American Journal of Operations Research, **2011, 1, 1-7**
doi: 10.4236/ajor.2011.11001 Published Online March 2011 (http://www.scirp.org/journal/ajor)

A Bivariate Software Reliability Model with Change-Point and Its Applications

Shinji Inoue, Shigeru Yamada
Department of Social Management Engineering, Graduate School of Engineering, Tottori University
E-mail: *{ino,yamada}@sse.tottori-u.ac.jp*

Received February 28, 2011*; Revised March* 20, 2011*; accepted March* 23, 2011

Abstract

Testing-time when a change of a stochastic characteristic of the software failure-occurrence time or software failure-occurrence time-interval is observed is called change-point. It is said that effect of the change-point on the software reliability growth process influences on accuracy for software reliability assessment based on a software reliability growth model (SRGM). We propose an SRGM with the effect of the change-point based on a bivariate SRGM, in which the software reliability growth process is assumed to depend on the testing-time and testing-effort factors simultaneously, for accurate software reliability assessment. And we discuss an optimal software release problem for deriving optimal testing-effort expenditures based on our model. Further, we show numerical examples of software reliability assessment based on our bivariate SRGM and estimation of optimal testing-effort expenditures by using actual data.

Keywords: Software Reliability, Software Reliability Growth Factor, Change-Point, Bivariate Software Reliability Growth Model, Optimal Testing-Effort Expending Problem

1. Introduction

We are required to conduct quantitative software quality/reliability assessment in terms of software quality assurance in a testing phase. And it is very important to measure software quality/reliability of the final software product with accuracy in the testing-phase. A software reliability growth model (abbreviated as SRGM) [1-4] is known as one of the useful mathematical tool for quantitative assessment of software reliability. This mathematical model enables us to describe a software reliability growth process observed in the actual testing-phase by treating the software failure-occurrence or the software fault-detection phenomenon as random variables.

Needless to say, it is preferable that the SRGMs are developed under feasible modeling assumptions, which reflect actual software failure-occurrence phenomena. Most of SRGMs proposed so far have been developed under the following assumptions: (1) the software reliability growth process depends only on the testing-time essentially, (2) the stochastic characteristics for the software failure -occurrence or the software fault-detection phenomenon does not change throughout the testing-phase. In an actual testing-phase, it is not necessarily that the common assumptions mentioned above is always appropriate. That means, it is natural to consider the software reliability growth process observed in the actual testing-phase depends on not only the testing-time but also the other software reliability growth factors, such as the testing-coverage, the testing-effort expenditures, the number of executed test-cases [5-7]. And the stochastic characteristics for the software failure-occurrence or the software fault-detection phenomenon changes due to changing the fault-target, the difference of the fault-density for each module, and so forth [8-14]. Especially, testing-time when the stochastic characteristic for the software failure-occurrence or the software fault-detection phenomenon notably changes is called change-point [8].

This paper discusses a bivariate SRGM with the effect of a change of the software reliability growth factors for overcoming the problem mentioned above. Our bivariate SRGM enables us to describe a software reliability growth process depending on the testing-time and the testing-effort factors, and also enables us to consider the effect of the change of the software reliability growth factors at change-point. Further, we discuss an optimal release problem for deriving optimal testing-effort expenditures based on our bivariate SRGM. Finally, we show numerical examples for two-dimensional software reliability analysis and estimation of an optimal testing-effort expenditures based on our bivariate model by using actual software fault data.

2. Two-Dimensional Software Reliability Growth Modeling

A bivariate SRGM in which the number of detectable faults in a software system is assumed to be finite can be developed by the following modeling assumptions [5, 15,16]:

(A1) Whenever a software failure is observed, the fault is detected immediately, and no new faults are introduced in the fault-detection procedures.

(A2) Each software failure occurs at independently and identically distributed random times with the bivariate probability distribution function $F(s,u) \equiv \Pr\{S \le s, U \le u\}$, where S and U are the ran dom variables representing the testing-time and cumulative testing-effort expenditure, respectively. And $\Pr\{A\}$ represents the joint density function and the probability of event A, respectively.

(A3) The initial number of faults in the software system, N_0(> 0), is a random variable, and is finite.

Figure 1 shows the stochastic quantities for the two-dimensional software failure-occurrence or the software fault-detection phenomenon. Now we define the two-dimensional stochastic process $\{N(s,u), s \ge 0, u \ge 0\}$ [17] representing the number of faults detected in the two-dimensional space [0,s] × [0,u]. Then, we have a bivariate probability mass function that m faults are detected in the two-dimensional space [0,s] × [0,u] as

$$\Pr\{N(s,u) = m\} = \sum_{n} \binom{n}{m} \{F(s,u)\}^{m} \{1 - F(s,u)\}^{n-m} \Pr\{N_0 = n\} \quad (m = 0, 1, 2, \cdots) \tag{1}$$

from the modeling assumptions. From Equation. (1), we can say that the stochastic behavior of the software failure-occurrence or the software fault-detection phenomenon can be characterized by assuming the probability mass function for the initial number of faults in the software system, N_0. In this paper, we assume that the initial fault content follows a Poisson distribution with parameter ω. Then we have:

$$\Pr\{N(s,u) = m\} = \frac{\{\omega F(s,u)\}^{m}}{m!} \exp[-\omega F(s,u)] \tag{2}$$

from Equation (1). Equation (2) is essentially equivalent to a two-dimensional nonhomogeneous Poisson process (abbreviated as a two-dimensional NHPP) [5,6] with mean value function $E[N(s,u)] = \Lambda(s,u) = \omega F(s,u)$, where $E[\cdot]$ denotes the expectation.

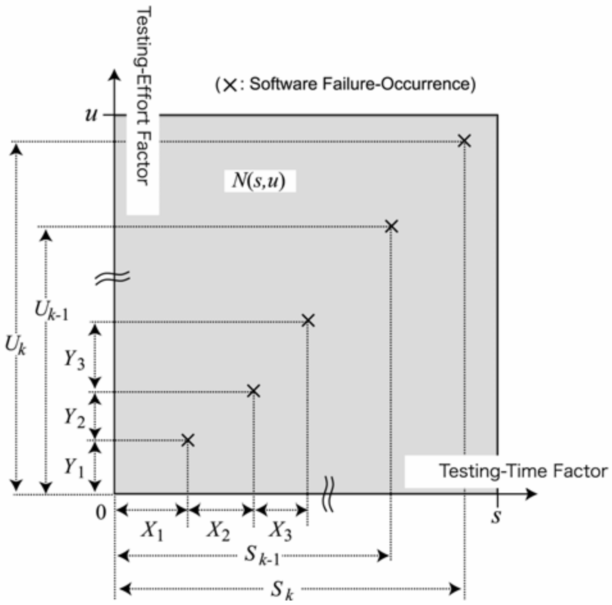

Figure 1. Stochastic quantities for the two-dimensional software failure-occurrence of fault-detection phenomenon

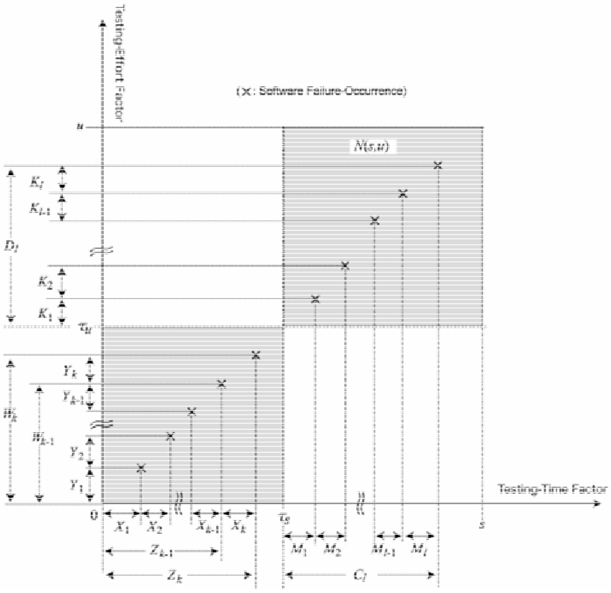

Figure 2. Two-dimensional stochastic quantities for the software failure-occurrence phenomenon with change –point

3. Two-Dimensional Change-Point Modeling

We discuss a bivariate software reliability growth modeling with effect of change-point on the software reliability growth process. In our research, we extend the assumption (A2) as that the stochastic characteristics of the software failure-occurrence phenomenon is changed at change-point. And the change-point occurs just only one time throughout testing-phase for the sake of simplification of this discussion.

Let us denote the change-point for the testing-time and the testing-effort factors by $\tau = \{\tau_s, \tau_u\}$ $(0<\tau_s<s_e, 0<\tau_u<u_e)$, where u_e represents the testing-effort expenditure expended up to the testing termination-time s_e. To start with, we define stochastic quantities for our bivariate software reliability growth modeling with effect of the change -point as shown in **Figure 2**. And we assume the following relationship between the software failure-occurrence time or time-interval before the change-point and those after the change-point:

$$\begin{cases} M_i = \alpha_s(X_i), & C_i = \alpha_s(Z_i), \\ K_i = \alpha_u(Y_i), & D_i = \alpha_u(W_i), \end{cases} \tag{3}$$

where $\alpha_s(\cdot)$ and $\alpha_u(\cdot)$ represent testing-environmental functions [18] for the testing-time and testing-effort factors, respectively. The testing-environmental function characterizes the relationship of the software failure -occurrence phenomenon before and after the change-point. Concretely, we assume that the relationship can be formulated as:

$$R = \alpha Q \tag{4}$$

in this paper. In Equation (4), R and Q represent column vectors, $R = (m\ k)^T$ $(m>0, k>0)$ and $Q = (x\ y)^T$ $(x>0, y>0)$, where the superscript T represents the transposed matrix and m, k, x, and y are the observed data for the random variables M, K, X, and Y in **Figure 2**, respectively. And α is the constant vector, $\alpha = (\alpha_s\ \alpha_u)$, which represents the relative magnitude of the effect of change-point on the software reliability growth process for each factor. Equation (4) is one of the examples for the testing-environmental function. However, we can get to know the effect of the change-point on the software reliability growth process simply by assuming Equation (4) as the testing-environmental function.

Now we suppose that $n(>0)$ faults have been detected up to change-point and their fault data from the test-beginning have been observed as (x_0, y_0), (x_1, y_1), $\cdots (x_n, y_n)$, where $x_{0=0}$, $y_{0=0}$, $0 < x_1 < x_2 \cdots < x_n \le \tau_s$, and $0 < y_1 < y_2 \cdots < y_n \le \tau_u$. Let us suppose that the number of faults detected in the testing-territory before change-point, $[0, s]\times[0, u](s \le \tau_s, u \le \tau_u)$, follows the two-dimensional NHPP in Equation (¥ref {mass3}) with mean value function $\Lambda_B(s, u)$. Then, the bivariate probability distribution function for the (M_1, K_1), can be derived by its cofunction. The cofunction is derived as the following conditional probability:

$$\begin{aligned} &\Pr\{M_1 > s, K_1 > u\} \\ &= \frac{\Pr\{X_{n+1} > \tau_s - x_n + s/\alpha_s, Y_{n+1} > \tau_u - y_n + u/\alpha_u\}}{\Pr\{X_{n+1} > \tau_s - x_n, Y_{n+1} > \tau_u - y_n\}} \end{aligned} \tag{5}$$

The denominator in Equation (5) can be derived as:

$$\begin{aligned} &\Pr\{X_{n+1} > \tau_s - x_n, Y_{n+1} > \tau_u - y_n\} \\ &= \exp\left[-\Lambda_B(\tau_s,\tau_u) + \Lambda_B(x_n,\tau_u) + \Lambda_B(\tau_s,y_n) - \Lambda_B(x_n,y_n)\right] \end{aligned} \tag{6}$$

based on the property of the two-dimensional NHPP. And also, we have the numerator as:

$$\begin{aligned} &\Pr\{X_{n+1} > \tau_s - x_n + s/\alpha_s, Y_{n+1} > \tau_u - y_n + u/\alpha_u\} \\ &= \exp\left[-\Lambda_B(\tau_s + s/\alpha_s, \tau_u + u/\alpha_u) - \Lambda_B(x_n, y_n)\right. \\ &\quad + \Lambda_B(x_n,\tau_u) + \Lambda_B(\tau_s, y_n) + \Lambda_B(\tau_s, \tau_u + u/\alpha_u) \\ &\quad \left. + \Lambda_B(\tau_s + s/\alpha_s, \tau_u) - 2\Lambda_B(\tau_s,\tau_u)\right] \end{aligned} \tag{7}$$

From Equations (6) and (7), Equation (5) can be written as:

$$\begin{aligned} &\Pr\{M_1 > s, K_1 > u\} \\ &= \exp\left[-\Lambda_B\left(\tau_s + s/\alpha_s, \tau_u + u/\alpha_u\right) + \Lambda_B\left(\tau_s, \tau_u + u/\alpha_u\right)\right. \\ &\quad \left. + \Lambda_B\left(\tau_s + s/\alpha_u, \tau_u\right) - \Lambda_B\left(\tau_s, \tau_u\right)\right] \end{aligned} \tag{8}$$

From Equation (8), the expected number of faults detected after the change-point, $\Lambda_A(s, u)$, is derived as:

$$\begin{aligned} \Lambda_A(s,u) &= -\log\left[\Pr\{M_1 > s - \tau_s, K_1 > u - \tau_u\}\right] \\ &= \Lambda_B\left(\tau_s + \frac{s-\tau_s}{\alpha_s}, \tau_u + \frac{s-\tau_u}{\alpha_u}\right) - \Lambda_B\left(\tau_s, \tau_u + \frac{u-\tau_u}{\alpha_u}\right) \\ &\quad - \Lambda_B\left(\tau_s + \frac{s-\tau_s}{\alpha_s}, \tau_u\right) + \Lambda_B\left(\tau_s, \tau_u\right) \end{aligned} \tag{9}$$

Accordingly, we have a mean value function with effect of change-point as:

$$\Lambda(s,u) \begin{cases} \Lambda_1(s,u) = \Lambda_B(s,u) \quad (\text{for} \quad 0 \le s \le \tau_s, 0 \le u \le \tau_u) \\ \Lambda_2(s,u) = \Lambda_B(\tau_s,\tau_u) + \Lambda_A(s,u) \\ \qquad = \Lambda_B\left(\tau_s + \dfrac{s-\tau_s}{\alpha_s}, \tau_u + \dfrac{u-\tau_u}{\alpha_u}\right) \\ \qquad -\Lambda_B\left(\tau_s, \tau_u + \dfrac{u-\tau_u}{\alpha_u}\right) \\ \qquad -\Lambda_B\left(\tau_s + \dfrac{s-\tau_s}{\alpha_s}, \tau_u\right) + 2\Lambda_B(\tau_s,\tau_u) \\ \quad (\text{for} \quad s > \tau_s, u > \tau_u) \end{cases} \tag{10}$$

Equation (10) imply that our bivariate change-point model for software reliability assessment can be developed by assuming the two-dimensional mean value function before change-point, $\Lambda_B(s, u)$.

4. Software Reliability Assessment Measure

Software reliability assessment measures are derived by stochastic properties of the SRGM, and play an important role in quantitative software reliability assessment based on the SRGM. An operational software reliability [5] is defined as the probability that a software failure does not occur in the time-interval $(s_e, s_e + \eta](s_e \geq 0, \eta \geq 0)$ given that the testing has been going up to testing-time s_e and the testing-effort has been expended up to u_e by testing-time s_e. From the basic notion of this measure and the stochastic properties in Equation (1), we can generally formulate the operational software reliability as:

$$\begin{aligned} & R(\eta \mid s_e, u_e) \\ & = \sum_k \Pr\{N(s_e+\eta, u_e) = k \mid N(s_e, u_e) = k\} \\ & \qquad \cdot \Pr\{N(s_e, u_e) = k\} \\ & = \sum_k \Big[\{F(s_e, u_e)\}^k \{1 - F(s_e+\eta, u_e)\}^{-k} \\ & \quad \cdot \sum_n \binom{n}{k} \{1 - F(s_e+\eta, u_e)\}^n \cdot \Pr\{N_0 = n\} \Big] \end{aligned} \tag{11}$$

Assuming that N_0 follows a Poisson distribution with parameter ω, we can derive the operational software reliability function as:

$$R(\eta \mid s_e, u_e) = \exp\left[-\{\Lambda(s_e+\eta, u_e) - \Lambda(s_e, u_e)\}\right] \tag{12}$$

by using Equation (11).

5. Parameter Estimation

Parameters of our bivariate model with change-point for software reliability assessment can be estimated by the method of maximum-likelihood. Suppose that we have observed K data pairs $(s_k, u_k, y_k)\,(k = 0,1,2,\cdots K)$ with respect to the number of fault y_k, which have been detected in the testing-territory $[0, \tau_s]\times[0, \tau_u]+(\tau_s, s_k]\times(\tau_u, u_k]$. The logarithmic likelihood function, $\ln L(\theta, \alpha \mid \tau)$, for the two-dimensional stochastic process $\{N(s,u), s \geq 0, u \geq 0\}$ given the change-point τ can be derived as:

$$\begin{aligned} \ln L(\theta, \alpha \mid \tau) = & \sum_{k=1}^{K} (y_k - y_{k-1}) \cdot \ln\big[\Lambda(s_k, u_k; \theta, \alpha \mid \tau) \\ & - \Lambda(s_{k-1}, u_{k-1}; \theta, \alpha \mid \tau)\big] - \Lambda(s_K, u_K; \theta, \alpha \mid \tau) \\ & - \sum_{k=1}^{K} \ln\big[(y_k - y_{k-1})!\big] \end{aligned} \tag{13}$$

where θ represents the set of the parameter in $\Lambda(s, u)$. Then, we obtain the following equation:

$$\frac{\partial \ln L(\theta, \alpha \mid \tau)}{\partial \theta} = \frac{\partial \ln L(\theta, \alpha \mid \tau)}{\partial \alpha} = 0 \tag{14}$$

The maximum-likelihood estimates are obtained by solving Equation (14) numerically.

6. Optimal Software Release Problem

If debugging cost before change-point and its after change-point a different each other, a trade-off problem between the effect of the change-point on the software reliability growth process and the related cost arises in a conventional optimal software release problem [19]. This paper discusses an optimal problem for estimating optimal shipping time and change-point of a software system based on our model proposed in this paper.

Now, we define cost parameter for formulating the expected total software cost with change-point as follows:

c_1: debugging cost for one fault before the change-point in the testing-phase ($c_1 > 0$),

c_2: debugging cost for one fault after the change-point in the testing-phase ($c_2 > 0$),

c_3: debugging cost for one fault in the operational phase ($c_1 < c_3$, $c_2 < c_3$)

c_4: testing cost per arbitrary testing-time and testing-effort ($c_4 > 0$).

Letting S, U, β_s, and β_u be the termination time of testing, the total testing-effort expended up to the termination time of the testing, the time duration from τ_s to S, and the testing-effort expended from τ_u to U, we have the expected total software cost, $C(S, U, \beta_s, \beta_u)$, as:

$$\begin{aligned} C(S, U, \beta_s, \beta_u) = & c_1 \Lambda_1(S - \beta_s, U - \beta_u) \\ & + c_2 \{\Lambda_2(S, U) - \Lambda_1(S - \beta_s, U - \beta_u)\} \\ & + c_3 \{\omega - \Lambda_2(S, U)\} + c_4 SU \end{aligned} \tag{15}$$

Generally, S^*, U^*, β_s^*, and β_u^*, which are the cost-optimal release time, the optimal testing-effort expenditures, the optimal time duration from τ_s to S^*, and the optimal testing-effort expended from τ_u to U^*, are derived by minimizing the expected total software cost, $C(S, U, \beta_s, \beta_u)$. Therefore, S^*, U^*, β_s^*, and β_u^*, are obtained by solving the following equations simultaneously:

$$\begin{aligned} & \frac{\partial C(S, U, \beta_s, \beta_u)}{\partial S} = \frac{\partial C(S, U, \beta_s, \beta_u)}{\partial U} \\ & \frac{\partial C(S, U, \beta_s, \beta_u)}{\partial \beta_s} = \frac{\partial C(S, U, \beta_s, \beta_u)}{\partial \beta_u} = 0 \end{aligned} \tag{16}$$

We should note that Equation (16) is a necessary condition for the optimal solutions. Equations (15) and (16) can be simplified to an optimal testing-effort expending problem for estimating optimal testing-effort expendi-

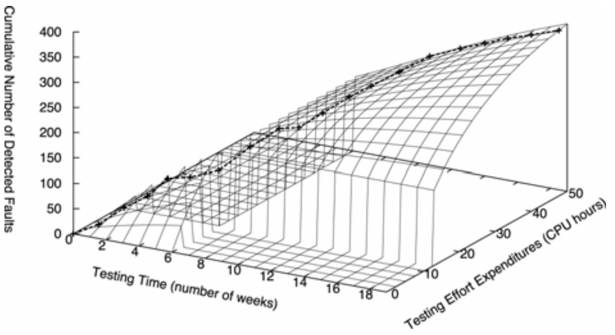

Figure 3. Estimated two-dimensional mean value function with effect of change-point, $\tau_s = 6, \tau_u = 12.89$

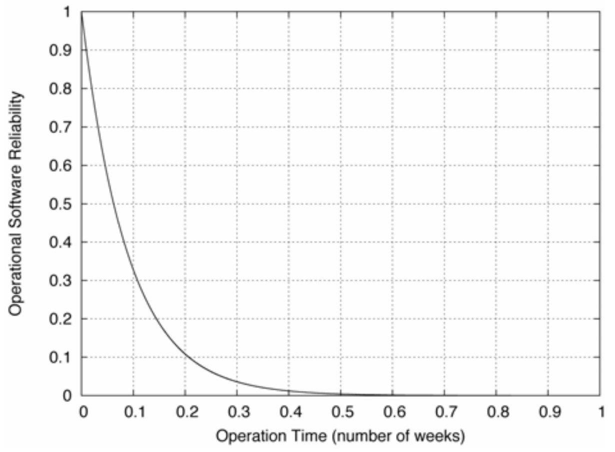

Figure 4. Estimated operational software reliability, $\hat{R}(\eta\,|\,19, 47.65)$

tures in the restricted testing-time. The expected total software cost for the optimal testing-effort expending problem is derived as:

$$C\left(\bar{S},U,\bar{\beta}_s,\bar{\beta}_u\right) = c_5\Lambda_2\left(\bar{S},U\right)+c_3\left\{\omega-\Lambda_2\left(\bar{S},U\right)\right\}+c_4\bar{S}U \tag{17}$$

where $\bar{S}$, $\bar{\beta}_s$, and $\bar{\beta}_u$ represent the given testing-time duration, β_s, and β_u, respectively. And c_5 is the debugging cost for one fault in the testing-phase, in which we assume that there is no difference between the debugging cost for one fault before and after the change -point. From the basic notion in Equation (16), the optimal testing-effort expenditures U^* needs to satisfy the following equation:

$$\frac{\partial C(\bar{S},U,\bar{\beta}_s,\bar{\beta}_u)}{\partial U} = \frac{c_2-c_1}{c_3}\left[\frac{c_3}{c_2-c_1}\bar{S}-h_2(\bar{S},U)\right]=0 \tag{18}$$

where $h_2\left(\bar{S},U\right)=\partial\Lambda_2\left(\bar{S},U\right)\big/\partial U$.

7. Numerical Examples

We show numerical examples of our bivariate model and its application to an optimal software release problem. In this paper, we use actual data consisted of 19 data pairs: $(s_k, u_k, y_k)(k = 0, 1, 2,\cdots,19;\ t_{19} = 19(weeks),\ s_{19} = 47.65$(CPU *hours*, $y_{19} = 328)$ [20]. And we assume that the software failure-occurrence time distribution before change-point follows the following bivariate probability distribution function [21]:

$$F\left(s,u\right)=\left(1-\mathrm{e}^{-as}\right)\left(1-\mathrm{e}^{-bs}\right)\left(1+z\mathrm{e}^{-as-bu}\right)$$
$$\left(a>0, b>0, -1\le z\le 1\right) \tag{19}$$

And we set $\alpha_s = \alpha_u = \alpha$ for simplification and we also set $\boldsymbol{\tau}=\left\{\tau_s,\tau_u\right\}=\left\{6,12.89\right\}$ by following the actual behavior of the software failure-occurrence phenomenon. Then, we estimate the parameters as $\hat{\omega}$, $\hat{a}$, $\hat{b}$, $\hat{z}$, and $\hat{\alpha}$, which are the estimates of ω, a, b, z, and α by the method of maximum likelihood discussed in Section 5.

Figure 3 shows the two-dimensional behavior of the estimated mean value function with effect of the change-point, where $\tau_s = 6$, $\tau_u = 12.89$. In **Figure 3**, the dotted line represents the actual behavior of the cumulative number of detected faults and the curved surface the estimated behavior. We see that the expected number of faults is estimated to be zero outside the software failure-occurrence territory, which has been explained in **Figure 2**. This is one of the feature for our two -dimensional SRGM with the effect of the change-point. Further, **Figure 4** shows the estimated operational software reliability, $\hat{R}(\eta\,|\,19, 47.65)$. From **Figure 4**, we can estimate the operational software reliability at 0.3 weeks after the termination time of the testing, $\hat{R}(0.3\,|\,19, 47.65)$, to be about 0.036.

Then, we show numerical examples for an optimal release problem for deriving optimal testing-effort expenditure based on our bivariate change-point model. This problem is one of the simplified problem for our optimal software release problem. In this problem, $h_2(\bar{S},U)$ in Equation (19) can be derived as:

$$h_2\left(\bar{S},U\right)=\frac{\partial\Lambda_2\left(\bar{S},U\right)}{\partial U} = -\frac{\omega b}{\alpha_u}\exp\left[-bD_1\left(U\right)\right]\left\{B\exp\left[-bD_1\left(U\right)\right]-A\right\} \tag{20}$$

where $A=2z\exp[-aD_2]-z\exp[-2aD_2]-(z+1)\exp[-a\tau_s] +z\exp[-2a\tau_s]$, $B=-2z\exp[-aD_2]+2z\exp[-2aD_2] +2z\exp[-a\tau_s]-2z\exp[-2a\tau_s]$, $D_1(U)=\tau_u+\dfrac{U-\tau_u}{\alpha_u}$, respectively. We can easily see that U^* can be obtained by solving the following equations:

$$\frac{c_4}{c_3-c_5}\bar{S}=h_2\left(\bar{S},U\right) \tag{21}$$

Table 1. Sensitivity analysis of the optimal testing-effort expenditures.

c_5	c_3	c_4	U^*	$C(\bar{S},U^*,\bar{\beta}_s,\bar{\beta}_u)$
1	5	5	3.2501	5117.1
1	5	3	10.856	4021.2
1	5	1	28.840	2925.3
1	10	1	43.318	5029.7
1	20	1	57.253	9238.6
1	40	1	70.975	17656
1	80	1	84.606	34492

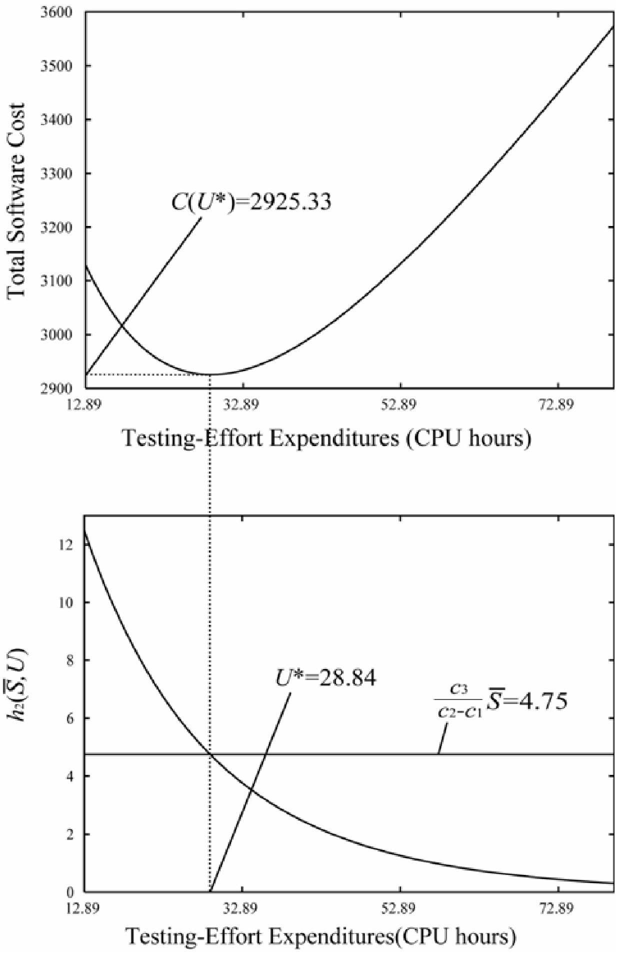

Figure 5. Time-dependent behavior of the estimated expected total software cost and $\hat{h}_2(\bar{S},U)$. (τ_s = 6, τ_u = 12.89, c_5 = 1, c_3 = 5, and c_4 = 1).

Table 1 shows the results of the sensitivity analysis of the optimal testing-effort expenditures. From **Table 1**, we can say that the optimal testing-effort expenditures are getting increased as the debugging cost for one faults in the operational phase is increased. On the other hand, the optimal testing-effort expenditures are also getting increased as the testing cost per arbitrary testing-time and testing-effort is decreased. Further, **Figure 5** shows the computed results for the time-dependent behavior of the expected total software cost and $\hat{h}_2(\bar{S},U)$, where c_5 = 1, c_3 = 5, and c_4 = 1. From **Figure 5**, we can estimate the optimal testing-effort expenditures, U^*, and the expected total software cost, $C(\bar{S},U^*,\bar{\beta}_s,\bar{\beta}_u)$ to be about 28.84 (CPU hours) and 2925.33, respectively.

8. Concluding Remarks

This paper discussed a bivariate software reliability growth modeling with the effect of the change-point and an optimal software release problem based on our model. Further, we showed numerical examples of software reliability analysis and an optimal testing-effort expending problem based on our model by using actual data. Our bivariate SRGM with the effect of change-point is expected to contribute high accuracy assessment of software reliability in a testing-phase. In the further studies, we have to investigate the effectiveness and validity of our model by using a lot of data sets collected from actual software development projects, and have to give a discussion on sufficient conditions for the optimal software release problem.

9. Acknowledgement

This work was supported in part by the Grant-in-Aid for Scientific Research (C), Grant No. 22510150, from the Ministry of Education, Sports, Science, and Technology of Japan.

10. References

[1] J.D. Musa, "A theory of software reliability and its application," *IEEE Transactions on Software Engineering*, Vol. SE-1, No. 3, 1975, pp. 312-327.

[1] S. Yamada and S. Osaki, "Software reliability growth modeling: Models and applications," *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 12, 1985, pp. 1431-1437. doi:10.1109/TSE.1985.232179

[2] J.D. Musa, D. Iannio, and K. Okumoto, "*Software Reliability: Measurement, Prediction, Application*," McGraw-Hill, 1987.

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

[4] T. Ishii and T. Dohi, "Two-dimensional software reliability models and their application," *Proceedings of the 12th Pacific Rim International Symposium on Dependable Computing*, 2006, pp. 3-10.

[5] S. Inoue and S. Yamada, "A bivariate Weibull-type software reliability growth model and its goodness-of-fit evaluation," (in Japanese) J. Information Processing Society of Japan, Vol. 49, no. 8, August 2008, pp. 2851

-2861.

[6] S. Inoue and S. Yamada, "A framework for two-dimensional software reliability modeling with program size," *Proceedings of the 14th ISSAT International Conference on Reliability and Quality in Design*, 2008, pp. 198-202.

[7] M. Zhao, "Change-point problems in software and hardware reliability," Communications in Statistics -- Theory and Methods, Vol. 22, No. 3, 1993, pp. 757-768. doi:10.1080/03610929308831053

[8] C.Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point," *Journal of Systems and Software*, Vol. 76, No. 2, 2005, pp. 181-194. doi:10.1016/j.jss.2004.04.024

[9] C.Y. Huang, "Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency," *Journal of Systems and Software*, Vol. 77, No. 2, 2005, pp. 139-155. doi:10.1016/j.jss.2004.10.014

[10] F.Z. Zou, "A change-point perspective on the software failure process," *Software Testing, Verification and Reliability*, vol. 13, no. 2, 2003, pp. 85-93. doi:10.1002/stvr.268

[11] J. Zhao, H.W. Liu, G. Cui, and X.Z. Yang, "Software reliability growth model with *change-point and environmental function," Journal of Systems and Software,* Vol. 79, No. 11, 2006, pp. 1578-1587. doi:10.1016/j.jss.2006.02.030

[12] S. Inoue and S. Yamada, "Software reliability measurement with change-point," Proceedings of the International Conference on Quality and Reliability, 2007, pp. 170 -175.

[13] S. Inoue and S. Yamada, "Optimal software release policy with change-point," Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, 2008, pp. 531-535. doi:10.1109/IEEM.2008.4737925

[14] N. Langberg and N.D. Singpurwalla, "A unification of some software reliability models," *SIAM Journal on Scientific Computing*, Vol. 6, No. 3, 1985, pp. 781-790. doi:10.1137/0906053

[15] D.S. Miller, "Exponential order statistic models of software reliability growth," *IEEE Transactions on Software Engineering*, Vol. SE-12, No. 1, 1986, pp. 12-24.

[16] D.N.P. Murthy, J. Baik, R.J. Wilson, and M.R. Bulmer, "Two-dimensional failure modeling," In: H. Pham, Ed., *Springer Handbook of Engineering Statistics*, Springer -Verlag, Berlin, 2006, pp. 97-111. doi:10.1007/978-1-84628-288-1_5

[17] H. Okamura, T. Dohi, and S. Osaki, "A reliability assessment method for software products in operational phase? Proposal of an accelerated life testing model," (in Japanese), *Transactions of IEICE*, Vol. J83-A, No. 3, 2000, pp. 294-301.

[18] S. Yamada and S. Osaki, "Cost-reliability optimal release policies for software systems," *IEEE Transactions on Reliability*, Vol. R-34, No. 5, 1985, pp. 422-424. doi:10.1109/TR.1985.5222222

[19] M. Ohba, "Software reliability analysis models," *IBM Journal of Research and Development*, Vol. 28, No. 4, 1984, pp. 428-443. doi:10.1147/rd.284.0428

[20] E.J. Gumbel, "Bivariate exponential distributions," *Journal of the American Statistical Association*, No. 55, 1960, pp. 698-707. doi:10.2307/2281591