Testing-Coverage Dependent Software Reliability Growth Modeling and Its Applications

Shinji INOUE† and Shigeru YAMADA†

†Department of Social Systems Engineering,
Faculty of Engineering, Tottori University,
4-101 Minami, Koyama-cho, Tottori-shi, Tottori 680-8552, Japan
Phone: +81 857-31-5303
Fax: +81 857-31-0882
E-mail: {ino, yamada}@sse.tottori-u.ac.jp

Abstract– *We discuss software reliability growth modeling considering with testing-coverage. The testing-coverage is one of the important metrics related to the software reliability growth process. First we develop an alternative testing-coverage function to describe time-dependent behavior of testing-coverage maturity process. Then, we propose a software reliability growth model by formulating the relationship between the testing-coverage maturity process and the software reliability growth process. And we derive several software reliability assessment measures which are useful metrics for quantitative assessment of software reliability. Finally, we show numerical examples of software reliability analysis based on the proposed model by using an actual data set.*

Keywords: Software reliability assessment; Testing-coverage; Alternative testing-coverage function; Discretized parameter estimation method; Nonhomogeneous Poisson process model.

1. Introduction

Software reliability assessment is one of the important issues to produce reliable software systems. A software reliability growth model (abbreviated as SRGM) [1–4] has been utilized as one of the fundamental techniques for quantitative assessment of software reliability. The SRGM describes the software fault-detection phenomenon or the software failure-occurrence phenomenon by applying stochastic and statistical theories. Especially, it is well-known that a nonhomogeneous Poisson process model (abbreviated as NHPP model) can characterize the software reliability growth process simply by supposing a suitable mean value function of an NHPP. Accordingly, the NHPP model has been utilized for software reliability assessment in many software houses and computer manufacturers from high applicability and simplicity of the model structure of an NHPP point of view [5]. Up to now, several specific NHPP models have been also proposed such as imperfect debugging SRGM's [6, 7], testing-domain dependent SRGM's [8, 9], and so forth. The testing-domain dependent SRGM's have been derived by considering the time-dependent behavior of a testing-domain coverage which is a factor related to the software reliability growth process.

In this paper we focus on a testing-coverage as a key factor related to the software reliability growth process. The testing-coverage is one of the important measures to evaluate the quality of testing and tested software products. There are several researches on the relationship between the testing-coverage and the software reliability. Specifically, Fujiwara and Yamada [9] and Malaiya et al. [10] have proposed a software reliability growth model with the testing-coverage, respectively, and Pham and Zhang [11] also proposed an NHPP model and software cost models with the testing-coverage. However, our approach is different from the researches above in terms of the relationship between the testing-coverage and the software reliability growth process. First we propose an alternative testing-coverage function to describe time-dependent behavior of a testing-coverage maturity process. Then, we formulate the relationship between the attained testing-coverage and the number of detected faults. Estimation methods for unknown parameters of the alternative testing-coverage function and our SRGM are also discussed, respectively. Finally, we derive several software reliability assessment measures which are useful metrics for quantitative software reliability assessment, and show numerical examples of software reliability analysis based on the proposed model by using an actual data set.

2. Testing-Coverage

Testing-coverage is one of the important measures to evaluate the quality of testing and tested software products. The typical testing-coverage measures are classified into several types in terms of control flow testing as follows: statement coverage, branch coverage, path coverage. For example, the statement coverage is measured on the basis of the statement-paths that have been executed at least once by the test-cases. This is called C0 testing-coverage measure.

Most SRGM's are ordinally developed by characterizing the relationship between the testing-time and the number of detected software faults. Accordingly, we need to characterize the relationship between the testing-time and the testing-coverage to develop an SRGM considering with testing-coverage maturity process first. In this section we discuss basic concepts to describe the time-dependent behavior of the testing-coverage maturity process.

2.1. Formulation

First we propose a basic equation to describe the time-dependent behavior of the testing-coverage maturity process, which is called as an alternative testing-coverage function in this paper. For developing the function, we assume that the testing-coverage maturity rate at any time is proportional to the difference between the target value and the current one of testing-coverage. Letting $C(t)$ be the ratio of testing-coverage attained by arbitrary testing time t, we can derive the following differential equation from the assumption:

$$\frac{dC(t)}{dt} = \beta(t)[\alpha - C(t)] \quad (\alpha > 0,\ \beta(t) > 0), \tag{1}$$

where α indicates the target value of testing-coverage to be attained, and $\beta(t)$ the testing-coverage maturity ratio at arbitrary testing time t. We can easily obtain the alternative testing-coverage function by solving the above differential equation with respect to $C(t)$.

2.2. Formulation with testing-skill

The testing-coverage discussed above helps software development managers to evaluate whether the test-cases have been designed to detect faults effectively. Accordingly, the time-dependent behavior of the testing-coverage depends on a testing-skill of test-case designers. In this paper we assume that the testing-skill of test-case designers increases as the ratio of testing progress goes on. Supposing that the testing-skill factor for the test-case designers is given as

$$r = \frac{b_{ini}}{b_{sta}}, \tag{2}$$

we extend $\beta(t)$ in Eq.(1) as follows:

$$\beta(t) \equiv B(C(t)) = b_{sta}\{r + (1-r)\frac{C(t)}{\alpha}\}, \tag{3}$$

where b_{ini} represents the initial testing-skill factor of the test-case designers, b_{sta} the steady-state one, and r the inflection coefficient. Substituting Eq.(3) into Eq.(1), we can obtain the following equation by solving the differential equation in Eq.(1):

$$C(t) = \frac{\alpha(1 - e^{-b_{sta}\cdot t})}{1 + z \cdot e^{-b_{sta}\cdot t}}, \tag{4}$$

where $z = (1-r)/r$. We call Eq.(4) an alternative testing-coverage function with testing-skill in this paper. The inflection point of the alternative testing-coverage function in Eq.(4) is derived as

$$t^* = \frac{\log z}{b_{sta}}. \tag{5}$$

Then, we have

$$C(t^*) = \frac{\alpha}{2}\left(1 - \frac{1}{z}\right). \tag{6}$$

In Eq.(4), $C(t)$ indicates an exponential growth curve if $r = 1$ for the case that the internal program structure is simple, and indicates an S-shaped one called a logistic curve if $r \to 0$ for the case that the internal program structure is complex and the testing-effort increases more and more as the testing time goes on.

3. Software Reliability Modeling

3.1. NHPP model

Time-dependent behavior of a software fault-detection process or a failure-occurrence phenomenon, i.e., a software reliability growth process, has been formulated by using a counting process ordinarily. An NHPP which is one of the counting processes is widely used for the software reliability growth modeling. A counting process $\{N(t),\ t \geq 0\}$ is said to be an NHPP with mean value function $H(t)$ if $N(t)$ obeys the following distribution:

$$\begin{cases} \Pr\{N(t) = n\} = \dfrac{\{H(t)\}^n}{n!}\exp\{-H(t)\} \\ \qquad (n = 0, 1, 2, \cdots), \\ H(t) = \displaystyle\int_0^t h(\tau)d\tau, \end{cases} \tag{7}$$

where $h(\tau)$ is the intensity function representing the instantaneous fault-detection rate. The time-dependent behavior of the fault-detection process is characterized by the mean value function $H(t)$ which means the expected number of faults detected in the time-interval $(0,\ t]$. In this paper we assume that the expected number of faults detected at testing time t is proportional to the expected current fault content. Accordingly, we can obtain the following differential equation:

$$\frac{dH(t)}{dt} = b(t)[a - H(t)], \tag{8}$$

where a represents the expected initial fault content in the software system, and $b(t)$ the fault-detection rate per fault at testing time t. The mean value function can be derived as follows by solving the differential equation in Eq.(8):

$$H(t) = a\left(1 - \exp[-\int_0^t b(\tau)d\tau]\right). \quad (9)$$

Most NHPP models can be characterized by $b(t)$ in Eq.(8) or Eq.(9), i.e., Eq.(9) is so-called a Goel-Okumoto SRGM [13] when $b(t) \equiv b$.

3.2. Modeling with testing-coverage

In this section we propose a software reliability growth model considering with the testing-coverage maturity process based on the NHPP. First we formulate the relationship between the testing-coverage muturity process and the expected number of detected faults.

Supposing that the expected number of faults detected at testing time t is proportional to the expected current fault content and the attained testing-coverage at testing time t, we can formulate the relationship as the following equation by using Eq.(4):

$$\frac{dH_C(t)}{dt} \bigg/ \frac{dC(t)}{dt} = s[a - H_C(t)], \quad (10)$$

where s is the fault-detection rate per attained testing-coverage and per fault. That is, $b(t)$ in Eq.(8) or (9) is given as $b(t) \equiv b_C(t) = s \cdot c(t)$ in which $c(t) \equiv dC(t)/dt$. Then, we can obtain the following solution by solving the differential equation in Eq.(10) with respect to $H_C(t)$:

$$H_C(t) = a\left[1 - \exp\{-s \cdot C(t)\}\right]. \quad (11)$$

We define the NHPP model with the mean value function in Eq.(11) as an SRGM with the testing-coverage. It is noted that the mean value function with the testing-coverage in Eq.(11) has the following property:

$$\lim_{t\to\infty} H_C(t) = a\left(1 - \exp[-s \cdot \alpha]\right), \quad (12)$$

which implies that $H_C(t)$ in Eq.(11) does not converge on the initial fault content a in the software system even if $t \to \infty$. Therefore, $\{a - H_C(\infty)\}$ represents the expected total fault content to be detected on the other testing-coverage factors.

4. Parameter Estimation

We discuss methods of parameter estimation for the alternative testing-coverage function in Eq.(4) and the SRGM in Eq.(11), respectively. We suppose that K data pairs $(t_k,\ x_k,\ y_k)(k = 0, 1, 2, \cdots, K)$ with respect to the total number of detected faults, y_k, and the total attained testing-coverage, x_k, during the time-interval $(0,\ t_k]$ are observed.

We first discuss an estimation method for the alternative testing-coverage function in Eq.(4). The method of least-squares is applied to Eq.(1) transformed into an integrable difference equation. Concretely speaking, first we derive the following integrable difference equation via using Hirota's bilinearization methods [14] from the differential equation in Eq.(1) with $\beta(t)$ in Eq.(3):

$$C_{n+1} - C_n = \delta r\alpha b_{sta} + \frac{\delta b_{sta}(1-2r)}{2} \cdot [C_n + C_{n+1}] - \frac{\delta b_{sta}(1-r)}{\alpha} C_n C_{n+1}. \quad (13)$$

Solving the above difference equation yields an exact solution of C_n representing the testing-coverage attained by nth testing-period as

$$C_n = \frac{\alpha\left[1 - \left(\frac{1-\frac{1}{2}\delta b_{sta}}{1+\frac{1}{2}\delta b_{sta}}\right)^n\right]}{1 + z\left(\frac{1-\frac{1}{2}\delta b_{sta}}{1+\frac{1}{2}\delta b_{sta}}\right)^n} \quad (z > 0,\ 0 \le r \le 1), \quad (14)$$

where δ represents the constant time-interval, that is, $t = n\delta$. We should note that Eq.(13) conserves the characteristics of the differential equation in Eq.(1) with $\beta(t)$ in Eq.(3). That is, the difference equation in Eq.(13) has an exact solution, and, as $\delta \to 0$, Eqs.(13) and (14) converge on the original differential equation in Eq.(1) with $\beta(t)$ in Eq.(3) and the exact solution in Eq.(4) of the differential equation, respectively. These properties above are features of a integrable difference equation derived by using the Hirota's bilinearization methods. From Eq.(13), a regression equation to get parameter estimates can be derived as the following:

$$Y_n = A + B_1 K_n + B_2 L_n, \quad (15)$$

where

$$\begin{cases} Y_n &= C_{n+1} - C_n \\ K_n &= C_n + C_{n+1} \\ L_n &= C_n C_{n+1} \\ A &= \delta\alpha r b_{sta} \\ B_1 &= \delta b_{sta}(1-2r)/2 \\ B_2 &= -\delta b_{sta}(1-r)/\alpha. \end{cases} \quad (16)$$

Using Eq.(15), we can estimate $\widehat{A}$, $\widehat{B_1}$, and $\widehat{B_2}$ by using the observed testing-coverage data, which are the estimates of A, B_1, and B_2, respectively. Therefore, we can obtain the parameter estimates $\widehat{\alpha}$, $\widehat{b_{sta}}$, and $\widehat{r}$ from Eq.(16) as follows:

$$\begin{cases} \widehat{\alpha} &= \widehat{A}\Big/\left(\sqrt{\widehat{B_1}^2 - \widehat{A}\widehat{B_2}} - \widehat{B_1}\right) \\ \widehat{b_{sta}} &= 2\sqrt{\widehat{B_1}^2 - \widehat{A}\widehat{B_2}}\Big/\delta \\ \widehat{r} &= \left(1 - \widehat{B_1}/\sqrt{\widehat{B_1}^2 - \widehat{A}\widehat{B_2}}\right)\Big/2. \end{cases} \quad (17)$$

Y_n, K_n, and L_n in Eq.(15) are independent of δ because δ is not used in calculating Y_n, K_n, and L_n in Eq.(15). Hence, we can obtain the same parameter estimates $\widehat{\alpha}$, $\widehat{b_{sta}}$, and $\widehat{r}$, respectively, when we choose any value of δ. It is said that this method can get more accurate parameter estimates than the ordinary method of least-squares [15].

Second we discuss an estimation method for the mean value function in Eq.(11). We use the method of maximum-likelihood to get the parameter estimates, $\widehat{a}$ and $\widehat{s}$, in the mean value function. Then, the logarithmic likelihood function is given as

$$\ln L = \sum_{k=1}^{K}(y_k - y_{k-1}) \cdot \ln[H_C(t_k) - H_C(t_{k-1})] - H_C(t_K) - \sum_{k=1}^{K}\ln[y_k - y_{k-1}], \quad (18)$$

from the properties of the NHPP. Furthermore, we can derive the following simultaneous equations by partially differentiating the logarithmic likelihood function $\ln L$ with respect to parameters a and s:

$$\frac{\partial \ln L}{\partial a} = \frac{\partial \ln L}{\partial s} = 0. \quad (19)$$

By solving the above simultaneous equations numerically, we can estimate $\widehat{a}$ and $\widehat{s}$ which are the estimates of a and s, respectively.

5. Assessment Measures

In this section we derive several software reliability assessment measures which are useful for quantitative assessment of software reliability and the progress of the software testing process. Specifically, we derive the software reliability function, the instantaneous and cumulative MTBF's (MTBF : mean time between software failures).

5.1. Software reliability function

Given that the testing or the user operation has been going up to time t, the probability that a software failure does not occur in the time-interval $(t,\ t+x](x \geq 0,\ t \geq 0)$ is derived as

$$R(x \mid t) = \exp[-\{H(t+x) - H(t)\}], \quad (20)$$

from Eq.(7). $R(x \mid t)$ in Eq.(20) is called a software reliability function. We can estimate the software reliability by using this equation.

5.2. Instantaneous MTBF

We discuss the instantaneous MTBF which has been used as one of the substitution for MTBF. An instantaneous MTBF is approximately given by

$$\mathrm{MTBF}_I(t) = \frac{1}{h(t)}. \quad (21)$$

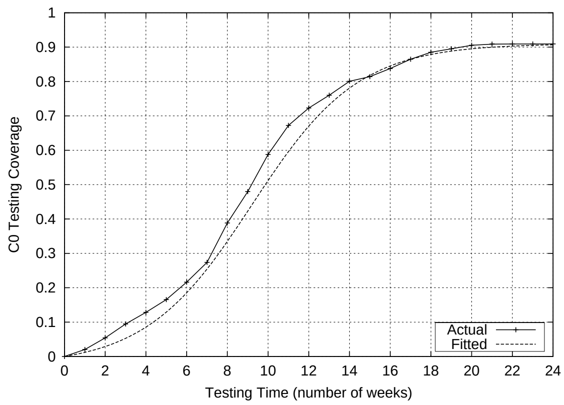

Fig. 1 : The estimated alternative testing-coverage function (on the C0 testing-coverage measure).

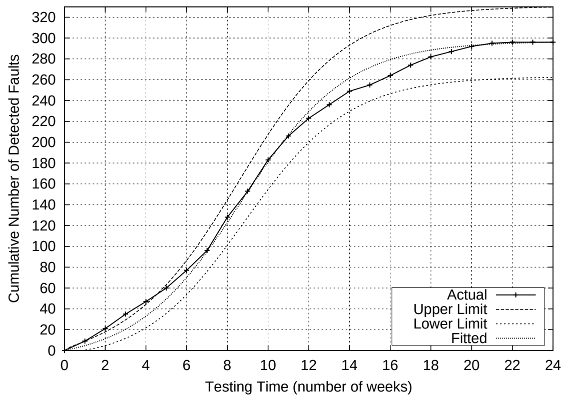

Fig. 2 : The estimated mean value function with its 95% confidence limits.

5.3. Cumulative MTBF

The cumulative MTBF is also the substitution for the MTBF. The cumulative MTBF is approximately derived as

$$MTBF_C(t) = \frac{t}{H(t)}. \quad (22)$$

If the instantaneous MTBF in Eq.(21) and the cumulative MTBF in Eq.(22) take on a large value, respectively, then we decide that the software system becomes more reliable.

6. Numerical Examples

For evaluating the performance of our model, we show numerical examples by using C0 testing-coverage measure data recorded along with fault count data collected from an actual testing of an embedded software system. There are totally 296 faults detected and 90.6% of the C0 testing-coverage measure attained within 24 weeks.

Figure 1 shows the estimated alternative testing-coverage function $\widehat{C}(t)$ of Eq.(4) in which

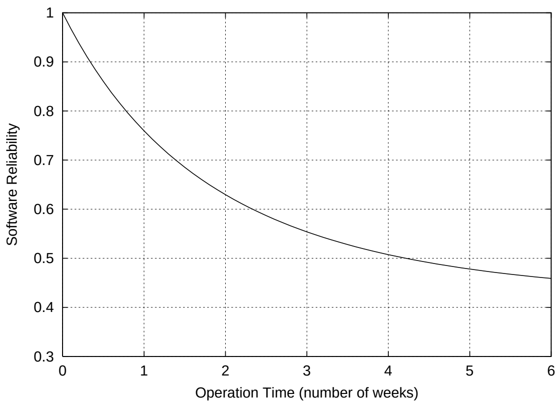

Fig. 3： The estimated software reliability.

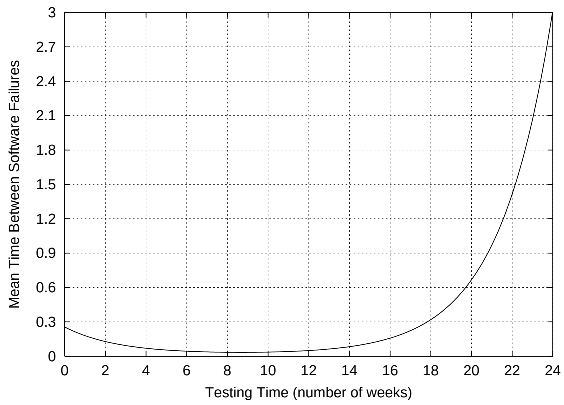

Fig. 4： The estimated instantaneous MTBF.

the parameter estimates are obtained as $\widehat{\alpha} = 90.796$, $\widehat{b_{sta}} = 0.388$, and $\widehat{z} = 52.338$. From Figure 1, we can see that the estimated alternative testing-coverage function fits well to the actual C0 testing-coverage measure data. Next, Figure 2 shows the estimated mean value function $\widehat{H_C}(t)$ of Eq.(11) and its 95% confidence limits, where the parameter estimates of $\widehat{H_C}(t)$ are obtained as $\widehat{a} = 919.9$ and $\widehat{s} = 0.4282$. The $100\gamma\%$ confidence limits for $\widehat{H_C}(t)$ are derived as

$$\widehat{H_C}(t) \pm K_\gamma \sqrt{\widehat{H_C}(t)}, \tag{23}$$

where K_γ indicates the $100(1+\gamma)/2$ percent point of the standard normal distribution. And then, we conduct the Kolmogorov-Smirnov (abbreviated as K-S) goodness-of-fit test [2, 3] to evaluate whether $\widehat{H_C}(t)$ fits statistically to the observed data. This statistical testing is considered to be efficient even if the sample size of data set is small [2]. We can verify that $\widehat{H_C}(t)$ fits to the observed data with the 5% level of significance from the result of the K-S goodness-of-fit testing.

Furthermore, Figure 3 depicts the estimated software reliability function at the termination of the testing in Eq.(20). If we assume that the developed software system is used in the operation phase which has the same environment as the testing phase, we can estimate the software reliability after 3 weeks from the termination time of the testing, $R(3 \mid 24)$, to be about $0 \cdot 554$. And Figure 4 shows the estimated instantaneous MTBF. We can estimate the instantaneous MTBF at the termination time of the testing, $\widehat{\mathrm{MTBF}}_I(24)$, to be about $3 \cdot 024$ (weeks).

7. Concluding Remarks

We have discussed software reliability growth modeling based on the testing-coverage which is one of the key factors related to the software reliability growth process in this paper. Specifically, the relationship between the number of detected faults and the attained testing-coverage has been discussed. And we have also discussed the parameter estimation methods of our models. In particular, we have obtained the parameter estimates of the alternative testing-coverage function by using the regression analysis based on the integrable difference equation derived from the original differential equation. Then, we derive several software reliability assessment measures such as the software reliability function, the instantaneous and cumulative MTBF's. Finally, we have shown numerical examples of the estimated alternative testing-coverage function and the estimated SRGM by using C0 testing-coverage measure data recorded along with fault count data collected in an actual testing phase. Especially, for embedded software systems, the proposed alternative testing-coverage function and the SRGM enable us to see the maturity process of the testing-coverage, and simultaneously estimate the number of faults detected in the testing phase.

Further studies are needed that we have to discuss more about the useful software reliability assessment measures and the performance of the proposed model by using several actual data measured on the other testing-coverage criteria.

Acknowledgments

This research was partially supported by the Grant-in-Aid for Scientific Research (C)(2) from the Ministry of Education, Culture, Sports, Science, and Technology of Japan under Grant No. 15510129. And the first author was financially supported by the Sasakawa Scientific Research Grant from the Japan Science Society, Grant No. 16-064.

References

[1] J.D. Musa, A. Iannino, and K. Okumoto, *Software Reliability: Measurement, Prediction, Application*, McGraw-Hill, New York, 1987.

[2] S. Yamada, *Software Reliability Models: Fundamentals and Applications* (in Japanese), JUSE Press, Tokyo, 1994.

[3] H. Pham, *Software Reliability*, Springer-Verlag, Singapore, 2000.

[4] S. Yamada, "Software Reliability Models," in *Stochastic Models in Reliability and Maintenance*, (S. Osaki, ed.), Springer-Verlag, Berlin, 2002, pp. 253–280.

[5] K. Kanoun, M. Kaaniche, and J.C. Laprie, "Qualitative and quantitative reliability assessment," *IEEE Software*, Vo. 14, No. 2, pp. 77–87, 1987.

[6] P.K. Kapur and S. Younes, "Modelling an imperfect debugging phenomenon in software reliability," *Microelectronics and Reliability*, Vol. 36, No. 5, pp. 645–650, 1996.

[7] P. Zeephongsekul, "Reliability growth of a software model under imperfect debugging and generation of errors," *Microelectronics and Reliability*, Vol. 36, No. 10, pp. 1475–1482, 1996.

[8] S. Yamada, H. Ohtera, and M. Ohba, "Software reliability growth model with testing-domain and comparisons of goodness-of-fit," *Computers & Mathematics with Applications*, Vol. 24, No. 1/2, pp. 79–86, 1992.

[9] T. Fujiwara and S. Yamada, "C0 coverage-measure and testing-domain metrics based on a software reliability growth model," *International Journal of Reliability, Quality and Safety Engineering*, Vol. 9, No. 4, pp. 329–340, 2002.

[10] Y.K. Malaiya, M.N. Li, J.M. Bieman, and R. Karcich, "Software reliability growth with test coverage," *IEEE Transactions on Reliability*, Vol. 51, No. 4, pp. 420–426, 2002

[11] H. Pham and X. Zhang, "NHPP software reliability and cost models with testing coverage," *European Journal of Operational Research*, Vol. 145, No. 2, pp. 443–454, 2003.

[12] S. Yamada, M. Kimura, and M. Takahashi, *Statistical Quality Control for TQM*, Corona, Tokyo, 1998.

[13] A.L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, Vol. R-28, No. 3, pp. 206–211, 1979.

[14] R. Hirota, "Nonlinear partial difference equations. V. Nonlinear equations reducible to linear equations," *Journal of Physical Society of Japan*, Vol. 46, No. 1, pp. 312–319, 1979.

[15] S. Yamada, S. Inoue, and D. Satoh, "Statistical data analysis modeling based on difference equations for software reliability assessment," (in Japanese), *Transactions of the Japan Society for Industrial and Applied Mathematics*, Vol. 12, No. 2, pp. 155–168, 2002.