B-023

Software Reliability Measurement with Effect of Change-Point by Using Environmental Function

Shinji Inoue† and Shigeru Yamada†

1. Introduction

Quantitative assessment of software reliability in an actual testing-phase in a software development process is one of the important activities for developing highly-reliable software systems. Software reliability growth model (abbreviated as SRGM) [1-3] is one of the useful mathematical tools for quantitative assessment of software reliability. Basically, SRGMs are developed by treating the software failure-occurrence time-interval as random variables and under the assumption that the stochastic characteristics of these random variables are the same throughout testing-phase. However, it is very difficult to say that this modeling assumption does not enable us to reflect a practical software failure-occurrence phenomenon because we often observe a phenomenon that the stochastic behavior of the software failure-occurrence time-interval notably changes due to a change of testing-environment on testing-activities, e.g., a change of fault target, a change of testing-effort expenditure, and so forth. Testing-time when such phenomenon is observed is called change-point [4]. It is known that a occurrence of the change-point influences accuracy of SRGM-based software reliability assessment. Under the background, software reliability growth modeling with the effect of change-point has been discussed so far [4-7]. In recent years, Inoue and Yamada [8,9] proposed software reliability growth modeling framework with the effect of change-point, and discussed a software management issue for deriving optimal shipping time and change-point from the point of view of cost minimization. However, it is very difficult to find research results, which discuss software reliability growth modeling with a relationship between the software failure-occurrence time intervals before change-point and those after change-point. In an actual testing-phase, it might be natural to consider that there exists a relationship between the time-intervals before the change-point and those after the change-point because a same software product is tested. And it is very important to know how the stochastic characteristic of the software failure-occurrence time-interval changes at the change-point from the point of view of software development management.

† Department of Social Management Engineering, Graduate School of Engineering, Tottori University.

In this paper, we discuss a new change-point modeling framework for software reliability assessment, in which we incorporate the relationship between the software failure-occurrence time intervals before change-point and those after change-point. Concretely, we use an environmental function for describing the relationship of the software failure-occurrence time-intervals before and after the change-point. Further, we check performance on software reliability assessment based on our model, which are developed based on our modeling framework in terms of a statistical goodness-of-fit test and mean square errors, and show examples of the application of our model for software reliability assessment by using actual change-point data.

2. Change-Point Modeling Framework

Our change-point modeling framework is developed based on the following basic modeling framework for nonhomogeneous Poisson process (NHPP)-based SRGMs in which the total number of detectable faults is assumed to be finite [2,10-12]:

(1) Whenever a software failure is observed, the fault which caused it will be detected immediately and no new faults are introduced in the fault-removing activities.

(2) Each software failure occurs at independently and identically distributed random times with the probability distribution, $F(t) \equiv \Pr\{T \le t\}$, where $\Pr\{A\}$ represents the probability of event A. And the probability density function is denoted by $f(t)$.

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

Now, let $\{N(t), t \ge 0\}$ denote a counting process representing the total number of faults detected up to testing-time t. From the basic assumptions above, the probability that m faults are detected up to testing-time t under the assumption that $N_0 = n$ is derived as

$$\Pr\{N(t)=m \mid N_0=n\} = \binom{n}{m}\{F(t)\}^m\{1-F(t)\}^{n-m}.$$

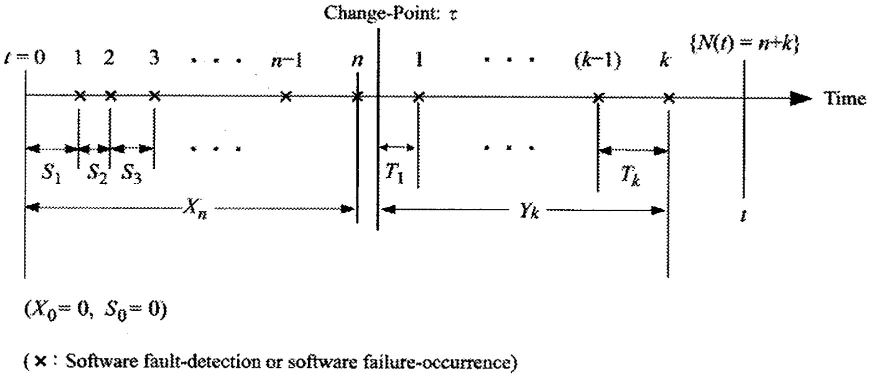

Figure 1 Stochastic quantities for software failure-occurrence or fault-detection phenomenon.

Then we have a probability mass function that m faults are detected up to testing-time t as

$$\begin{aligned}\Pr\{N(t)=m\} &= \sum_n \binom{n}{m}\{F(t)\}^m\{1-F(t)\}^{n-m}\frac{\omega^n}{n!}\exp[-\omega] \\ &= \frac{\{\omega F(t)\}^m}{m!}\exp[-\omega F(t)] \quad (m=0,1,2,\cdots),\end{aligned} \tag{1}$$

in which it is assumed that the initial fault content follows a Poisson distribution with mean ω. Eq. (1) is equivalent to an NHPP with mean value function:

$$E[N(t)] \equiv \Lambda(t) = \omega F(t).$$

Now, we define stochastic quantities being related to our change-point modeling framework in this paper as shown in Fig. 1. We assume that the stochastic quantities before and those after change-point have the following relationships:

$$\begin{cases} Y_i = \alpha(X_i) \\ T_i = \alpha(S_i) \\ J_i(t) = K_i(\alpha(t)), \end{cases} \tag{2}$$

respectively, where $\alpha(t)$ is a testing-environmental function representing the relationship between the stochastic quantities of the software failure-occurrence times or time-intervals before change-point and those after the change-point, $J_i(t)$ and $K_i(t)$ the probability distribution functions with respect to the random variables S_i and T_i, respectively. In this paper, we assume that the testing-environmental function is given as [13]

$$\alpha(t) = \alpha t \quad (\alpha > 0), \tag{3}$$

where α is the proportional constant representing the relative magnitude of the effect of change-point on the software reliability growth process. Eq. (3) 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 Eq. (3) as the testing-environmental function.

At this time we need to derive a probability distribution function of the first software failure-occurrence after the change-point to develop mean value function representing the number of fault detected after the change-point. Suppose that n faults have been detected up to the change-point and their fault-detection times from the test-beginning (t=0) have been observed as $0 < x_1 < x_2 < \cdots < x_n \le \tau$, where τ represents change-point. Then, the probability distribution function of T_1, a random variable representing the time duration from the change-point to the first software failure-occurrence after the change-point, can be derived as

$$\begin{aligned}\bar{J}_1(t) &\equiv \Pr\{T_1 > t\} \\ &= \frac{\Pr\{S_{n+1} > \tau - x_n + t/\alpha\}}{\Pr\{S_{n+1} > \tau - x_n\}} \\ &= \frac{\exp[-\{M_B(\tau + t/\alpha) - M_B(x_n)\}]}{\exp[-M_B(\tau) - M_B(x_n)]},\end{aligned} \tag{4}$$

where $\bar{J}_1(t)$ indicates the cofunction of the probability distribution function and $M_B(t)(\equiv \omega K_1(t))$ represents the expected number of faults detected up to change-point or a mean value function for the NHPP before

Table 1 Actual data sets.

	The number of data pairs	Unit of Measurement	Change-Point	Shape of Growth Curves
DS1	26	Day	18	S-shaped
DS2	29	Day	24	S-shaped
DS3	26	Day	17	Exponential
DS4	29	Day	18	Exponential
DS5	28	Day	18	Exponential

change-point. From Eq. (4), the expected number of faults detected up to $t \in (\tau, \infty)$ after change-point, $M_A(t)$, can be formulated as

$$\begin{aligned} M_A(t) &= -\log \Pr\{T_1 > t - \tau\} \\ &= -\log \bar{J}_1(t-\tau) \\ &= M_B(\tau + {}^{t-\tau}/_{\alpha}) - M_B(\tau). \end{aligned} \quad (5)$$

Then, the expected number of faults detected up to testing-time t $(t \in (\tau, \infty), 0 < \tau < t)$ can be derived as

$$\Lambda(t) = \begin{cases} \Lambda_B(t) = M_B(t) \quad (0 \le t \le \tau) \\ \Lambda_A(t) = M_B(\tau) + M_A(t) \\ \qquad = M_B(\tau + {}^{t-\tau}/_{\alpha}) \quad (\tau < t). \end{cases} \quad (6)$$

From Eq. (6), we can see that an NHPP-based SRGM with change-point can be developed by assuming a suitable probability distribution function for the software failure-occurrence time before change-point.

3. Goodness-of-Fit Evaluation

We conduct goodness-of-fit comparisons of our models with existing SRGMs, which do not incorporate the effect of the change-point. We arrange five data sets: DS1, DS2, DS3, DS4, and DS5. These actual data sets were collected in actual testing-phases for a web system and the change-point was generated by changing the tester and increasing the test personnel. Table 1 shows some information being related to these data.

In this paper, we develop change-point models based on our modeling framework by assuming two-types of mean value functions before change-point: exponential and delayed S-shaped SRGMs [1-3], which are the same meaning that we assume the software failure-occurrence times distribution before change-point follows an exponential distribution and a gamma distribution with k=2, $\Gamma(2)$, respectively. Then, we have the following mean value functions with the effect of change-point as

$$\Lambda(t) = \begin{cases} \Lambda_B(t) = a(1 - e^{-bt}) \quad (0 \le t \le \tau) \\ \Lambda_A(t) = a\{1 - \exp[-b(\tau + {}^{t-\tau}/_{\alpha})]\} \\ \qquad (t > \tau) \end{cases} \quad (7)$$

and

$$\Lambda(t) = \begin{cases} \Lambda_B(t) = a\{1 - (1 + bt)\exp[-bt]\} \\ \qquad (0 \le t \le \tau) \\ \Lambda_A(t) = a\{1 - (1 + b(\tau + {}^{t-\tau}/_{\alpha})) \\ \qquad \exp[-b(\tau + {}^{t-\tau}/_{\alpha})]\} \\ \qquad (t > \tau), \end{cases} \quad (8)$$

respectively. Parameters of our SRGMs can be estimated by using the method of maximum likelihood.

Now we investigate goodness-of-fit of our models statistically. This paper conducts Kolmogorov-Smirnov Test (K-S goodness-of-fit test) [3] for investigating goodness-of-fit of our models to the actual data. The K-S goodness-of-fit test can be conducted along with the following procedure. Suppose that we have observed n data pairs (t_i, y_i) $(i = 0,1,2,\cdots,n)$ with respect to the total faults, y_i, detected during constant time-interval $(0, t_i](0 < t_1 < t_2 < \cdots < t_n)$, the K-S test statistic, D, can be written as

$$\left.\begin{aligned} D &= \max_{i \le i \le n} D_i \\ D_i &= \max\left\{\left|{}^{H(t_i)}/_{H(t_n)} - {}^{y_i}/_{y_n}\right|, \left|{}^{H(t_i)}/_{H(t_n)} - {}^{y_{i-1}}/_{y_n}\right|\right\} \end{aligned}\right\}. \quad (9)$$

Table 2 The results of K-S goodness-of-fit.

	Proposed Model (Exponential SRGM-Based)	Existing Model (Exponential SRGM)	Proposed Model (Delayed S-Shaped SRGM-Based)	Existing Model (Delayed S-Shaped SRGM)
DS1	0.265227	0.216352*	0.1934*	0.177095*
DS2	0.227221*	0.27147	0.17183*	0.197832*
DS3	0.125986*	0.119922*	0.149066*	0.171058*
DS4	0.076802*	0.085945*	0.127143*	0.13292*
DS5	0.124526*	0.117522*	0.11127*	0.119403*

(*: 5% significant)

Table 3 The results of goodness-of-fit comparison based on MSE.

	Proposed Model (Exponential SRGM-Based)	Existing Model (Exponential SRGM)	Proposed Model (Delayed S-Shaped SRGM-Based)	Existing Model (Delayed S-Shaped SRGM)
DS1	18.4005	**16.0848**	7.77583	**7.21958**
DS2	**24.6323**	32.0834	22.1304	**19.3189**
DS3	**2.19263**	2.40089	**6.05874**	6.87789
DS4	**0.331126**	0.348901	**1.60806**	1.70385
DS5	**2.2943**	2.47239	**7.49945**	7.63319

The K-S test statistic D is needed to compared with a critical value $d_{n;\alpha}$, where n represents the number of data pairs and α a level of significance which is usually given as 0.01 or 0.05. Then, we judge that an NHPP model fits to the observed data at a level of significance α if $D < d_{n;\alpha}$. Table 2 shows the results of K-S goodness-of-fit test. From the results of K-S goodness-of-fit, our proposed models and corresponding existing models (exponential and delayed S-shaped SRGMs) mostly fits to the actual data sets statistically. However, we cannot make sure the evidence that our models have better performance than the existing models, which do not incorporate the effect of the change-point. And we conduct goodness-of-fit comparisons of our models in Eqs. (7) and (8) with corresponding existing SRGMs, which do not incorporate the effect of change-point, from the view point of mean square errors (MSE). The MSE is calculated as

$$MSE = \frac{1}{n}\sum_{k=1}^{n}[y_k - \hat{y}(t_k)]^2, \qquad (10)$$

where $\hat{y}(t_k)$ represents the estimated cumulative number of faults detected during a time-interval $(0, t_k]$. The smaller MSE represents that the model fits well to the actual data. Table 3 shows the results of goodness-of-fit comparisons based on the MSE. In Table 3, the bold-text represents having better performance compared with an existing corresponding model and the underlined bold-text represents having best performance for each data set. From Table 3, for a delayed S-shaped SRGM-based model, it is very difficult to find the effectiveness of considering with the effect of change-point. However, we can say that taking the effect of the change-point into consideration in software reliability growth modeling is one of the effective ways for improving the accuracy of software reliability assessment based on an SRGM.

4. Numerical Examples

We show numerical examples of the applications of our model in Eq. (7) for software reliability assessment by using actual data. In this paper, we use DS3, for which our proposed exponential SRGM-based model indicated best performance on goodness-

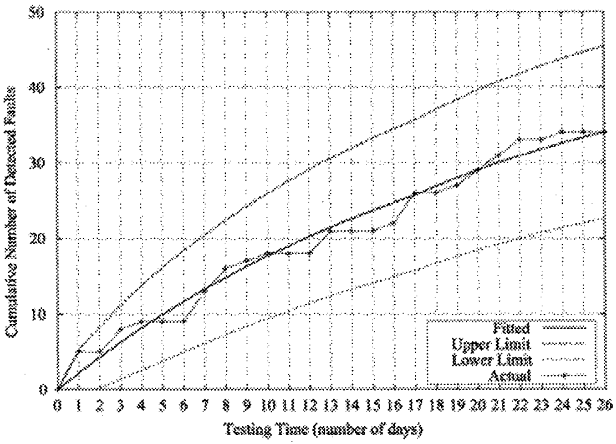

Figure 2 Estimated mean value function with change-point and its 95% confidence limits. (change-point=17 (days))

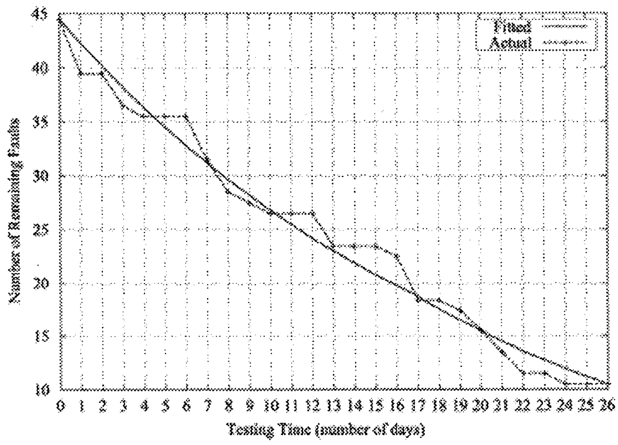

Figure 3 Estimated expected number of remaining faults. (change-point=17 (days))

-of-fit in the previous section.

To start with, we estimate parameters of our model, a, b, and α simultaneously based on the method of maximum likelihood given that the change-point $\tau = 17$. As the result of parameter estimation, we can get $\hat{a} = 44.482$, $\hat{b} = 0.0506$, and $\hat{\alpha} = 0.781$, which are the estimated parameter of a, b, and α, respectively. Based on the parameter estimates, Figure 2 shows the estimated mean value function with the effect of the change-point in Eq. (7) and its 95% confidence limits. From Fig. 2, we can see that the time-dependent behavior of the estimated number of detected faults changes at the change-point, and fits well to the actual behavior. And we can say that the testing-environment after the change-point is harder than that before the change-point since $\hat{\alpha} = 0.781$.

Further, we show numerical examples for software reliability assessment measures based on our model in Eq. (7), such as the expected number of remaining fault, software reliability function, and a cumulative MTBF. The expected number of remaining faults indicates the time-dependent behavior for the expected number of undetected faults in the software system. Then the expected number of remaining faults, $M(t)$, is derived as

$$M(t) \equiv E[N(\infty) - N(t)] = \Lambda(\infty) - \Lambda(t) = a - \Lambda(t). \quad (11)$$

Figure 3 shows the estimated time-dependent behavior for the expected number of remaining faults. From Fig. 3, we can estimate the residual fault content at the termination time of the testing, $\hat{M}(26)$, to be about 10 faults. The software reliability function represents the probability that a software failure does not occur in the time-interval $(t, t+x](t \geq 0, x \geq 0)$ given that the testing or the user operation has been going up to time t. Then, the software reliability function is derived as

$$R(x \mid t) = \exp[-\{\Lambda(t+x) - \Lambda(t)\}] \quad (12)$$

if the counting process $\{N(t), t \geq 0\}$ follows the NHPP with mean value function $\Lambda(t)$. We should note that Eq. (12) is derived under the condition that the software system is operated in the same environment as the testing-phase after the change-point. Figure 4 shows the estimated software reliability $\hat{R}(x \mid 26)$. From Fig. 4, we can estimate $\hat{R}(1.0 \mid 26) \approx 0.5176$ under the assumption that the software is operated in the same environment as the testing-phase after the change-point. Finally, we discuss a cumulative mean time between software failures (cumulative MTBF), which is one of the substitute measures of the usual MTBF for the NHPP model, in which the number of detectable faults in the software is assumed to be finite. The cumulative MTBF as defined as follows:

$$MTBF_C(t) = \frac{t}{\Lambda(t)}. \quad (13)$$

Figure 5 depicts the time-dependent behavior of the estimated cumulative MTBF. From Fig. 5, we can estimate the cumulative MTBF at the termination time of the testing to be about 0.7647 (days).

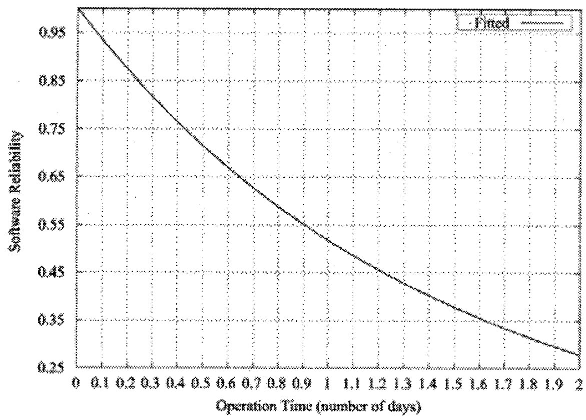

Figure 4 Estimated software reliability function.

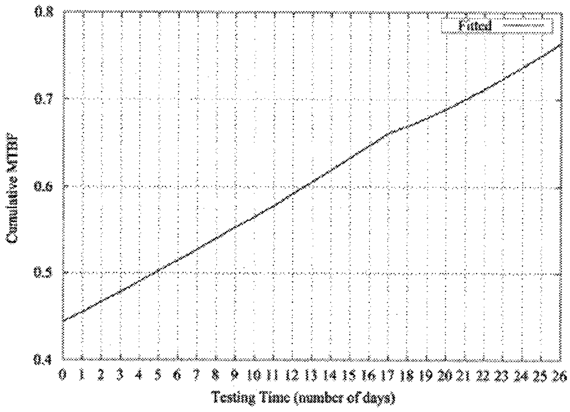

Figure 5 Estimated cumulative MTBF. (change-point=17 (days))

5 . Conclusion

We discussed change-point modeling framework for software reliability assessment, in which the relationship between software failure-occurrence times before change-point and those after the change-point was incorporated by using an testing-environmental function. In our modeling approach, we can easily develop a change-point model by assuming a mean value function before the change-point. And this paper investigated performance on software reliability assessment based on our two-types specific change-point models, which were developed by assuming mean value functions before change-point follow exponential and delayed S-shaped SRGMs, by using K-S goodness-of-fit-test and MSE. Further, by comparing of our models with corresponding existing SRGMs, which do not incorporate the effect of change-point, we can see that taking the effect of the change-point into consideration in software reliability growth modeling is one of the effective method for improving accuracy of quantitative software reliability assessment based on SRGM.

We need to investigate more the effectiveness of our modeling approach by comparing existing change-point models. And, we are going to develop a more feasible testing-environmental function for getting higher accuracy for software reliability assessment based on an SRGM developed under our modeling framework.

Acknowledgement

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

References

[1] S. Yamada and S. Osaki, "Software reliability growth modeling: Models and applications," *IEEE Trans. Soft. Eng.* vol. 11, no. 12, pp. 1431-1437, 1985.

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

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

[4] M. Zhao, "Change-point problems in software hardware reliability," *Commun. Statist. ---Theory Meth.*, vol. 22, no. 3, pp. 757-768.

[5] C.Y. Huang, "Performance analysis of software reliability growth models with testing-effort and change-point," *J. Sys. Soft.*, vol. 76, no. 2, pp. 181-194, 2005.

[6] F.Z. Zou, "A change-point perspective on the software failure process," *Softw. Test., Verif. Reliab.*, vol. 13, no. 2, pp. 85-93, 2003.

[7] J. Zhao, H.W. Liu, G. Cui, and X.Z. Yang, "Software reliability growth model with change-point and environmental function," *J. Sys. Soft.*, vol. 79, no. 11, pp. 1578-1587, 2006.

[8] S. Inoue and S. Yamada, "Software reliability measurement with change-point," *Proc. Intern. Conf. Qual. Reliab.* (ICQR 2007), pp. 170-175, 2007.

[9] S. Inoue and S. Yamada, "Optimal software release policy with change-point," *Proc. IEEE Intern. Conf. Indust. Eng. Eng. Mana.* (IEEM 2008), pp. 531-535, 2008.

[10] D.S. Miller, "Exponential order statistic models of software reliability growth," *IEEE Trans. Soft. Eng.*, vol. SE-12, no. 1, pp. 12-24, 1986.

[11] A.E. Raftery, "Inference and prediction for a general order statistic model with unknown population size," J. ASA., vol. 82, no. 400, pp. 1163-1168, 1987.

[12] H. Joe, "Statistical inference for general-order-statistics and nonhomogeneous-Poisson-process software reliability models," IEEE Trans. Soft. Eng., vol. 15, no. 11, pp. 1485-1490, 1989.

[13] H. Okamura, D. Dohi, and S. Osaki, "A reliability assessment method for software products in operational phase? --- Proposal of an accelerated life testing model," (in Japanese), *Trans. IEICE*, J83-A(3), pp. 294-301.