Proceedings of the National Conference; INDIACom-2007
Computing for Nation Development, February 23 – 24, 2007
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

On Modelling Software Reliability Growth Phenomena for Errors of Different Severity

P.K. Kapur*[1] Archana Kumar*[2] V.B.Singh[3] F.K. Nailana[4]
*Department of Operational Research, University of Delhi, Delhi, India
*1 pkkapur1@gmail.com *2 arckumar@gmail.com
[1]Delhi College of Arts and Commerce, University of Delhi, Delhi, India
3 singhvb@rediffmail.com
[4]Department of Computer Science, University of Limpopo, South Africa

ABSTRACT
Software testing and debugging are very complex and expensive process. The time to remove a fault depends on the complexity of the detected faults, the skills of the debugging team, the available manpower etc. Therefore, the time delayed by the detection and correction process cannot be ignored as it plays an important role in software development process. It is imperative to clearly understand the software development environment and accordingly there is need to develop a model, which can explicitly explain the software technology that has been used to develop the software. Thus it becomes all the more important that the Software Reliability Growth Models should explicitly take into account errors of different severity. Such an approach can capture the variability in the growth curves depending on the environment it is being used. In this paper we have developed a very general flexible Software Reliability Growth Model accounting for errors of different severity using lag function. The model has been validated on three software datasets and it is shown that the proposed model fairs comparatively better than the existing ones.

KEYWORDS
Software Reliability, Software Reliability Growth Model (SRGM), Severity, Lag Function, Non Homogeneous Poisson Process (NHPP).

INTRODUCTION

Software Reliability Growth Models (SRGMs) have emerged as people try to understand the characteristics of how and why software fails, and try to quantify software reliability. Numerous models have been developed since the early 1970s, but how to quantify software reliability still remains largely unsolved. There is no single model that can be used in all situations. No model is complete or even representative.

Many Software Reliability Growth Models (SRGMs) have been proven to be successful in estimating the software reliability and the number of errors remaining in the software. It has also been observed that the relationship between the testing time and the corresponding number of faults removed is either Exponential or S-Shaped or a mix of the two [2,5,6,7,10,17,22,23]. The software includes different types of faults and each fault requires different strategies and different amounts of testing effort to remove it.

Ohba [18] proposed the Hyper-exponential SRGM, assuming that software consists of different modules. Each module has its characteristics and thus the faults detected in a particular module have their own peculiarities. Therefore, the Fault Removal Rate for each module is not the same. He suggested that the fault removal process for each module be modeled separately and that the total fault removal phenomenon is the addition of the fault removal process of all the modules. Yamada et al. [24] proposed a modified exponential SRGM assuming the software contains two types of faults. Kapur et al [9,10] proposed an SRGM with three types of fault. For each type, the Fault Removal Rate per remaining faults is assumed to be time independent. The first type is modeled by an Exponential model of Goel and Okumoto [6]. The second type is modeled by Delayed S-shaped model of Yamada et al. [22]. The third type is modeled by three stages Erlang model proposed by Kapur et. al [9]. The total removal phenomenon is again modeled by the superposition of the three SRGMs [9]. Later they extended their model to cater for more types of faults [9].

Software testing and debugging are very complex and expensive process. The time to remove a fault depends on the complexity of the detected faults, the skills of the debugging team, the available manpower, etc. Therefore, the time delayed by the detection and correction process cannot be ignored as it plays an important role in software development process. This time dependent function that measures the expected delay in correcting a detected fault at any time is known as lag function. The problem of time dependent function has been addressed by Huang et. al [14] in their paper. Schneidewind [20] proposed an approach to model the fault-correction process by using a constant delayed fault- detection process. He assumed that the rate of fault correction was proportional to the rate of failure detection. However, if this assumption is not met in practice, the model will underestimate the remaining faults in the code. Later, Xie and Zhao [21] pointed out that this assumption was too restrictive. They extended the Schneidewind model to a continuous version by substituting a time-dependent delay function for the constant delay.

In the software reliability growth phase, the software testing process in a sense, determines the nature of the failure data. There are many factors that affect software testing. These factors are unlikely to be kept stable during the entire process of software testing, with the result that the underlying statistics of the failure process is likely to experience major changes. This arises the need for defining the faults detected in such a way that caters faults of different severity. Severity of a failure or fault is the impact it has on the operation of a software-based system. Kapur et. al [11,13] did some work in this area.

In this paper we have proposed software reliability growth model defining errors of different severity using lag and logistic error detection function. The model is based on Non Homogenous Poisson Process (NHPP) and can be used to estimate and predict the reliability of the software products. For the estimation of the parameters of the proposed model, SPSS is used. The goodness-of-fit of the proposed models is compared with Generalised Erlang Model with logistic function [8]. The Generalised Erlang Model with logistic function [8] is combination of exponential model [6], Flexible Delayed s-Shaped model [7] and Erlang K-3 Stage model with logistic function [10]. The new proposed model provides significant improved goodness-of-fit results.

The paper is organized as follows. Section 2 incorporates time dependent delay function into software reliability growth modelling defining errors of different severity. Sections 3, 4 and 5 give the method used for parameter estimation and criteria used for validation and evaluation of the proposed model. We conclude this paper in section 6.

2. SOFTWARE RELIABILITY GROWTH MODELLING

2.1 MODEL ASSUMPTIONS

The proposed model has the following explicit assumptions [9,12,16].

1. The fault detection process follows the NHPP.
2. The software system is subject to failures at random times caused by the manifestation of remaining faults in the system.
3. The fault removal process i.e., the debugging process is perfect.
4. The expected number of faults removed in $(t, t+\Delta t)$ is proportional to the number of faults remaining to be removed.
5. The proportionality is constant over time.
6. Total number of faults is finite.
7. The detected dependent fault may not be immediately removed and it lags the fault detection process by a delay effect factor Δt.
8. No new faults are introduced during the fault removal process.
9. Each time a failure is observed, an immediate (delayed) effort takes place to decide the cause of the failure in order to remove it. The time delay between the failure observation and its subsequent removal is assumed to represent the severity of faults. The more severe the fault, more the time delay.
10. The fault isolation / removal rate with respect to testing effort intensity is proportional to the number of observed failures whose cause are yet to be identified.

2.2 MODEL NOTATIONS:

$m(t)$	Expected number of faults identified in the time interval (0,t] during testing phase
a	Total fault content
a_1, a_2, a_3	Initial fault content
b_1, b_2, b_3	Fault detection rates
$m_1(t), m_2(t), m_3(t)$	Mean number of fault
β	Constant parameter in the Fault Removal Rate function.
Δt	Delay effect factor

2.3 FRAMEWORK FOR MODELLING FOR PROPOSED SRGM

The total removal phenomenon for Generalised Erlang Model with logistic function [8] is modeled by the superposition of the three SRGMs [6,7,9] as given in equation 1.

$$m(t) = a_1(1-e^{-b_1 t}) + a_2 \frac{\left[1-\{1+b_2 t\}e^{-b_2 . t}\right]}{1+\beta\ \bar{e}^{b_2 t}} + a_3 \frac{\left[1-\left\{1+b_3 t+\frac{b_3^2 t^2}{2}\right\}e^{-b_3 . t}\right]}{1+\beta\ \bar{e}^{b_3 t}} \quad (1)$$

Case 1:

Let $m'_1(t) = b(t)[a_1 - m_1(t)]$ (2)

Where $b(t) = b_1$ (3)

Solving the equation (2) under the boundary conditions at $t=0$, $m_1(t)=0$ we get

$$m_1(t) = a_1(1-e^{-b_1 t}) \quad (4)$$

Case 2:

Let $m'_1(t) = b(t)[a_2 - m_1(t)]$ (5)

Here we assume that $m_1(t)$ is the number of fault detected and isolated and we assume that it follows a logistic curve i.e.

$$b(t) = \frac{b_2}{1 + \beta e^{-b_2 t}} \quad (6)$$

which depicts that learning between testing and isolation team grows as the time passes.

Solving the equation (5) under the boundary conditions at $t = 0,\ m_1(t) = 0$ we get

$$m_1(t) = \frac{a_2\left(1 - e^{-b_2 t}\right)}{1 + \beta e^{-b_2 t}} \quad (7)$$

Let $m_2(t) = m_1(t - \Delta t)$ (8)

Where $\Delta t = \frac{1}{b_2}\log(1 + b_2 t)$ (9)

Substituting value of Δt in (8) and solving under the boundary conditions at $t = 0,\ m_2(t) = 0$, we get

$$m_2(t) = a_2 \frac{\left[1 - (1 + b_2 t)e^{-b_2 . t}\right]}{1 + (1 + b_2 t)\beta\ \bar{e}^{b_2 t}} \quad (10)$$

Case 3:

Let $m'_1(t) = b(t)\left[a_3 - m_1(t)\right]$ (11)

Where $b(t) = \frac{b_3}{1 + \beta e^{-b_3 t}}$ (12)

Solving the equation (11) under the boundary conditions at $t = 0,\ m_1(t) = 0$ we get

$$m_1(t) = \frac{a_3\left(1 - e^{-b_3 t}\right)}{1 + \beta e^{-b_3 t}} \quad (13)$$

Let $m_2(t) = m_1(t - \Delta t)$ (14)

Where $\Delta t = \frac{1}{b_3}\log(1 + b_3 t)$ (15)

Substituting value of Δt in (14) and solving under the boundary conditions at $t = 0,\ m_2(t) = 0$, we get

$$m_2(t) = a_3 \frac{\left[1 - (1 + b_3 t)e^{-b_3 . t}\right]}{1 + (1 + b_3 t)\beta\ \bar{e}^{b_3 t}} \quad (16)$$

Let $m_3(t) = m_2(t - \Delta t)$ (17)

Where $\Delta t = \frac{1}{b_3}\log\left(1 + b_3 t + \frac{b_3{}^2 t^2}{2}\right)$ (18)

Substituting value of Δt in (17) and solving under the boundary conditions at $t = 0,\ m_3(t) = 0$, we get

$$m_3(t) = a_3 \frac{\left[1 - \left(1 + b_3 t + \frac{b_3^2 t^2}{2}\right)e^{-b_3 . t}\right]}{1 + \left(1 + b_3 t + \frac{b_3^2 t^2}{2}\right)\beta\ \bar{e}^{b_3 t}} \quad (19)$$

2.4 MODELLING TOTAL FAULT REMOVAL PHENOMENON

The proposed model is the sum of case1, case2 and case3. (4), (10) and (19) are mean value functions of respective NHPPs [9]. Thus, the mean value function of superposed NHPP is:

PROPOSED GENERALISED ERLANG MODEL WITH LOGISTIC ERROR DETECTION AND LAG FUNCTION

$$m(t) = m_1(t) + m_2(t) + m_3(t) \quad (20)$$

$$\begin{aligned} m(t) &= a_1\left(1 - e^{-b_1 t}\right) \\ &+ a_2 \frac{\left[1 - (1 + b_2 t)e^{-b_2 . t}\right]}{1 + (1 + b_2 t)\beta\ \bar{e}^{b_2 t}} \\ &+ a_3 \frac{\left[1 - \left(1 + b_3 t + \frac{b_3^2 t^2}{2}\right)e^{-b_3 . t}\right]}{1 + \left(1 + b_3 t + \frac{b_3^2 t^2}{2}\right)\beta\ \bar{e}^{b_3 t}} \end{aligned} \quad (21)$$

where $a_1 + a_2 + a_3 = a$

3. PARAMETER ESTIMATION

The success of mathematical modeling approach to reliability evaluation depends heavily upon quality of failure data collected. The parameters of the SRGMs are estimated based upon these data. Hence efforts should be made to make the data collection more explicit and scientific. Usually data is collected in one of the following two ways. In the first case the times between successive failures are recorded. The other easier and commonly collected data type is known as the grouped data. Here testing intervals are specified and number of failures experienced during each such interval is noted. The proposed model is non-linear and presents extra problems in estimating the parameters. Technically, it is more difficult to find the solution for non-linear models using Least Square method and requires numerical algorithms to solve it. Statistical software packages such as SPSS helps to overcome this problem. SPSS is a Statistical Package for Social Sciences. It is a comprehensive and flexible statistical analysis and data management system. SPSS can take data from almost any type of file and use them to generate tabulated reports, charts, and plots of distributions and trends, descriptive statistics, and conduct complex statistical analysis. SPSS Regression Models enables the user to apply more sophisticated models to the data using its wide range of nonlinear regression models. For the estimation of the parameters of the proposed model, Method of Least Square (Non Linear Regression method) has been used.

4. COMPARISON CRITERIA FOR SRGMS

The performance of SRGMs are judged by their ability to fit the past software fault data (goodness of fit).

4.1 GOODNESS OF FIT CRITERIA

The term goodness of fit is used in two different contexts. In one context, it denotes the question if a sample of data came from a population with a specific distribution. In another context, it denotes the question of "How good does a mathematical model (for example a linear regression model) fit to the data"?

A. THE MEAN SQUARE FITTING ERROR (MSE):

The model under comparison is used to simulate the fault data, the difference between the expected values, $\hat{m}(t_i)$ and the observed data y_i is measured by MSE [9] as follows.

$$MSE = \sum_{i=1}^{k} \frac{(\hat{m}(t_i) - y_i)^2}{k} \qquad (22)$$

where k is the number of observations. The lower MSE indicates less fitting error, thus better goodness of fit.

B. THE AKAIKE INFORMATION CRITERION (AIC):

The criteria is defined as *AIC* = -2(the value of the maximum log likelihood function) + 2(the number of the parameters used in the model).This index [1,9] takes into account both the statistical goodness of fit and the number of parameters that are estimated in competing models. Lower values of AIC indicate the preferred model.

C. COEFFICIENT OF MULTIPLE DETERMINATION (R^2):

We define this coefficient as the ratio of the sum of squares resulting from the trend model to that from constant model subtracted from 1[9].

i.e. $$R^2 = 1 - \frac{\text{residual SS}}{\text{corrected SS}}. \qquad (23)$$

R^2 measures the percentage of the total variation about the mean accounted for the fitted curve. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well. The larger R^2, the better the model explains the variation in the data.

D. PREDICTION ERROR (PE):

The difference between the observation and prediction of number of failures at any instant of time i is known as PE_i. Lower the value of Prediction Error better is the goodness of fit [19].

E. VARIATION:

The standard deviation of PE is known as variation.

$$Variation = \sqrt{\left(\tfrac{1}{N-1}\right)\sum (PE_i - Bias)^2} \qquad (24)$$

Lower the value of Variation better is the goodness of fit [19].

F. ROOT MEAN SQUARE PREDICTION ERROR:

It is a measure of closeness with which a model predicts the observation.

$$RMSPE = \sqrt{(Bias^2 + Variation^2)} \qquad (25)$$

Lower the value of Root Mean Square Prediction Error better is the goodness of fit [19].

G. THE KOLMOGOROV-SMIRNOV TEST

The Kolmogorov-Smirnov test (K-S test) [4] is a non-parametric test. It tries to determine if two datasets differ significantly. The Kolmogorov-Smirnov (K-S) test is based on the empirical distribution function (ECDF). Since it is non-parametric, it treats individual observations directly and is applicable even in the case of very small sample size, which is usually the case with SRGM validation.

Given *N ordered* data points $\boldsymbol{Y_1, Y_2, ..., Y_N}$, the ECDF is defined as $E_N = n(i)/N$ (26)

where $\boldsymbol{n(i)}$ is the number of points less than $\boldsymbol{Y_i}$ and the $\boldsymbol{Y_i}$ are ordered from smallest to largest value. This is a step function that increases by $1/N$ at the value of each ordered data point. The Kolmogorov-Smirnov test statistic is defined as

$$D = \max_{1 \le i \le N}\left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i)\right) \qquad (27)$$

where $\boldsymbol{F}$ is the theoretical cumulative distribution of the distribution being tested which must be a continuous distribution. Lower the value of Kolmogorov-Smirnov test better is the goodness of fit.

5. MODEL VALIDATION AND DATA DESCRIPTION

To check the validity of the proposed model and to find out its software reliability growth, it has been tested on three Data Sets. The Proposed Model i.e. Generalised Erlang Model with logistic and lag function has been compared with Generalised Erlang Model with logistic function [8]. The Proposed Model provides better goodness of fit for all the datasets its applicability and flexibility. However, the increased accuracy achieved shows the capability of the model to capture different types of failure datasets e.g. Exponential, s-Shaped.

DS-1

This data is cited from Misra [15]. The software was tested for 38 weeks during which 2456.4 computer hours were used and 231 faults were removed. The Parameter Estimation result and the goodness of fit results for the proposed model are given in Table 1. The goodness of fit curve for DS-1 is given in Figure 1. The K-S statistic result for DS-1 is given in Table 3.

DS-2

This data is cited from M.Ohba [18]. The software was tested for 19 weeks during which 47.65 computer hours were used and 328 faults were removed. The Parameter Estimation result and the goodness of fit result for the proposed model are given in Table 2. The goodness of fit curve for DS-2 is given in Figure 2. The K-S statistic result for DS-2 is given in Table 3.

TABLE 1: MISRA 231 FAULTS

Parameter Estimation and Comparison Criteria	Models under Comparison	
	Proposed Model	Generalized Erlang Model with Logistic Function[8]
a	297	309
b_1	.23	.13
b_2	.32	.23
b_3	.17	.10
β	23.46	36.67
R^2	.99815	.99739
MSE	6.86	9.70
AIC	203.04	204.25
Bias	.06	-0.10
Variation	2.653	3.155
RMSPE	2.654	3.157

TABLE 2: OHBA 328 FAULTS

Parameter Estimation and Comparison Criteria	Models under Comparison	
	Proposed Model	Generalized Erlang Model with Logistic Function[8]
a	338	354
b_1	.31	.32
b_2	.56	.33
b_3	.50	.29
β	8.98	11.31
R^2	.99532	.99460
MSE	48.26	55.70
AIC	201.5	209.8
Bias	.26	.51
Variation	7.132	7.649
RMSPE	7.137	7.667

TABLE 3: K-S STATISTIC

Datasets	Models under Comparison	
	Proposed Model	Generalized Erlang Model with Logistic Function[8]
DS-1	.975	.980
DS-2	.523	.646

GOODNESS OF FIT CURVES FOR DS-1AND DS-2

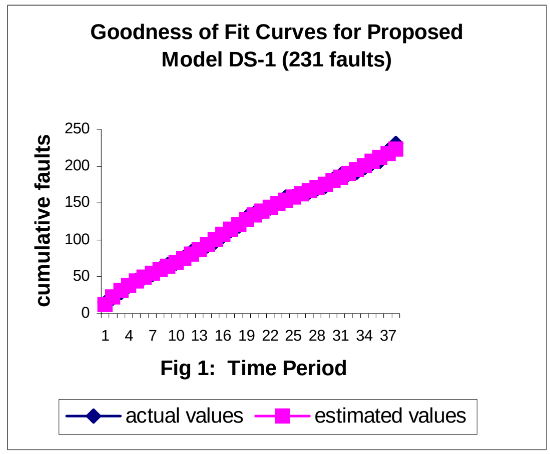

Figure 1

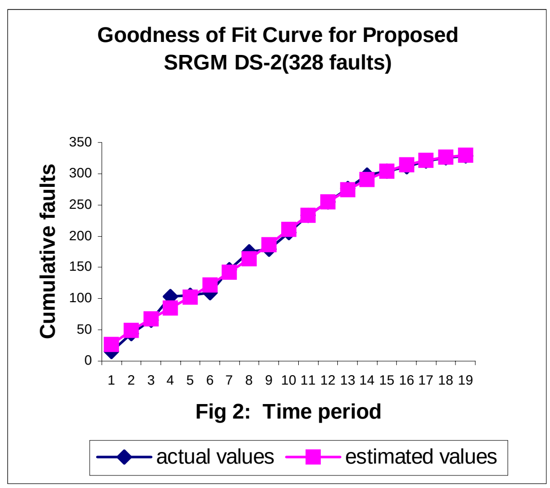

Figure 2

The Parameter estimated for the Proposed Generalised Erlang Model using lag and logistic function are better than those of Generalised Erlang Model with logistic function [8] as shown in Tables 1-2. The Proposed Generalised Erlang Model using lag and logistic function gives better goodness of fit as shown in Figures 1-2 for Datasets 1-2 than Generalised Erlang Model with logistic function [8] as shown by the MSE, R^2, AIC, Bias, Variation, RMSPE, K-S Test in the above tables.

The values of fault detection rate b_2, b_3 are higher than those of b_1 as depicted in Tables 1-2 because the testing team have to spend more time to analyze the cause of the failure and therefore requires greater efforts to remove them as the faults in the software development component can be different severity.

6. CONCLUSION

In this paper we propose a software reliability growth model defining errors of different severity using lag function. We tried to incorporate both failure dependency and time dependent delay function into software reliability growth modelling. The time delayed by the detection and correction process can't be ignored as it plays an important role in software development process. The fault-correction process can be modeled as a delayed fault detection process and it lags the detection process by a time dependent delay. Thus the proposed delay effect factor can be used to measure the expected time lag in correcting the detected faults during software development. Estimation results show that the proposed framework to incorporate both failure dependency and time-dependent delay function for SRGM has a fairly accurate prediction capability.

FUTURE SCOPE

Concept of change point using testing effort function can be incorporated in the proposed model for further research.

REFERENCES

[1]. Akaike H.(1974), "A New Look at Statistical Model Identification"-IEEE Transactions on Automatic Control; 19: pp. 716-723.

[2]. Bittanti S, Bolzern P, Pedrotti E, Scattolini R (1988). "A Flexible Modelling Approach for Software Reliability Growth" Software Reliability Modelling and Identification (Ed.) G. Goos And J. Harmanis, Springer Verlag, Berlin, pp 101-140

[3]. Brooks WD and Motley RW (1980), "Analysis of Discrete Software Reliability Models-Technical Report" (RADC-TR-80-84) New York: Room Air Development Center.

[4]. Chakravart; Laha; Roy (1967) "Handbook of Methods of Applied Statistics", Volume I, John Wiley and Sons, pp. 392-394.

[5]. Downs T and Scott A. (1992); "Evaluating the Performance of Software Reliability Models", IEEE transactions on Reliability 41(4): pp 532-538.

[6]. Goel AL and Okumoto K. (1979); "Time Dependent Error Detection Rate Model For Software Reliability and Other Performance Measure"-IEEE Transactions on Reliability; R-28 (3): pp 206-211.

[7]. Kapur P.K., Goswami D.N., Bardhan A.K.;(2005), "A Flexible Delayed S-Shaped Software Reliability Growth Model," Journal of Computer Society of India Vol 35, No. 4, pp 10-16.

[8]. Kapur P.K., Gupta A, Kumar A., Yamada S (2005) "Flexible Software Reliability Growth Models for Distributed Systems" published in OPSEARCH, vol.42 no.4 pp 378-398.

[9]. Kapur P.K., R.B. Garg and S. Kumar; (1999), "Contributions to Hardware and Software Reliability", World Scientific, Singapore.

[10]. Kapur P.K., Younes S and Agarwala S.; (1995), "Generalised Erlang Model with n types of faults", ASOR Bulletin, 14: pp 5-11.

[11]. Kapur P.K., Younes S and Grover P.S.; (1995), "Software Reliability Growth Model with Errors of Different Severity". Computer Science and Informatics (India) 25(3): pp 51-65.

[12]. Kapur P.K., Younes S., (1995) "Software Reliability Growth Model with Error Dependency". Microelectronics and Reliability 35, pp. 273-278.

[13]. Kapur PK., and Garg RB; (2002), "Why Software Reliability Growth Modelling should define errors of different Severity", Journal of Indian Statistical Association Vol 40, 2, 2002, pp 119-142.

[14]. Lo J.H., Kuo S. Y., Lyu M.R., and Huang C. Y. (2006), "Modelling Fault Detection and Correction Processes in Software Reliability Analysis," IEEE Trans. on Reliability, in Revision.

[15]. Misra PN (1983), "Software Reliability Analysis", IBM System Journal 22(3) pp 262-270.

[16]. Musa (1999), "Software Reliability Engineering: More Reliable Software, Faster Development and Testing", McGraw-Hill.

[17]. Ohba M. (1984), "Infection S-Shaped Software Reliability Growth Model" Stochastic Models in Reliability Theory", Springer- Verlag, Berlin, pp. 144-162,.

[18]. Ohba, M.; (1984), "Software Reliability Analysis Models", IBM Journal of research and Development 28, 428-443.

[19]. Pillai K. and V.S.S. Nair; (1997) "A Model for Software Development Effort and Cost Estimation", IEEE Transactions on Software Engineering; vol. 23(8), pp. 485-497.

[20]. Schneidewind (2001), "Modelling the Fault Correction Process," Proceedings of the 12th IEEE International Symposium on Software Reliability Engineering, pp. 185-190, Hong Kong, China.

[21]. Xie and Zhao (1992), "The Schneidewind Software Reliability Model Revisited," Proceedings of the 3rd IEEE International Symposium on Software Reliability Engineering, pp. 184-192, Research Triangle Park, North Carolina.

[22]. Yamada, Ohba, and Osaki (1983), "S-Shaped Reliability Growth Modelling for Software Error Detection," IEEE Trans. Reliability, Vol. R-32, No. 5, pp. 475-484.

[23]. Yamada, Ohba, and Osaki; (1984), S-Shaped Software Reliability Growth Models and their Applications, IEEE Transactions on Reliability, 1984; R-33: pp169-175.

[24]. Yamada S, Osaki S. and Narihisa H.;(1985), Software Reliability Growth Models with Two Types

of Errors-Recherche Operationnelle /Operations Research (RAIRO) 19:87-104.