

SHIGERU YAMADA

SHUNJI OSAKI

HIROYUKI NARIHISA

**A software reliability growth model with
two types of errors**

RAIRO. Recherche opérationnelle, tome 19, n° 1 (1985),
p. 87-104

http://www.numdam.org/item?id=RO_1985__19_1_87_0

© AFCET, 1985, tous droits réservés.

L'accès aux archives de la revue « RAIRO. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

A SOFTWARE RELIABILITY GROWTH MODEL WITH TWO TYPES OF ERRORS (*)

by Shigeru YAMADA ⁽¹⁾, Shunji OSAKI ⁽²⁾
and Hiroyuki NARIHISA ⁽¹⁾

Abstract. — This paper considers a software error detection process with two types of errors: Some are easy to be detected and the other are difficult to be. The software reliability growth model for such an error detection process is formulated by a nonhomogeneous Poisson process. The error detection rate per error (per unit time) characterizing the software reliability growth of the model depends on the testing time. The assessment measures for software reliability are discussed. The model parameters and the assessment measures are estimated by the method of maximum likelihood. The model is applied to actual software error data.

Keywords: Software error; Types of errors; Software reliability growth model; Nonhomogeneous Poisson process; Error detection rate.

Résumé. — Cet article considère un processus de détection d'erreurs dans les logiciels avec deux types d'erreurs : certaines sont aisées à détecter, et les autres difficiles. Le modèle de croissance de fiabilité de logiciel pour un tel processus de détection d'erreur est formulé par un processus de Poisson non homogène. Le taux de détection d'erreurs par erreur (par unité de temps) caractérisant la croissance de la fiabilité du logiciel du modèle dépend du temps d'épreuve. Les mesures d'évaluation pour la fiabilité des logiciels sont examinées. Les paramètres du modèle et les mesures d'évaluation sont estimés par la méthode du maximum de vraisemblance. Le modèle est appliqué à un cas concret.

1. INTRODUCTION

Today software reliability is one of the urgent issues in software engineering for attaining the high reliability of a computer system. In this area the software reliability assessment is very important to measure and predict the reliability and performance of a software system. Several reliability models

(*) Received November 1983.

⁽¹⁾ Graduate School of Systems Science, Okayama University of Science, Ridai-cho 1-1, Okayama 700, Japan.

⁽²⁾ Department of Industrial and Systems Engineering, Faculty of Engineering, Hiroshima University, Higashi-Hiroshima 724, Japan.

have been proposed and investigated in recent years (Goel and Okumoto [5], Jelinski and Moranda [6], Kremer [7], Littlewood [8], Moranda [9], Musa [10], Schick and Wolverton [15], Shanthikumar [16], Shooman [17], Yamada and Osaki [19], and so on). These models attempt to estimate the error content of a software system and the time interval between software failures, and to predict the software reliability. The common approach to software reliability is to describe a software error detection process which represents the behavior of errors during testing phase in the software development. During this phase the software system is tested to detect and correct software errors. A software failure is defined as an unacceptable departure of program operation caused by an error in the software system. The models applicable to the assessment of software reliability during the testing phase are called software reliability growth models (Ramamoorthy and Bastani [13]).

This paper develops a software reliability growth model which provides a plausible description of software failure occurrence phenomena. The software error detection process is characterized by two types of errors: Some are easy to be detected and the other are difficult to be. The detectability of an error in such an error detection process is considered to be nonhomogeneous over the testing period. Then, the error detection rate per error is assumed not to be constant but depends on the testing time. The constant error detection rate is the usual assumption of the software reliability growth models proposed so far. The software reliability growth model considered here is formulated by a stochastic process which is based on a nonhomogeneous Poisson process (NHPP) (Ross [14] and Thompson [18]).

The description of the model is presented in section 2. The meaningful assessment measures for software reliability are discussed in section 3. The maximum likelihood estimations of the model parameters are given in section 4. The asymptotic properties of the maximum likelihood estimates of the model parameters are applied to the estimations of the assessment measures for software reliability in section 5. Finally, the application of the model to actual data is presented in section 6.

2. MODEL DESCRIPTION

Consider an implemented software system which is tested in the software development. During the testing phase software errors remaining in the system can be detected and corrected. The usual assumptions are made:

1. A software system is subject to failures at random times caused by errors present in the system.

2. A failure is caused by an error.
3. Each time a failure occurs the error which caused it can be immediately removed, and no other error are introduced.

Such an error detection (or failure occurrence) is modeled by a software reliability growth theory. The reliability growth theory has been originally developed for assessing the reliability in the hardware product development (Crow [2] and Duane [3]). This paper develops a useful software reliability growth model reflecting the error detection process for real software development projects. The model can estimate software reliability in terms of the number of errors remaining in the system.

In general the chance of detecting an error on a given test run is not constant. The reason is that the errors detected early in the testing are different from those detected later on. This can be incorporated by assuming that there are two types of errors: Some errors that are easy to be detected and the other errors that are difficult to be. The former are defined as type 1 errors, the latter as type 2 errors. The existence of such two types of errors is known by the test personnel in the actual testing (e.g., Ohba and Kajiyama [12]). An error detection is assumed to mean a failure occurrence synonymously. The error detection process with such two types of errors can be described by an NHPP.

Suppose that the total expected number of errors to be eventually detected is a and the content proportion of type i ($i=1, 2$) error is $p_i > 0$ where $p_1 + p_2 = 1$. Then, the initial error content of type i ($i=1, 2$) errors is $p_i \cdot a$. The total number of errors detected up to time t for a type i error detection process is assumed to follow an NHPP with the mean value function:

$$m_i(t) = p_i a (1 - e^{-b_i t}) \quad (i=1, 2), \quad (1)$$

where $m_i(0) = 0$ and $m_i(\infty) = p_i a$. The parameter b_i ($i=1, 2$) is constant and is interpreted as the error detection rate of type i error per error (per unit time), i. e.,

$$\lim_{h \rightarrow 0} \frac{m_i(t+h) - m_i(t)}{(p_i a - m_i(t)) h} = b_i \quad (i=1, 2), \quad (2)$$

where it can be assumed that:

$$0 < b_2 < b_1 < 1. \quad (3)$$

Goel and Okumoto [5] first introduced an elementary NHPP model for an error detection process without distinction of error types by assuming that the error detection rate per error is constant.

Let $\{N(t), t \geq 0\}$ denote a counting process representing the total number of types 1 and 2 errors detected up to time t . Then, the error detection process with two types of errors is formulated by an NHPP as:

$$\Pr \{N(t) = n\} = \frac{\{m_p(t)\}^n}{n!} e^{-m_p(t)} \quad (n=0, 1, 2, \dots), \quad (4)$$

$$m_p(t) = \sum_{i=1}^2 m_i(t), \quad (5)$$

where:

$$m_p(0) = 0 \quad \text{and} \quad m_p(\infty) = a. \quad (6)$$

The growth curves of $m_p(t)$ are plotted in figure 1. Obviously, if $b_1 = b_2$, then the NHPP model defined by (4) and (5) coincides with that of Goel and Okumoto [5]. From (5) the NHPP's intensity function which means the number of types 1 and 2 errors detected per unit time at time t is:

$$\lambda_p(t) \equiv \frac{dm_p(t)}{dt} = a \sum_{i=1}^2 p_i b_i e^{-b_i t}. \quad (7)$$

It is of great use to investigate the reliability growth aspect in terms of the error detection rate per error at an arbitrary testing time point. The error detection rate per error (per unit time) at time t is given by:

$$d_p(t) \equiv \lambda_p(t)/(a - m_p(t)) = \sum_{i=1}^2 \left(\frac{p_i e^{-b_i t}}{p_1 e^{-b_1 t} + p_2 e^{-b_2 t}} \right) \cdot b_i. \quad (8)$$

From (8), it is shown that:

$$\frac{dd_p(t)}{dt} < 0 \quad \text{for} \quad t \geq 0, \quad (9)$$

which implies that $d_p(t)$ is a monotone decreasing function with:

$$d_p(0) = \sum_{i=1}^2 p_i b_i \quad \text{and} \quad d_p(\infty) = b_2. \quad (10)$$

The equations (9) and (10) imply that most of remaining errors in the late phase of testing are type 2 errors, i. e., being difficult to be detected. Further, if $p_1 > p_2$, i. e., type 1 errors (easy to be detected) of the initial error content remain in the system more than type 2 errors (difficult to be detected), which

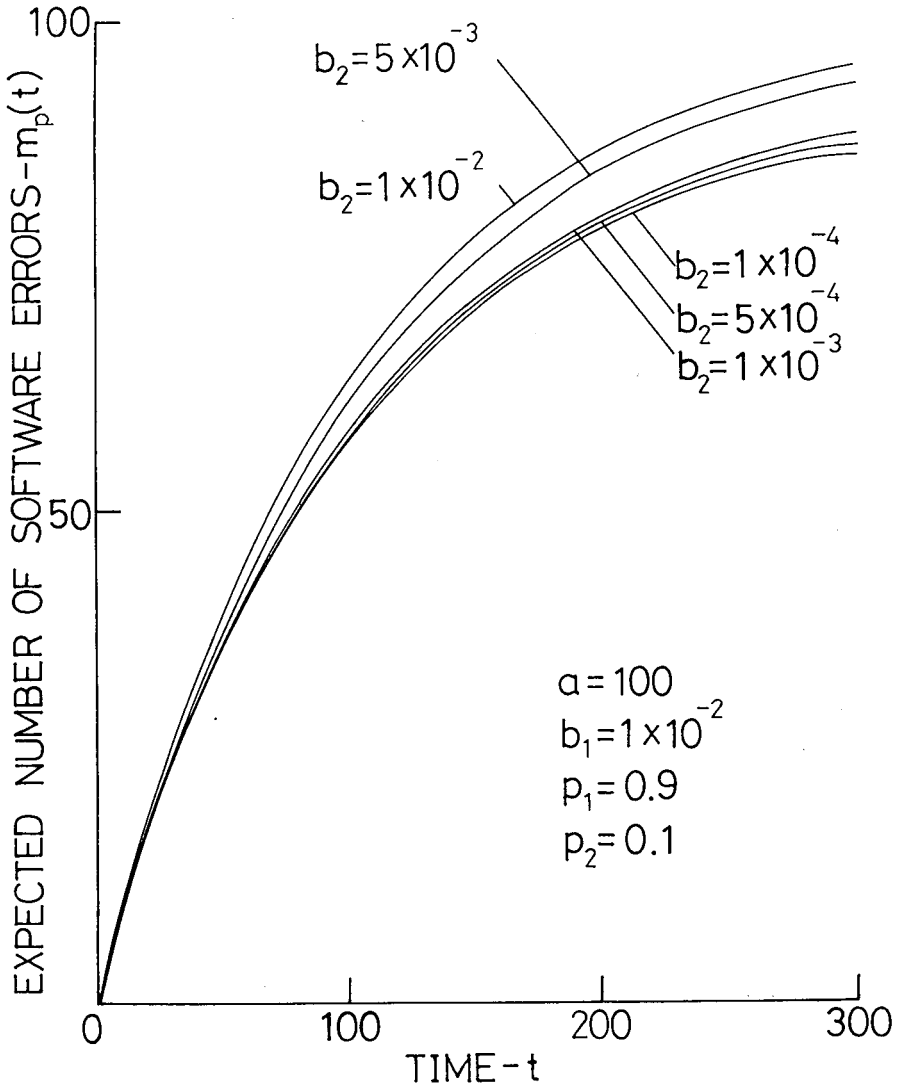


Figure 1. - Software reliability growth curves of $m_p(t)$.

may be a common case, then:

$$\frac{d^2 d_p(t)}{dt^2} \begin{cases} < 0 & (0 < t < \tilde{t}), \\ > 0 & (\tilde{t} < t < \infty), \end{cases} \tag{11}$$

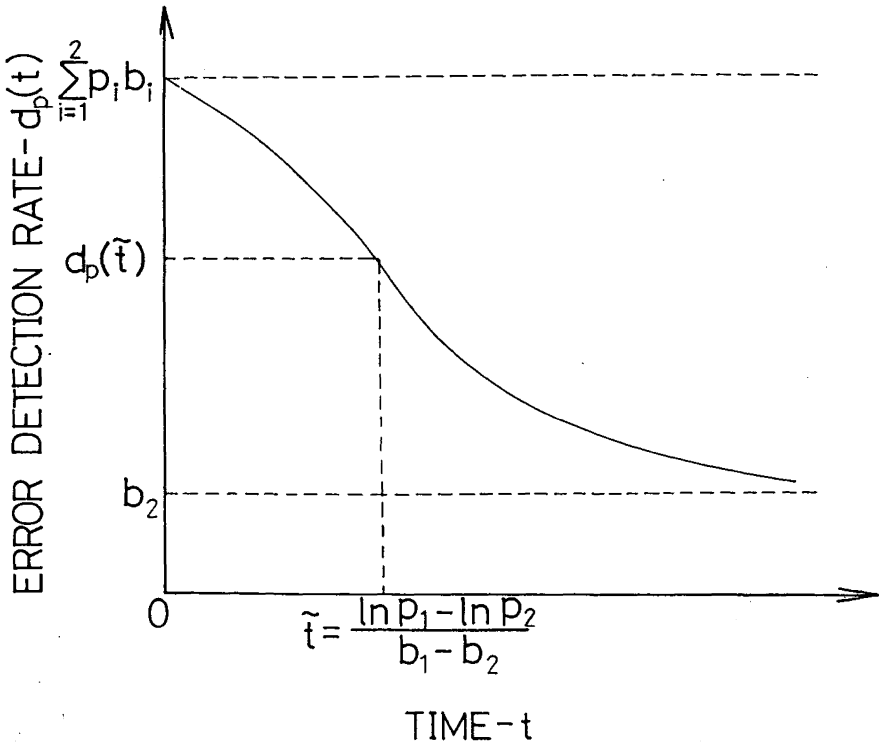


Figure 2. — An error detection rate per error (per unit time) of $d_p(t)$.

where \tilde{t} is an inflection point of $d_p(t)$ and is given by:

$$\tilde{t} = \frac{\ln p_1 - \ln p_2}{b_1 - b_2}. \quad (12)$$

If $p_1 \leq p_2$, i. e., type 2 errors of the initial error content remain in the system more than type 1 errors, then:

$$\frac{d^2 d_p(t)}{dt^2} < 0 \quad (0 < t < \infty). \quad (13)$$

The curve of $d_p(t)$ is plotted in figure 2.

3. ASSESSMENT MEASURES FOR SOFTWARE RELIABILITY

In the following, the assessment measures for software reliability are discussed for the NHPP model defined by (4) and (5). The measures discussed here are:

- (1) time-interval between software failures;
- (2) number of software errors remaining in the system;
- (3) software reliability.

3.1. Time interval between failures

Let X_k denote a random variable representing the time-interval between $(k - 1)$ st and k th failures ($k = 1, 2, \dots$). Then, $S_k \equiv \sum_{i=1}^k X_i$ is a random variable representing the k th failure occurrence time. The joint probability density function of $\{S_1, S_2, \dots, S_n\}$ is given by:

$$f_{S_1, S_2, \dots, S_n}(s_1, s_2, \dots, s_n) = \exp[-m_p(s_n)] \prod_{i=1}^n \lambda_p(s_i). \tag{14}$$

If the test is stopped at a fixed time, say T_c , then the joint probability density function of $\{S_1, S_2, \dots, S_n\}$ is given by:

$$f_{S_1, S_2, \dots, S_n}(s_1, s_2, \dots, s_n | T_c) = \exp[-m_p(T_c)] \prod_{i=1}^n \lambda_p(s_i), \tag{15}$$

where $0 \leq s_1 \leq s_2 \leq \dots \leq s_n \leq T_c$. The software error data set is said to be subject to fixed time censoring and T_c is called a censoring time. From (14) the marginal density of $\{S_k, S_{k+1}, \dots, S_n\}$ can be obtained as:

$$f_{S_k, S_{k+1}, \dots, S_n}(s_k, s_{k+1}, \dots, s_n) = \exp[-m_p(s_n)] \frac{\prod_{i=k}^n \lambda_p(s_i) \{m_p(s_k)\}^{k-1}}{\Gamma(k)} \tag{16}$$

$$(k = 1, 2, \dots, n),$$

where $\Gamma(k)$ denotes a Gamma function. The probability density function $f_{S_k}(s_k)$ of S_k can be obtained by deriving the marginal density of S_k from (16):

$$f_{S_k}(s_k) = \frac{\lambda_p(s_k) \{m_p(s_k)\}^{k-1}}{\Gamma(k)} \exp[-m_p(s_k)] \quad (17)$$

$$(k = 1, 2, \dots, n).$$

It should be noted that the cumulative distribution function of S_k is improper since:

$$F_{S_k}(\infty) = \lim_{s_k \rightarrow \infty} \int_0^{s_k} f_{S_k}(u) du = 1 - \frac{1}{\Gamma(k)} \int_a^\infty y^{k-1} e^{-y} dy < 1. \quad (18)$$

The equation (18) implies that there does not exist the mean of the distribution of S_k . Then, the joint distribution function of $\{S_1, S_2, \dots, S_n\}$ is also improper. Consequently, there does not exist the mean time-interval of failures. The similar discussion above on the distribution of time-interval between failures is presented in detail by Yamada and Osaki [21].

3.2. Number of remaining errors

Let the number of errors remaining in the system at time t be:

$$\bar{N}(t) \equiv N(\infty) - N(t). \quad (19)$$

The limiting distribution of $N(t)$ is given by:

$$\lim_{t \rightarrow \infty} \Pr \{N(t) = n\} = \frac{a^n}{n!} e^{-a} \quad (n = 0, 1, 2, \dots), \quad (20)$$

which means a Poisson distribution with mean a . Then, the expectation of $\bar{N}(t)$ is:

$$E[\bar{N}(t)] = a - m_p(t) = a \sum_{i=1}^2 p_i e^{-b_i t}. \quad (21)$$

The covariance between $N(\infty)$ and $N(t)$ is given by:

$$\text{Cov}[N(\infty), N(t)] = m_p(t), \quad (22)$$

and the variance of $\bar{N}(t)$ can be obtained as:

$$\text{Var}[\bar{N}(t)] = \text{Var}[N(\infty)] + \text{Var}[N(t)] - 2 \text{Cov}[N(\infty), N(t)] = a - m_p(t), \quad (23)$$

which is equal to the expectation of $\bar{N}(t)$.

Now, suppose that n_d errors have been detected up to time t . The conditional distribution of $\bar{N}(t)$, given that $N(t) = n_d$, and its expectation are given by:

$$\Pr\{\bar{N}(t) = x | N(t) = n_d\} = \frac{\{a - m_p(t)\}^x}{x!} \exp[-\{a - m_p(t)\}] \quad (24)$$

$$(x = 0, 1, 2, \dots),$$

$$E[\bar{N}(t) | N(t) = n_d] = a - m_p(t). \quad (25)$$

That is, the distribution of the number of remaining errors at time t is a Poisson distribution with mean $\{a - m_p(t)\}$.

3.3. Software reliability

Assume that the failure occurrence time S_{k-1} for the $(k-1)$ st failure is given. Then, the conditional reliability function of X_k is given by:

$$R(x | s) \equiv \Pr\{X_k > x | S_{k-1} = s\} = \exp\left[-\sum_{i=1}^2 e^{-b_i s} m_i(x)\right], \quad (26)$$

which represents the probability that a failure does not occur in $(s, s+x]$. The conditional reliability function $R(x | s)$ is called software reliability (*cf.* Goel and Okumoto [5]). From (26), if $x \rightarrow \infty$, then:

$$1 - R(\infty | s) = 1 - \exp\left[-a \sum_{i=1}^2 p_i e^{-b_i s}\right] < 1, \quad (27)$$

which implies that the conditional time-interval distribution of X_k is an improper distribution ($k = 1, 2, \dots$).

4. MAXIMUM LIKELIHOOD ESTIMATIONS OF PARAMETERS

First, suppose that the data on n failure occurrence times $\mathbf{s} \equiv (s_1, s_2, \dots, s_n)$ ($0 \leq s_1 \leq s_2 \leq \dots \leq s_n$) are observed during the testing. Then, the likelihood function for the unknown parameters a and b_i ($i = 1, 2$) in the NHPP model with $m_p(t)$, given \mathbf{s} , is given by (14). Taking the natural logarithm of the

likelihood function, the maximum likelihood estimates \hat{a} and \hat{b}_i ($i=1, 2$) can be obtained by solving the following likelihood equations under condition that $0 < b_2 < b_1$:

$$\frac{n}{a} = \sum_{i=1}^2 p_i (1 - e^{-b_i s_n}), \tag{28}$$

$$a s_n e^{-b_j s_n} = \sum_{k=1}^n \frac{(e^{-b_j s_k} - b_j s_k e^{-b_j s_k})}{\binom{2}{\sum_{i=1}^2 p_i b_i e^{-b_i s_k}}} \tag{29}$$

$(j=1, 2).$

From (15), if the data set s is subject to censoring at time T_c , then the simultaneous likelihood equations are given by:

$$\frac{n}{a} = \sum_{i=1}^2 p_i (1 - e^{-b_i T_c}), \tag{30}$$

$$a T_c e^{-b_j T_c} = \sum_{k=1}^n \frac{(e^{-b_j s_k} - b_j s_k e^{-b_j s_k})}{\binom{2}{\sum_{i=1}^2 p_i b_i e^{-b_i s_k}}} \tag{31}$$

$(j=1, 2).$

Next, suppose that the data on the cumulative number of detected errors, y_k , in a given time interval $(0, t_k]$ ($k=1, 2, \dots, n; 0 < t_1 < t_2 < \dots < t_n$) are observed, where:

$$\mathbf{t} \equiv (t_1, t_2, \dots, t_n) \quad \text{and} \quad \mathbf{y} \equiv (y_1, y_2, \dots, y_n).$$

Then, the joint probability mass function of:

$$\{N(t_1)=y_1, N(t_2)=y_2, \dots, N(t_n)=y_n\},$$

i. e., the likelihood function for the unknown parameters a and b_i ($i=1, 2$) in the NHPP model with $m_p(t)$, given (\mathbf{t}, \mathbf{y}) , is given by:

$$L \equiv \Pr \{ N(t_1)=y_1, N(t_2)=y_2, \dots, N(t_n)=y_n \} \\ = \prod_{k=1}^n \frac{\{m_p(t_k) - m_p(t_{k-1})\}^{y_k - y_{k-1}}}{(y_k - y_{k-1})!} \exp[-\{m_p(t_k) - m_p(t_{k-1})\}], \tag{32}$$

where $t_0 \equiv 0$ and $y_0 \equiv 0$. Then, the natural logarithm of the likelihood function can be obtained as:

$$\ln L = \sum_{k=1}^n (y_k - y_{k-1}) \ln a + \sum_{k=1}^n (y_k - y_{k-1}) \ln \left[\sum_{i=1}^2 p_i (e^{-b_i t_{k-1}} - e^{-b_i t_k}) \right] - a \sum_{i=1}^2 p_i (1 - e^{-b_i t_n}) - \sum_{k=1}^n \ln (y_k - y_{k-1})! \quad (33)$$

Letting $\partial \ln L / \partial a = \partial \ln L / \partial b_i = 0$ ($i = 1, 2$), the following simultaneous likelihood equations can be obtained:

$$\frac{y_n}{a} = \sum_{i=1}^2 p_i (1 - e^{-b_i t_n}), \quad (34)$$

$$a t_n e^{-b_j t_n} = \sum_{k=1}^n \frac{(y_k - y_{k-1}) (t_k e^{-b_j t_k} - t_{k-1} e^{-b_j t_{k-1}})}{\left\{ \sum_{i=1}^2 p_i (e^{-b_i t_{k-1}} - e^{-b_i t_k}) \right\}} \quad (35)$$

$(j = 1, 2),$

which can be solved numerically under condition that $0 < b_2 < b_1$.

5. ASYMPTOTIC PROPERTIES OF ESTIMATED PARAMETERS AND THEIR APPLICATIONS

For the software error data (t, y) , the distribution of the estimated parameters for large samples can be obtained. That is, when the sample size n is sufficiently large, the maximum likelihood estimates \hat{a} and \hat{b}_i ($i = 1, 2$) asymptotically follow a trivariate normal distribution as:

$$\begin{matrix} \hat{a} \\ \hat{b}_1 \\ \hat{b}_2 \end{matrix} \sim N \left(\begin{matrix} a \\ b_1 \\ b_2 \end{matrix}, \Sigma \right) \quad (n \rightarrow \infty). \quad (36)$$

In (36) the means are true values of a and b_i ($i = 1, 2$), respectively, and the variance-covariance matrix of \hat{a} and \hat{b}_i ($i = 1, 2$) is given by the inverse matrix of the Fisher's information matrix.

Noting that:

$$E[N(t_k)] = m_p(t_k) \quad (k = 1, 2, \dots, n), \tag{37}$$

the Fisher's information matrix for \hat{a} and \hat{b}_i ($i = 1, 2$) can be derived from (33) as:

$$\begin{aligned} \mathbf{F} = & \begin{matrix} E\left[-\frac{\partial^2 \ln L}{\partial a^2}\right] & E\left[-\frac{\partial^2 \ln L}{\partial a \partial b_1}\right] & E\left[-\frac{\partial^2 \ln L}{\partial a \partial b_2}\right] \\ E\left[-\frac{\partial^2 \ln L}{\partial b_1 \partial a}\right] & E\left[-\frac{\partial^2 \ln L}{\partial b_1^2}\right] & E\left[-\frac{\partial^2 \ln L}{\partial b_1 \partial b_2}\right] \\ E\left[-\frac{\partial^2 \ln L}{\partial b_2 \partial a}\right] & E\left[-\frac{\partial^2 \ln L}{\partial b_2 \partial b_1}\right] & E\left[-\frac{\partial^2 \ln L}{\partial b_2^2}\right] \end{matrix} \\ & \begin{matrix} \frac{\beta_1 + \beta_2}{a} & \alpha_1 & \alpha_2 \\ \alpha_1 & a \sum_{k=1}^n \frac{\delta_1^2(k)}{[\gamma_1(k) + \gamma_2(k)]} & a \sum_{k=1}^n \frac{\delta_1(k) \delta_2(k)}{[\gamma_1(k) + \gamma_2(k)]} \\ \alpha_2 & a \sum_{k=1}^n \frac{\delta_1(k) \delta_2(k)}{[\gamma_1(k) + \gamma_2(k)]} & a \sum_{k=1}^n \frac{\delta_2^2(k)}{[\gamma_1(k) + \gamma_2(k)]} \end{matrix}, \tag{38} \end{aligned}$$

where:

$$\alpha_i = p_i t_n e^{-b_i t_n}, \tag{39}$$

$$\beta_i = p_i (1 - e^{-b_i t_n}), \tag{40}$$

$$\gamma_i(k) = p_i (e^{-b_i t_{k-1}} - e^{-b_i t_k}), \tag{41}$$

$$\delta_i(k) = p_i (t_k e^{-b_i t_k} - t_{k-1} e^{-b_i t_{k-1}}), \tag{42}$$

for $i = 1, 2$. Applying the maximum likelihood estimates \hat{a} and \hat{b}_i ($i = 1, 2$) to (38) and calculating the inverse matrix of it, the estimated asymptotic variance-covariance matrix $\hat{\Sigma}$ can be obtained as:

$$\hat{\Sigma} = \mathbf{F}^{-1} \Big|_{a=\hat{a}, b_i=\hat{b}_i (i=1, 2)}. \tag{43}$$

Using the maximum likelihood estimates \hat{a} and \hat{b}_i ($i = 1, 2$) together with their asymptotic properties, the point and interval estimations of the assessment measures for software reliability, i.e., the expected number of errors remaining in the system at time t of (21) and the software reliability of (26), can be made. Let $f(a, b_1, b_2)$ denote a function of the parameters a and b_i ($i = 1, 2$). Then, the maximum likelihood estimate $\hat{f}(a, b_1, b_2)$ of $f(a, b_1, b_2)$,

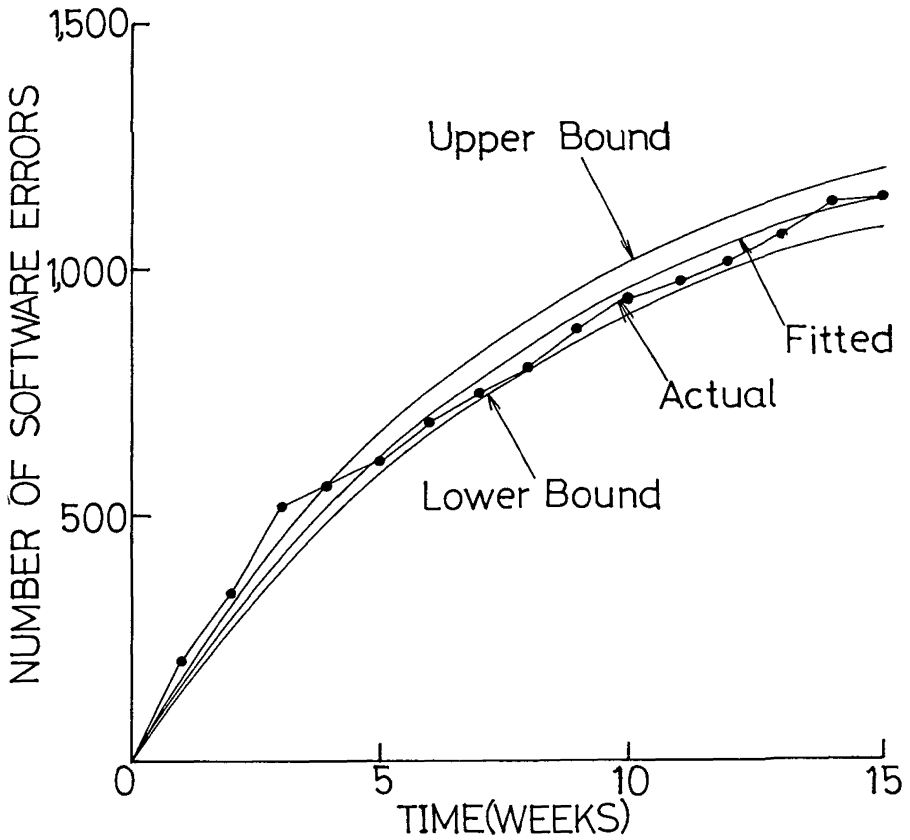


Figure 3. — Estimated mean value function $\hat{m}_p(t)$ and the 90% confidence bounds for the observed data.

is given by:

$$\hat{f}(a, b_1, b_2) = f(\hat{a}, \hat{b}_1, \hat{b}_2), \tag{44}$$

(see Bain [1]). For large samples, if $f(a, b_1, b_2)$ is continuously differentiable, then $\hat{f}(a, b_1, b_2)$ follows asymptotically a normal distribution. The mean is $f(a, b_1, b_2)$ for the true values of a and b_i ($i=1, 2$) and the variance is:

$$\text{Var}[\hat{f}(a, b_1, b_2)] = \begin{pmatrix} \frac{\partial f}{\partial a} & \frac{\partial f}{\partial b_1} & \frac{\partial f}{\partial b_2} \end{pmatrix} \Sigma \begin{pmatrix} \frac{\partial f}{\partial a} \\ \frac{\partial f}{\partial b_1} \\ \frac{\partial f}{\partial b_2} \end{pmatrix}. \tag{45}$$

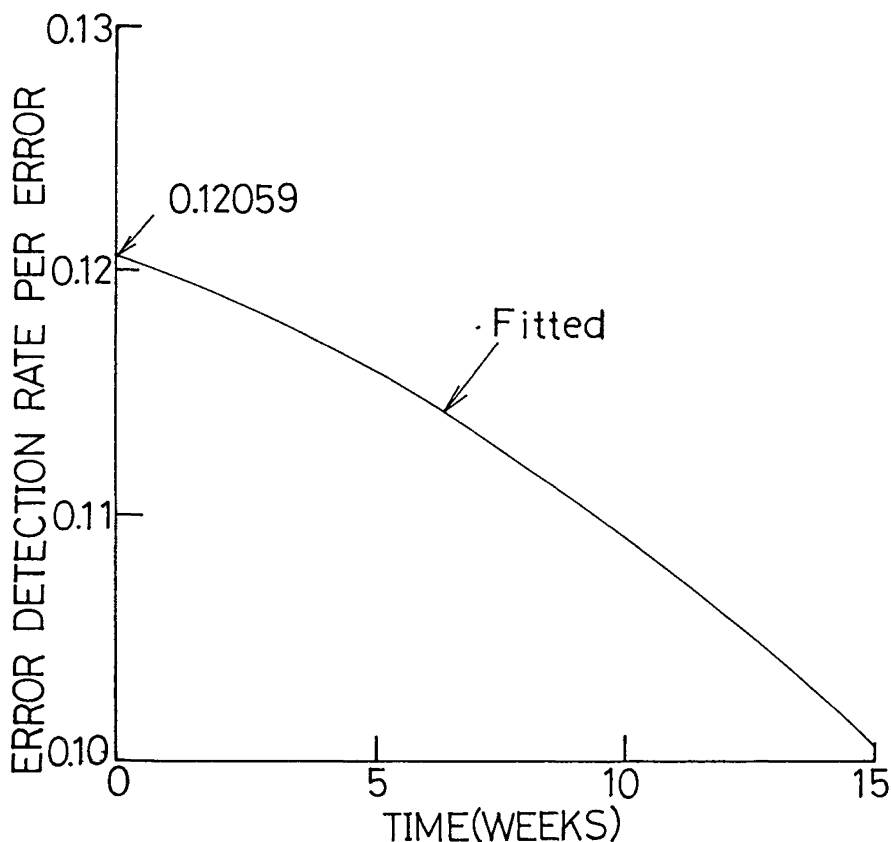


Figure 4. — Estimated error detection rate per error $\hat{d}_p(t)$.

Thus, the $100\gamma\%$ confidence bounds of $f(a, b_1, b_2)$ are given by:

$$\hat{f}(a, b_1, b_2) \pm K_\gamma \sqrt{\text{Var}[\hat{f}(a, b_1, b_2)]}, \quad (46)$$

where K_γ is $100(1+\gamma)/2\%$ point of the standard normal distribution (see Nelson [11]).

Applying $E[\bar{N}(t)]$ of (21) and $R(x|s)$ of (26) to $f(a, b_1, b_2)$ and calculating the quantities for the maximum likelihood estimates \hat{a} and \hat{b}_i ($i=1, 2$), the point and interval estimations of the assessment measures can be made.

6. EXAMPLE OF APPLICATION

For illustration of software reliability analysis, consider the software error data set DS1 cited by Goel [4], which is available in the form (t_k, y_k) , $k = 1, 2, \dots, 15$ (weeks). Based on a Newton-Raphson method, solving (34) and (35) numerically for the data set yields:

$$\hat{a} = 1398.6, \quad \hat{b}_1 = 0.1300, \quad \text{and} \quad \hat{b}_2 = 0.0359, \quad (47)$$

where it is assumed that prespecified content proportions are $p_1 = 0.9$ and $p_2 = 0.1$. The estimated $\hat{m}_p(t)$ is shown in figure 3 along with the actual data and the 90% confidence bounds. Then, the estimated $\hat{d}_p(t)$ characterizing the software reliability growth of the NHPP model with (47) is shown in figure 4. Applying a Kolmogorov-Smirnov goodness-of-fit test (see Yamada and Osaki [20]), we showed that at a 5% level of significance the NHPP model with (47) adequately fits the observed data.

From (21) and (26) the estimated $\hat{E}[\bar{N}(t)]$ and $\hat{R}(x|s)$ for the estimated parameters of (47) can be obtained. For the estimated parameters of (47), the Fisher's information matrix is given by:

$$\hat{F} = \begin{pmatrix} 2.4901 \times 10^{-4} & 1.8573 & 0.3824 \\ 1.8573 & 3.7041 \times 10^4 & 4.5500 \times 10^3 \\ 0.3824 & 4.5500 \times 10^3 & 7.1578 \times 10^2 \end{pmatrix} \quad (48)$$

Then, using the asymptotic properties for the maximum likelihood estimates of model parameters, we can obtain the confidence bounds for the estimated $\hat{E}[\bar{N}(t)]$ and $\hat{R}(x|s)$ ($s = 15.0$ (weeks)) which are shown in figures 5 and 6 along with the 90% confidence bounds, respectively.

7. CONCLUDING REMARKS

This paper discussed the software reliability growth model describing an error detection process with two types of errors. The underlying stochastic process is an NHPP with mean value function:

$$m_p(t) = \sum_{i=1}^2 p_i a (1 - e^{-b_i t}).$$

The reliability growth is characterized by the error detection rate per error (per unit time) in the model.

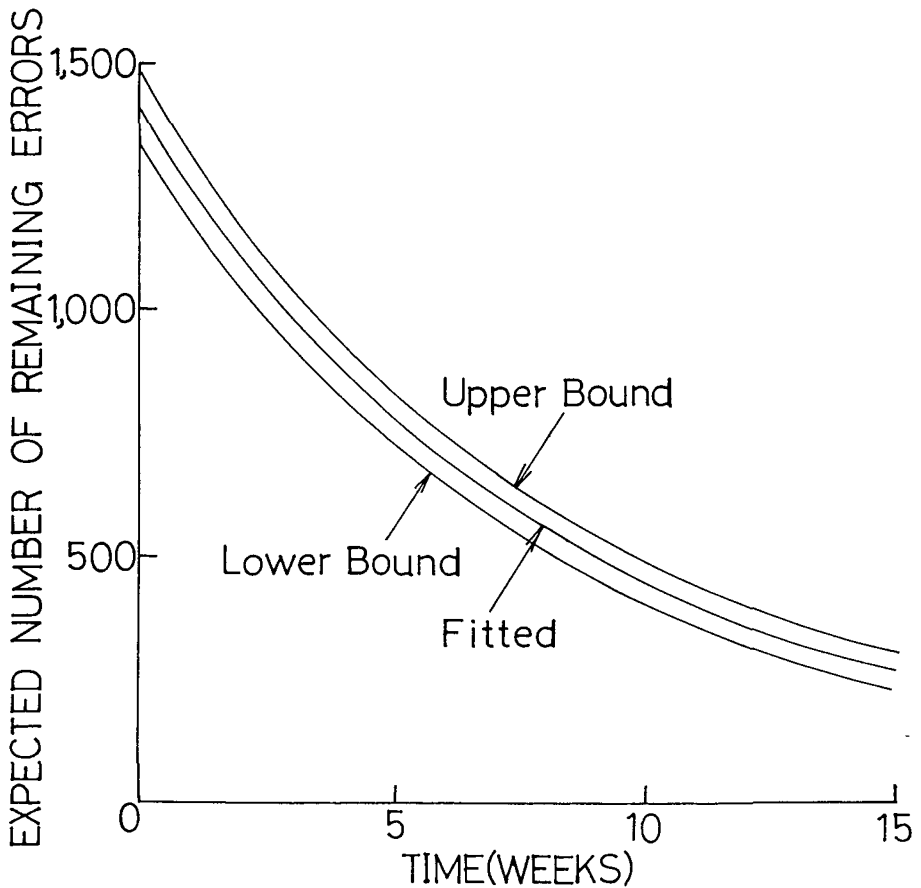


Figure 5. — Estimated expected number of remaining errors $\hat{E}[\bar{N}(t)]$ and the 90% confidence bounds for the observed data.

In general the difficulty of an error detection may be of some degrees. This can be incorporated by assuming that there are k types of errors. Then, if the errors detected during the testing are classified into k types, then the software reliability growth model in this paper can be generalized as the NHPP model with the mean value function:

$$m_p(t) = \sum_{i=1}^k p_i a (1 - e^{-b_i t}),$$

where:

$$\sum_{i=1}^k p_i = 1, \quad p_i > 0,$$

$$0 < b_k < b_{k-1} < \dots < b_2 < b_1 < 1.$$

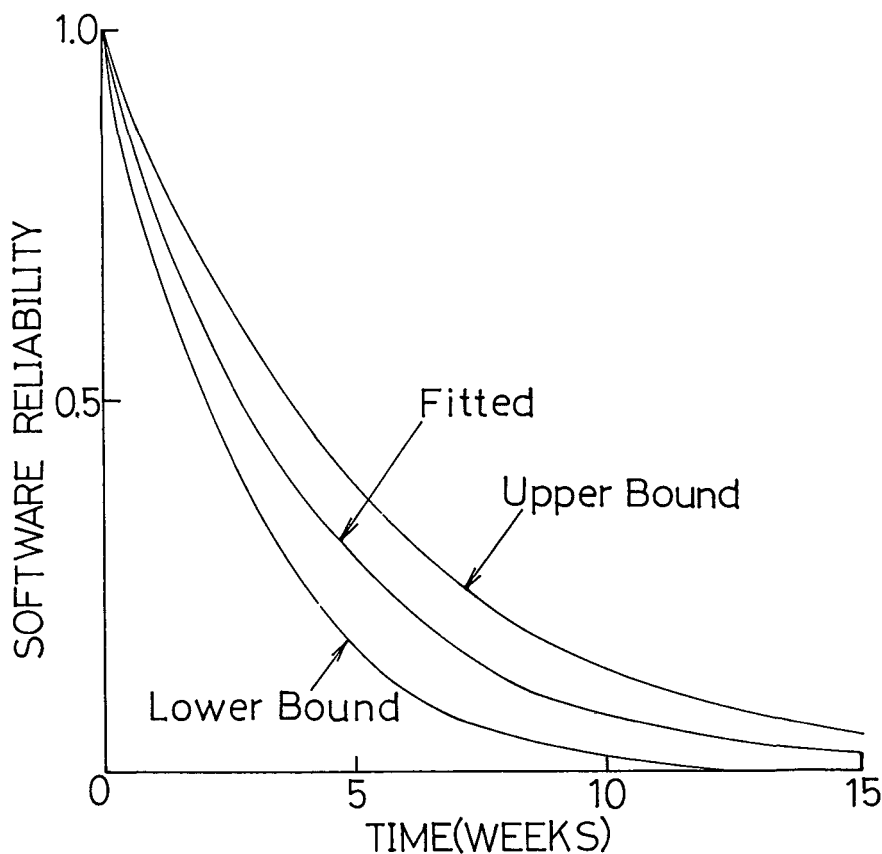


Figure 6. — Estimated software reliability $\hat{R}(x|s)$ ($s=15.0$) and the 90% confidence bounds for the observed data.

REFERENCES

1. L. J. BAIN, *Statistical Analysis of Reliability and Life-Testing Models*, Marcel Dekker, Inc., New York, 1978.
2. L. H. CROW, *On Tracking Reliability Growth*, Proc. 1975 Annu. Reliability and Maintainability Symp., 1975, pp. 438-443.
3. J. T. DUANE, *Learning Curve Approach to Reliability Monitoring*, I.E.E.E. Trans. Aerosp., Vol. 2, 1964, pp. 563-566.
4. A. L. GOEL, *Software Error Detection Model with Applications*, J. Syst. Software, Vol. 1, 1980, pp. 243-249.
5. A. L. GOEL and K. OKUMOTO, *Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures*, I.E.E.E. Trans. Reliab., Vol. R-28, 1979, pp. 206-211.
6. Z. JELINSKI and P. B. MORANDA, *Software Reliability Research*, in *Statistical Computer Performance Evaluation*, W. FREIBERGER, Ed., Academic Press, New York, 1972, pp. 465-484.

7. W. KREMER, *Birth-Death and Bug Counting*, I.E.E.E. Trans. Reliab., Vol. R-32, 1983, pp. 37-47.
8. B. LITTLEWOOD, *Theories of Software Reliability: How Good Are They and How Can They Be Improved?*, I.E.E.E. Trans. Software Eng., Vol. SE-6, 1980, pp. 489-500.
9. P. B. MORANDA, *Event-Altered Rate Models for General Reliability Analysis*, I.E.E.E. Trans. Reliab., Vol. R-28, 1979, pp. 376-381.
10. J. D. MUSA, *Software Reliability Measurement*, J. Syst. Software, Vol. 1, 1980, pp. 223-241.
11. W. NELSON, *Applied Life Data Analysis*, John Wiley and Sons, Inc., New York, 1982.
12. M. OHBA and M. KAJIYAMA, *Inflection S-shaped Software Reliability Growth Model* (in Japanese), I.P.S.-Japan Proc. W.G.S.E. Meeting, Vol. 28, 1983.
13. C. V. RAMAMOORTHY and F. B. BASTANI, *Software Reliability—Status and Perspectives*, I.E.E.E. Trans. Software Eng., Vol. SE-8, 1982, pp. 354-371.
14. S. M. ROSS, *Stochastic Processes*, John Wiley and Sons, Inc., New York, 1983.
15. G. J. SCHICK and R. W. WOLVERTON, *An Analysis of Competing Software Reliability Models*, I.E.E.E. Trans. Software Eng., Vol. SE-4, 1978, pp. 104-120.
16. J. G. SHANTHIKUMAR, *A State- and Time-Dependent Error Occurrence-Rate Software Reliability Model with Imperfect Debugging*, Proc. 1981 Nat. Comput. Conf., 1981, pp. 311-315.
17. M. L. SHOOMAN, *Software Reliability: Measurement and Models*, Proc. 1975 Annu. Reliability and Maintainability Symp., 1975, pp. 485-491.
18. W. A. THOMPSON, JR., *On The Foundations of Reliability*, Technometrics, Vol. 23, 1981, pp. 1-13.
19. S. YAMADA and S. OSAKI, *Software Reliability Growth Models and Their Comparisons* (in Japanese), Trans. I.E.C.E. Japan, Vol. J65-D, 1982, pp. 906-912; also Systems. Computers. Controls, Vol. 13, 1982, pp. 42-49.
20. S. YAMADA and S. OSAKI, *Reliability Growth Models for Hardware and Software Systems Based on Nonhomogeneous Poisson Processes: A Survey*, Microelectron. Reliab., Vol. 23, 1983, pp. 91-112.
21. S. YAMADA and S. OSAKI, *S-shaped Software Reliability Growth Models with Four Types of Software Error Data*, Int. J. Systems Sci., Vol. 14, 1983, pp. 683-692.