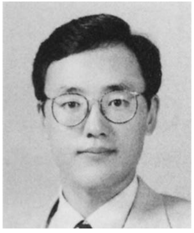
[19] N. Takahashi and Y. Kamayachi, "An Empirical Study of a Model for Program Error Prediction," *Proceedings 8th International Conference on Software Engineering*, London, 1985, pp. 330–336.

[20] X. Zhang and H. Pham, "An Analysis of Factors Affecting Software Reliability," *Jornal of Systems and Software*, to be published.



Young Sik Park is a Senior Member of Engineering Staff in ETRI. He received the B.S. degree in mathematics from Seoul National University in 1983 and the M.S. degree in statistics from Chungnam National University in 1985. He has worked on software integration and testing of telecommunication systems such as TDX1A, TDX-10 ISDN, and ATM switching system since he joined ETRI in 1985. His research interests include configuration management in system development methodology, software reliability and SPICE (Software Process Improvement and Capability dEtermination).

Byeong-Nam Yoon is a Vice President of National Computerization Agency,Korea. He received the B.S. degree in electronics from Han-Yang University in 1975 and the M.S. degree in computer science from Chong-Ju University in 1989. In 1997, he received the Ph.D. degree in computer science from Chungnam National University. He served as project leader of AIN, ATM system development during 1993–1999, ICPS HiTel system project during 1991–1995, and TDX-1A, TDX-1B, TDX-10 Switch System during 1983–1991, respectively. His research interests include high performance network computing architecture and algorithm, software engineering, multi-service switching node and open protocol, intelligent network service architecture and protocol.



Jae-Hak Lim is an Assistant Professor at the Department of Accounting at Taejon National University of Technology, Taejon, Korea. Dr. Lim has taught in the area of statistics, management science and information system. He received the B.S. degree in computer science and statistics and the M.S. degree in statistics from Chungnam National University in 1983 and 1986, respectively. In 1994, he received the Ph.D. degree in statistics from University of Nebraska at Lincoln. During 1994–1997, he worked on evaluating the reliability of ATM switching system at ETRI. His research interests include software reliability, system reliability modeling, statistical inference on life testing and maintenance policy.