Change-point modeling with difference in software failure-occurrence time-interval for reliability assessment

Shinji Inoue¹ and Shigeru Yamada¹

¹ Department of Social Management Engineering, Graduate School of Engineering, Tottori University, 4-101 Minami, Koyama, Tottori, Tottori 680-8552, Japan
E-mail: {ino, yamada}@sse.tottori-u.ac.jp

accepted July 17, 2010

Summary
This paper discusses change-point modeling for software reliability assessment with a relationship of the software failure-occurrence time-interval before and after change point by using a testing-environmental function. In an actual testing phase of a software development process, it might be better to consider that there exists a relationship between the time intervals before change-point and those after change-point because the software product is not changed during the testing phase. Our modeling framework enables us to develop a software reliability growth model with effect of change-point by giving a suitable software failure-occurrence time distribution before change-point. Finally, we show examples of the application of software reliability analysis based on our model by using actual data.

Key words: Software reliability assessment, change-point, software reliability growth model, testing-environmental function, stochastic process.

1. INTRODUCTION

Software reliability growth models (SRGMs) [1-3] are known as mathematical tools for quantitative software reliability assessment. Ordinarily, SRGMs are developed by treating the software failure-occurrence time- or the fault-detection time-intervals as random variables. And, it is assumed that the stochastic characteristics for these quantities are same throughout the testing phase. However, this assumption does not enable us to reflect a practical software failure-occurrence phenomenon in software reliability growth modeling because we often observe a phenomenon that stochastic behavior of software failure-occurrence time interval notably changes due to a change of testing-environment during 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 the change-point influences accuracy of reliability assessment based on SRGMs.

Under the background, software reliability growth modeling with change-point has been discussing in recent year. Zhao [4] has been proposed change-point models by extending exponential, Weibull, and Pareto hazard rate models, respectively. Ohtera and Yamada [5] have been discussed a change-point occurrence mechanism from the view point of software development management. Huang [6,7] has discussed a change-point model with difference between instantaneous testing-effort expenditures before change-point and after change-point, and also discussed an optimal software release problem based on the change-point models. Zou [8] proposed a change-point model based on an exponential SRGM by considering difference of the fault-detection rate. Zhao et al. [9] proposed an SRGM by incorporating a environmental function representing the difference of the fault-detection rate. In recent years, a generalization approach for software reliability growth modeling with change-point [10] and an optimal change-point and software release policy [11], which derives an optimal change-point and software shipping time simultaneously from the view point of software development management, have been discussed.

However, it is 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 better to consider that there exists a relationship between the time intervals before change-point and those after change-point because the software product is not changed during the testing phase. This paper discusses a framework for software reliability growth modeling with the effect of change-point on a software reliability growth process by formulating the relationship between the software failure-occurrence time intervals before change-point and those after change-point. Our approach discussed in this paper basically follows a software reliability growth modeling technique for operational software reliability assessment, in which the software failure-intensity difference between the testing and the operational phases has been formulated by using a environmental function [12]. Finally, this paper shows numerical examples for software reliability assessment with change-point based on our modeling approach by using actual data.

2. BASIC MODELING FRAMEWORK

Basically, it is known that almost all the SRGMs in which the total number of detectable faults is finite are developed under the following basic assumptions [13-16]:

(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, \( F(t) \equiv \Pr(T \leq t) \), 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, \( N(0) \), is a random variable, and is finite.

Now, let \( \{N(t), t \geq 0\} \) 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 \( t \) is derived as
\[ \Pr(N(t) = m) = \sum_{n} \binom{n}{m} [F(t)]^n (1 - F(t))^{n-m} \Pr(N_0 = n) \quad (m = 0, 1, 2, \ldots). \] (1)

As a well-known result, if we assume that the initial fault content, \( N_0 \), follows a Poisson distribution with mean \( \omega \), the counting process \( \{N(t), t \geq 0\} \) in Eq. (1) can be rewritten as

\[
\Pr(N(t) = m) = \sum_{n} \binom{n}{m} [F(t)]^n (1 - F(t))^{n-m} \frac{\omega^n}{m!} \exp(-\omega) \\
= \exp(-\omega) \frac{\omega^m}{m!} \sum_{n} \frac{(\omega(1 - F(t)))^{n-m}}{(n-m)!} \\
= \frac{(\omega F(t))^m}{m!} \exp(-\omega F(t)) \quad (m = 0, 1, 2, \ldots). \tag{2}
\]

Eq. (2) is equivalent to a nonhomogeneous Poisson process (NHPP) with mean value function \( \omega F(t) \). We need to give a suitable software failure-occurrence times distribution to develop a specific NHPP model. For an example, we obtain an exponential SRGM \([17]\), \( \mathbb{E}[N(t)] = \omega(1 - e^{-\lambda t}) \), which is one of the representative NHPP models, if we assume that the software failure-occurrence times distribution follows an exponential distribution with parameter \( \lambda \).

### 3. MODELING WITH CHANGE-POINT

We point out a problem in software reliability growth modeling with change-point by discussing an existing software reliability growth modeling approach \([10]\). After that, we discuss our modeling approach in this paper. In our discussion, we assume that change-point occurs at most once throughout the testing, and the testing-termination time is denoted by \( T \).

#### 3.1 Existing Modeling Approach \([10]\)

We discuss an existing framework for software reliability growth modeling with change-point based on the basic modeling framework. Extending the assumptions on basic modeling framework (2), we describe a difference between the software hazard rates before and after the change-point as

\[
z(t) = \begin{cases} 
  z_1(t) & (0 \leq t \leq \tau) \\
  z_2(t) & (t > \tau),
\end{cases}
\]

where \( \tau (0 < \tau < T) \) indicates change-point. From Eq. (3), the software failure-occurrence times distributions before and after change-point is derived as

\[
F(t) = \begin{cases} 
  1 - \exp[-\int_{0}^{t} z_1(x) \, dx] & (0 \leq t \leq \tau) \\
  1 - \exp[-\int_{0}^{t} z_1(x) \, dx - \int_{\tau}^{t} z_2(x) \, dx] & (t > \tau),
\end{cases} \tag{4}
\]

respectively. In Eq. (4), the software failure-occurrence times distribution over \( (0 < t < \tau) \) is denoted as \( F_1(t) \) and over \( (\tau < t < T) \) as \( F_2(t) \), respectively. Then, we can develop a mean value function with change-point, \( H(t) \), as

\[
H(t) = \omega F(t) \\
= \omega(F_1(t)U_1(t - \tau) + F_2(t)U_2(t - \tau)), \tag{5}
\]
where \( U_1(x) \) and \( U_2(x) \) represent the step functions:
\[
U_1(x) = \begin{cases} 
0 & (x < 0) \\
1 & (x \geq 0), 
\end{cases} \quad U_2(x) = \begin{cases} 
0 & (x \leq 0) \\
1 & (x > 0), 
\end{cases}
\]
respectively. From Eqs. (5) and (6), we can develop an SRGM with change-point by assuming the software failure-occurrence rate functions for the software failure-occurrence times distributions before and after the change-point, respectively.

### 3.2 Proposed Modeling Approach

Existing modeling approach mentioned above is developed by focusing on the change of the software hazard rate at the change-point. However, this modeling approach has not been discussed the relationship between the software failure-occurrence time-intervals before and those after change-point. In an actual testing phase, it might be natural to consider that there exists such relationship since we test an identical software system in an actual testing phase even if change-point occurs.

Now we define the following stochastic quantities being related to our modeling approach in this paper:

- \( X_i \) = the \( i \)-th software failure-occurrence time before change-point \((X_0 = 0, i = 0,1,2,\cdots)\).
- \( S_i \) = the \( i \)-th software failure-occurrence time-interval before change-point \((S_i = X_i - X_{i-1}, S_0 = 0, i = 0,1,2,\cdots)\).
- \( Y_i \) = the \( i \)-th software failure-occurrence time after change-point \((Y_0 = 0, i = 0,1,2,\cdots)\).
- \( T_i \) = the \( i \)-th software failure-occurrence time-interval after change-point \((T_i = Y_i - Y_{i-1}, T_0 = 0, i = 0,1,2,\cdots)\).

Figure 1 depicts the stochastic quantities for the software failure-occurrence or fault-detection phenomenon with change-point. We assume that the stochastic quantities before and those after the change-point have the following relationships:
\[
\begin{align*}
Y_i &= \alpha(X_i), \\
T_i &= \alpha(S_i), \\
J_i(t) &= K_i(\alpha(t)),
\end{align*}
\]
respectively, where \( \alpha(\cdot) \) is a test-environmental function representing the relationship between the software failure-occurrence times or time-intervals before change-point and those after 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.

We assume that the test-environmental function is given as \( \alpha(t) = \alpha(t) (\alpha > 0) \) [12], where \( \alpha \) is the proportional constant representing the relative magnitude of the effect of change-point on the software reliability growth process. Suppose that \( n \) faults have been detected up to change-point and their fault-detection times from the test-beginning \((t = 0)\) have been observed as \( 0 < x_1 < x_2 < \cdots < x_n \leq \tau \). Then, the probability distribution function of \( T_i \), a random variable representing the time-interval from change point to the \((n+1)\)-st software failure-occurrence, can be derived as
The expected number of faults detected up to $t \in (\tau, \infty)$ after change-point, $M_A(t)$, can be formulated as

$$M_A(t) = -\log \Pr\{T_i > t - \tau\} = -\log \bar{J}_1(t - \tau) = M_B(\tau + \frac{t - \tau}{\alpha}) - M_B(\tau).$$

From Eq. (10), we can see that several types of NHPP-based SRGM with change-point can be developed by assuming a suitable probability distribution function for the software failure-occurrence times before change-point.

### 4. RELIABILITY ASSESSMENT MEASURES

We discuss software reliability assessment measures, such as the expected number of remaining faults, a software reliability function, and a cumulative mean time between software failures. The expected number of remaining faults is a software reliability assessment measure representing the expected number of faults remaining in the software at arbitrary testing-time $t$. Suppose a counting process, $\{N(t), t \leq 0\}$, follows the NHPP in Eq. (2), then the expected number of remaining faults is derived as
\[ M(t) = \mathbb{E}[N(t)] = \mathbb{E}[N(\infty) - N(t)] = \omega - \Lambda(t), \]  

(12)

where \( \Lambda(t) \) represents a mean value function of the NHPP.

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

\[ R(x | t) = \exp\left[-\left(\Lambda(t + x) - \Lambda(t)\right)\right] \]

(13)

if the counting process \( \{N(t), t \geq 0\} \) follows the NHPP with mean value function \( \Lambda(t) \).

Anyway, on well-known NHPP models, such as exponential and delayed S-shaped SRGMs, the usual mean time between software failures (MTBF) cannot be derived if an NHPP model is developed under the assumption that the number of detectable faults is finite. The reason why is that the cumulative distribution function indicating the probability that a software failure occurs during the time-interval \((t, t + x] (t \geq 0, x \geq 0)\),

\[ F(x | t) = 1 - R(x | t) = 1 - \exp\left[-\left(\Lambda(t + x) - \Lambda(t)\right)\right], \]

(14)

has the following properties:

\[
\begin{align*}
F(0 | t) &= 0 \\
F(\infty | t) &= 1 - \exp[-M(t)].
\end{align*}
\]

(15)

by using Eqs. (12) and (14). That is, \( F(x | t) \) does not satisfy the property of the usual cumulative distribution function. The cumulative MTBF is one of the substitution of the usual MTBF, and given as

\[ \text{MTBF}_c(t) = \frac{t}{\Lambda(t)}. \]

(16)

5. NUMERICAL EXAMPLES

We show numerical examples of our model by using actual fault count data. We use the following fault count data collected in an actual testing phase: \((t_k, y_k) (k = 0, 1, 2, \ldots, 19; t_{19} = 19 \text{ (weeks)}, y_{19} = 328) [18].\)

In this paper, we assume that the software failure-occurrence times distribution before change-point follows an exponential distribution with parameter \( b \). That is, the time-dependent behavior of the expected number of faults follows an exponential SRGM. At this time, \( \omega \) and \( \hat{b} \), which are the parameter estimations of \( \omega \) and \( b \), can be estimated by the method of maximum likelihood. The parameters \( a \) and \( \tau \) are given in this parameter estimation since these parameters are related to change-point and its effect on the software reliability growth process.

Suppose that we observed \( K \) data pairs \((t_k, y_k) (k = 0, 1, 2, \ldots, K)\) with respect to the total number of faults, \( y_k \), detected during a constant time-interval \((0, t_1] (t_0 < t_1 < \cdots < t_k)\). Model parameters can be estimated by using the method of maximum-likelihood. The logarithmic likelihood function, \( \ln L(\theta \mid \tau) \), for the stochastic process \( \{N(t), t \geq 0\} \) can be derived as
\[
\ln L(\theta | \alpha, \tau) = \sum_{k=1}^{K} (y_k - y_{k-1}) \ln[H(t_k; \theta | \alpha, \tau) - H(t_{k-1}; \theta | \alpha, \tau)] - H(t_k; \theta | \alpha, \tau) - \sum_{i=1}^{K} \ln[y_k - y_{k-1}],
\]

where \( L(\theta | \alpha, \tau) \) represents the likelihood function for the stochastic process \( \{N(i, t \geq 0)\} \), \( \theta \) a set of parameters in an SRGM with change-point. Then, we can obtain parameter estimates by solving numerically the simultaneous log-likelihood function: \( \frac{\partial \ln L(\theta | \alpha, \tau)}{\partial \theta} = 0 \) with respect to the model parameters, respectively.

We show numerical examples for software reliability assessment based on our model. Figure 2 shows the estimated mean value function, in which we set \( \alpha = 2.0 \) and \( \tau = 14 \) as a best fit case in terms of the mean square errors, and its 90% confidence limits. From Figure 2, we can say that our model fit well to the actual behavior by considering with the change of the software failure-occurrence phenomenon at change-point. By using the estimated mean value function, we can estimate the software reliability assessment measures. Figure 3 depicts the estimated software reliability function, \( \hat{R}(x|19) \), in which the software would be shipped at the termination time of testing and it is assumed that the software would be operated under the same environment as in the testing. From Figure 3, we can estimate the software reliability \( \hat{R}(0.5|19) \) to be about \( 1.4815 \times 10^{-2} \).

Additionally, Figure 4 shows the estimated cumulative MTBF. We can observe that the trend of the time-dependent behavior of the estimated cumulative MTBF is obviously changed at the change-point in Fig. 4. From Fig. 4, the cumulative MTBF at the termination of testing is estimated to be about \( 5.7927 \times 10^{-2} \) (weeks). We can see that the software reliability rapidly increases after change-point.

**Figure 2. Estimated mean value function and its 90% confidence limits (\( \alpha = 2.0 \) and \( \tau = 14 \))**

### 6. CONCLUDING REMARKS

This paper has pointed out problems on existing software reliability growth modeling with change-point, and discussed a new modeling approach with change-point to overcome such problem. Our modeling approach has focused the relationship between the software failure-occurrence time-intervals before change-point and those after change-point, and formulated such relationship by using an testing-environmental function. Such methodology enables us to figure out the effect of change-point on the software reliability growth process, and to develop several types of SRGMs.
with change-point based on our modeling framework in this paper. As we mentioned, we can develop an NHPP model by assuming the mean value function representing the time-dependent behavior of the expected number of fault before change-point in our modeling framework.

In further studies, we have to develop more plausible testing-environmental functions and to investigate how to estimate the parameter in the testing-environmental function. And we have to check performance on software reliability assessment based on SRGMs developed under our modeling approach by using many fault-count data collected in actual testing phases.

Figure 3. Estimated software reliability function \((\alpha = 2.0 \text{ and } \tau = 14)\)

Figure 4. Estimated cumulative MTBF \((\alpha = 2.0 \text{ and } \tau = 14)\)

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


