

Available online at www.sciencedirect.com



The Journal of Systems and Software

The Journal of Systems and Software 79 (2006) 1578-1587

www.elsevier.com/locate/jss

# Software reliability growth model with change-point and environmental function

Jing Zhao \*, Hong-Wei Liu, Gang Cui, Xiao-Zong Yang

School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, PR China

Received 20 June 2005; received in revised form 19 January 2006; accepted 12 February 2006 Available online 29 March 2006

#### Abstract

This paper presents a SRGM (software reliability growth model) with both change-point and environmental function based on NHPP (non-homogeneous Poisson process). Although a few research projects have been devoted to the change-point problems of SRGMs, consideration of the variation of environmental factors in the existing models during testing time is limited. The proposed model is one of a few NHPP models, which takes environmental factors as a function of testing time. FDR (fault detection rate) is usually used to measure the effectiveness of fault detection by test techniques and test cases. A FDR function after the change-point of the testing is proposed, which is computed from both environmental factors and FDR before the change-point of the testing. A NHPP SRGM with both change-point and environmental function called CE-SRGM is built which integrates the FDR before the change-point of the testing and the proposed FDR function after the change-point of the testing. The experimental results show that the predictive power of CE-SRGM is better than those of other SRGMs.

Keywords: Software reliability growth model; Non-homogeneous Poisson process (NHPP); Time-varying environmental factor; Change-point

# 1. Introduction

Reliability is a primary concern for both software developers and software users. With the rapid development of computer technology, computers are widely used to control safety-critical and civilian systems. High quality software products are greatly demanded in those application areas. To determine system reliability, the software reliability must be evaluated carefully. Many mathematical models called SRGMs (software reliability growth models) have been developed to describe the software-debugging phenomenon (Musa et al., 1989; Lyu, 1996; Huang et al., 1997; Pham et al., 1999; Kuo et al., 2001; Malaiya et al., 2002; Huang and Kuo, 2002). Under the assumption that testing is performed in accordance with a given operational profile (Lyu, 1996), SRGMs use the failure history

\* Corresponding author. E-mail address: zhaoj@ftcl.hit.edu.cn (J. Zhao). obtained during testing to predict the field behaviors of the program. NHPP models, as a class of SRGMs (Musa et al., 1989), are extensively used. NHPP SRGMs have been quite successful tools in practical software reliability engineering (Pham et al., 1999; Malaiya et al., 2002).

Most software reliability models assume that each failure occurs independently and randomly according to the same distribution during the fault detection process. However, in more realistic situations, the failure distribution can be affected by many factors, such as operational environment, testing strategy and resource allocation. Once these factors are changed during the software-testing phase, the software failure intensity function could increase or decrease non-monotonically, which is identified as a change-point problem (Zhao, 1993). In general, the FDR (fault detection rate) is used to measure the effectiveness of fault detection of test techniques and test cases. At the beginning of the testing, the FDR depends on fault discovery efficiency, fault density, test effort, and inspection rate. Later, in the middle stage of the testing phase, the FDR

<sup>0164-1212/\$ -</sup> see front matter @ 2006 Elsevier Inc. All rights reserved. doi:10.1016/j.jss.2006.02.030

depends on some parameters such as the failure-to-fault relationship, the code expansion factor, the skill of test teams, program size, and software testability, etc. Consequently, the FDR may be changed (Huang, 2005). The change-point problems have been studied by many researchers (Zhao, 1993; Chang, 2001; Zou, 2003; Shyur, 2003; Huang, 2005), and a few change-points are suggested by those researchers to avoid the piecemeal effects (Zou, 2003).

Although reliability models with change-points achieve a great improvement in the accuracy of evaluation of software reliability (Zou, 2003; Shyur, 2003; Huang, 2005), the models describe the difference of testing environments before and after the change-point using two entirely different FDRs, while traditional models have ignored such differences completely. In fact, there are both differences and links between the FDRs before and after the change-point. Software testing is an integrated and continuous process. The software testing process consists of several testing stages, including unit testing, integration testing, and system testing. At the stages of testing, the test teams and the operating systems are similar. So, the FDRs before and after the change-point should have some links with each other because of the similarity of the environments, and these links can be described using environmental factors. Environmental factors that profile the software development process have much impacts on software reliability, which is studied by some researchers (Zhang and Pham, 2000; Zhang et al., 2001), who identify six factors that have the most significant impact on software reliability including software complexity, programmer skill, testing effort, testing coverage, testing environment, and frequency of program specification change. Huang only used the factor of testing effort to link FDRs together (Huang, 2005). Environmental factors include many other important factors that affect software reliability in addition to the testing effort, which need to be considered and incorporated into the software reliability assessment (Zhang and Pham, 2000). So, environmental factors can be used to associate the FDR before the change-point with FDR after the change-point.

In order to quantify the environment mismatch due to the change-point problems of testing, a environmental factor proposed by Hoang Pham is used to describe the differences between the system test environment and the field environment (Zhang et al., 2002). In fact, the environments that respective phases experience during the software testing process are also different, and thus the environmental factors of Pham are extensively applied to the case where the testing phase of software has the change-point. Hoang Pham et al. define the environmental factor as k = $\bar{b}_{\text{test}}/\bar{b}_{\text{field}}$ , which is used to link the FDRs of the testing phase and the field operational phase, and  $b_{\text{test}}$  and  $b_{\text{field}}$ represent the long-term average per fault failure rate during the system test and the field, respectively. Hoang Pham et al. assume that the environment factor is constant. From the aspect of the software testing process, the testing phase is based on a testing profile, develops test cases, and uses various test strategies (such as boundary value test, equivalence class test, path test considering coverage ratio). Different testing cases have different failure detection capability. At any of the testing phases, testers will firstly run the test cases with strong testing capability and high percentage of coverage to improve the testing speed and efficiency, which will lead to reduction of the FDR. If the testing transfers to a new phase, the FDR still decreases similarly. It is very difficult to ensure that the two FDRs decrease in a same proportion during the testing phases. Therefore, for better description of the impact of environment on the FDR, a function varying with time should be used to describe environmental factors.

The objectives of this paper are, first, to link different testing phases using environmental factors and changepoint technique, second, to analyze the changing trend of environmental factors, and third, to incorporate the environmental function and the change-point to build SRGM, so that the assumed conditions of the model are closer to the actual environment of software testing, thereby improving the accuracy of evaluation of reliability.

The paper provides three primary contributions as follows:

- 1. The problems of existing models with the change-point and classic models occurring when describing software testing are analyzed.
- 2. The benefits of using environmental factors to associate the FDR before the change-point with the FDR after the change-point are analyzed, and compared with other researchers' methods.
- 3. A method for estimating the failure intensity after the change-point is proposed. The environmental factor  $\bar{k}(t)$  can be calculated according to the testing result of a previous version of an project or the testing result of a similar project performed previously, and given certain testing data before the change-point, the FDR before the change-point, b(t), can be calculated and finally, the initial failure intensity after the change-point can be obtained.
- 4. The SRGM incorporating environmental factor and change-point is proposed, and compared with other similar SRGMs based on two sets of failure data. The comparison result shows that the proposed model is better than other models within these two sets of failure data.

The rest of this paper is organized as follows. In Section 2, the changing influence of the environmental factor is analyzed. In Section 3, the FDR after the change-point of testing is computed from the time-varying environmental factor and the FDR before the change-point of testing, and then a NHPP model with change-point is derived which integrates the FDR before the change-point of testing and the proposed FDR function after the change-point of testing. Section 4 evaluates the proposed SRGM and

other NHPP SRGMs using two sets of software failure data. Finally, conclusions are given in Section 5.

# 2. Environmental factors varying with the testing process

# 2.1. Analysis of environmental factors varying with the testing process

The FDR is used to measure the effectiveness of fault detection of test techniques and test cases. The three kinds of FDR functions during software testing are as follows:

- (1) Constant b (Goel and Okumoto, 1979).
- (2) An increasing function with respect to the testing time (Pham et al., 1999).
- (3) A decreasing function with respect to the testing time (Yamada et al., 1983).
- (4) First increasing and then decreasing function with respect to the testing time (Liu et al., 2005).

Therefore, if the same SRGM is used before and after the change-point, there are four kinds of cases for the FDRs before and after change-point as shown in Fig. 1(a)–(d), where  $\tau$  is the change-point, and  $b_{bf}(t)$ denotes the FDR before the change-point,  $b_{af}(t)$  the FDR after the change-point. It can be seen from Fig. 1 that, the environmental factor is a constant in the first case and may be a variable in the other three cases. Thus, more generally, the environmental factor should be defined as a function of time. To account for the time-varying environmental factor, we propose the environment factor function varying with testing time. Let  $b_{bf}(t)$  denote FDR before the change-point, and  $b_{af}(t)$  denote FDR after the change-point of testing. Thus, the varying environmental factor is defined as follows:

$$k(t) = b_{\rm bf}(t)/b_{\rm af}(t) \quad t \in (\tau, +\infty]$$
(1)

Through the above analysis, the environmental function should be a varying form. In this paper, the change-point analysis is based on the assumption that there is one change-point.

To further explain the changing trend of the environmental factor with time, an analysis is made below using actual failure data. The average time-varying environmental factor is defined as follows:

$$\bar{k}(t) = \bar{b}_{\rm bf}(t)/\bar{b}_{\rm af}(t) \tag{2}$$

where  $\bar{b}_{bf}(t)$  and  $\bar{b}_{af}(t)$  represent the average FDRs before and after the change-point of testing.

Assuming the change-point occurs at time  $\tau$ , and after that environment of testing is changed, the testing ends at  $t_{end}$ . The expected number of faults detected and removed by time  $\tau$  is  $m(\tau)$ . After the change-point of testing, the expected number of faults detected and removed by time t is m(t), the actual number of total faults detected and removed during time t is N(t). The number of residual faults,  $\overline{N}(t)$ , can be calculated as

$$\overline{N}(t) = a - N(t) \tag{3}$$



Fig. 1. (a) Constant FDR, (b) an increasing FDR function with respect to the testing time, (c) a decreasing FDR function with respect to the testing time and (d) first increasing and then decreasing FDR function with respect to the testing time.

 $\overline{N}(t)$  can be obtained by replacing *a* with its least squares estimate (LSE)  $\hat{a}$  of G–O model applying all the testing failure data. The failure rate after the change-point of the software is given by

$$\lambda(t) = b_{\rm af}(t) \times [a - m(t)] \tag{4}$$

The following equation can be used for the average failure rate calculation:

$$\bar{\lambda}(t_i) = \frac{N(t_i) - N(t_{i-1})}{t_i - t_{i-1}}$$
(5)

where  $N(t_i)$  is the actual total number of faults detected by time  $t_i$ .

A variation of (4) can be obtained by replacing m(t) with N(t),  $\lambda(t)$  with  $\overline{\lambda}(t)$ . The following alternative representation can be used for the average FDR calculation:

$$\bar{b}_{\rm af}(t_i) = \frac{\bar{\lambda}(t_i)}{\hat{a} - N(t_i)} \tag{6}$$

Discrete and varying  $\bar{b}_{af}(t)$  can be obtained by applying Eq. (6), which can be used to derive discrete and average time-varying environmental factors of  $\bar{k}(t)$ , and thus the changing influences of time-varying environmental factor can be calculated as follows:

$$k(t_i) = b_{\rm bf}(t_i)/b_{\rm af}(t_i) \quad t_i \in (\tau, t_{\rm end}]$$

$$\tag{7}$$

# 2.2. Numerical and data analysis

The discrete time-varying environmental factors are derived via analysis of real failure data sets (Ohba, 1984; Musa et al., 1989). The first set of real data of PL/I data-base application, DS1, was from a study by Ohba (1984), and the second data set, DS2, was reported by Musa et al. (1989) based on failure data from a real time command and control application system. For both sets of failure data, assuming the change-point  $\tau$  is given since the testing strategy and resource allocation can be tracked during software development process (Zhao, 1993; Musa, 1998; Zou, 2003; Shyur, 2003). The change-point  $\tau$  of DS1 and DS2 are located around the sixth weeks and the eighth week (Zhao, 1993; Chang, 2001; Zou, 2003; Shyur, 2003; Huang, 2005).

### 2.2.1. Analysis with first data set and model selection

The G–O model, Yamada delayed S-shape model and logistic growth curve model are fit to the failure data (Ohba, 1984). The parameter's value of these models can

Table 1 Goodness-of-fit of three models for the first data set

Comparison criteria	SSE	R-square	LSE
G–O model (Goel and Okumoto, 1979)	332	0.9383	a = 236, b = 0.115
Yamada delayed S-shape (Yamada et al., 1983)	187.5	0.9751	a = 128.5, b = 0.6254
Logistic growth curve model (Yamada et al., 1983)	108.9	0.9856	a = 112.6, A = 19.37, b = 1.184

be estimated using least squares estimate method and are listed in Table 1, in which the goodness-of-fit is shown.

In this analysis, the goodness-of-fit of the curve is measured by the sum of squares of errors, SSE, and correlation index of the regression curve equation, *R*-square. The SSE and the *R*-square are defined as follows:

$$SSE = \sum_{i=1}^{n} (y_i - m(\hat{t}_i))^2$$
(8)

$$R\text{-square} = \frac{\sum_{i=1}^{n} (m(\hat{t}_i) - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$
(9)

where, *n* represents the number of failures that have been detected in the failure data sets,  $m(\hat{t}_i)$  represents the estimated value of the accumulated number of failures up to the time  $t_i$ , and  $y_i$  represents the observed value of the accumulated number of failures up to the time  $t_i$ :

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{10}$$

The smaller the value of SSE is, the better the curve fits. The *R*-square can take on any value between 0 and 1, with a value closer to 1 indicating a better fit.

From comparisons of these models, the goodness-of-fit of logistic growth curve is better than that of the others, so the logistic growth curve model is selected as the model that fit the failure data before the change-point of testing. The  $b_{bf}(t)$  can be concluded as follows:

$$\frac{\mathrm{d}m(t)}{\mathrm{d}t} = b_{\mathrm{bf}}(t)(a - m(t)) \tag{11}$$

The mean value function of logistic growth curve is

$$m(t) = \frac{a}{1 + A\mathrm{e}^{-bt}} \tag{12}$$

By substituting (12) into (11), the FDR of  $b_{bf}(t)$  can be derived as follows:

$$b_{\rm bf}(t) = \frac{b}{(1 + Ae^{-bt})} \tag{13}$$

 $b_{\rm bf}(t)$  of Eq. (13) is the non-decreasing S-shape, which denotes the testers learning-process. The learning is closely related to the changes in the efficiency of testing during a testing phase. The idea is that in organizations that have advanced software processes, testers might be allowed to improve their testing process as they learn more about the product (Pham et al., 1999; Zhang et al., 2002). This could result in a fault detection rate increase monotonically over the testing period. As the testing continues, the increase of FDR becomes slow gradually, the failure intensity of software will decrease significantly, the effectiveness of the testing will be lowered, and thus the tester will adopt new testing technologies and measures to improve the number of failures detected within a unit time, therefore the change-point is generated.  $\overline{b}_{bf}(t)$  can be approximately replaced by the FDR at the maximum level before the change-point of testing,  $\bar{b}_{bf}$ , which can be concluded as follows:



Fig. 2. Average fault detection rate after the change-point.

$$\bar{b}_{bf} = \lim_{t \to \infty} \frac{b}{1 + Ae^{-bt}} = b \tag{14}$$

 $\bar{b}_{af}(t)$  represents the average FDR after the change-point of testing, which can be derived using real failure data as shown in Fig. 2.

The changing trends of  $\bar{k}(t)$  is shown in Fig. 3.

2.2.2. Analysis with the second data set and model selection Since the data before the change-point of testing shows a S-shaped pattern (Musa et al., 1989), the logistic curve and Yamada S-shaped model are fit to the failure data of test-



Fig. 3. Environmental factor of the first data set.

ing. The parameter's value of these models can be estimated using least squares estimate method and are listed in Table 2, in which goodness-of-fit is shown. The logistic growth curve is selected as the model that fit the failure data before the change-point of testing.

 $\bar{b}_{af}(t)$  represents the average FDR after the change-point of testing, which can be derived using real failure data as shown in Fig. 4.

The changing trends of  $\bar{k}(t)$  is shown in Fig. 5.



Fig. 4. Average fault detection rate after the change-point.



Fig. 5. Environmental factor of the second data set.

Table 2

Goodness-of-fit of logistic model and Yamada model for data (Musa et al., 1989)

Comparison criteria	SSE	R-square	LSE
Logistic growth curve model (Yamada et al., 1983)	11.4	0.9483	a = 80, A = 245.7, b = 0.4908
Yamada delayed S-shape (Yamada et al., 1983)	29.1	0.8647	a = 200, b = 0.05

#### 2.2.3. Description of environmental function

From the above experiments of two data sets, the approximately decreasing trends of  $\bar{k}(t)$  are derived. This can be explained as following: with the testing proceeding, the effective use of testing strategies and tools of non-random testing makes the average FDR after the change-point of testing approximately non-decreasing, thus the average environmental factor,  $\bar{k}(t)$ , is decreasing with time. The approximately decreasing trend, of  $\bar{k}(t)$ , can be described as follows (Yamada et al., 1983):

$$\bar{k}(t) = B \cdot \exp(-\partial \cdot t) \tag{15}$$

# 3. A NHPP model with change-point and environmental function

The change-point problems have been studied by many researchers (Zhao, 1993; Chang, 2001; Zou, 2003; Shyur, 2003; Huang, 2005), and the same SRGMs are built before and after the change-point. However, the environment changes at the change-point. At the beginning of the test, testers may improve the test process rapidly as they learn more about the software product, and the FDR depends on the learning ability of testers. As time passes by, the learning-process slows down, the effective use of testing strategies and testing techniques makes the FDR higher than random testing (Beizer, 1990). Therefore, the SRGMs are different at the change-point. In multitudinous nonhomogeneous Poisson reliability models, the two most important parameters are the number of initial software faults and the FDR. If no faults are introduced into the software, the total number of faults remains constant, therefore the most important parameter is the FDR (Kuo et al., 2001). A certain relation can be established between the testing phase before and after change-point when establishing the software reliability model with change-point. How to obtain the FDR after the change-point of testing from the FDR before the change-point of the testing phase? The FDR after the change-point can be transformed from the environmental factor and the FDR before the changepoint of testing. It can be derived by Eqs. (2) and (15):

$$\bar{b}_{af}(t) = \frac{\bar{b}_{bf}(t)}{\bar{k}(t)} = \frac{\bar{b}_{bf}(t)}{Be^{-\partial * t}} = \frac{\bar{b}_{bf}(t) * e^{\partial * t}}{B}$$
(16)

A SRGM with change-point and environmental function can be built, and the present model assumes:

- (1) The failures of software testing obey NHPP.
- (2) Once a fault is detected, it will be eliminated at once, and no new faults are introduced.
- (3) The failure intensity at any time is proportional to the number of faults hidden in the software.
- (4) Before the change-point of testing, the fault detection rate captures the learning-process of software testers; and after the change-point of testing, the fault detection rate is the integrated result of environmental effects and the FDR before the change-point.

The mean value function before the change-point of the testing can be concluded as follows:

$$m(t) = \frac{a}{1 + A * \exp(-b * t)} \quad t \leqslant \tau \tag{17}$$

where  $\tau$  is the change-point occurs during software testing, and after that the software is delivered to another environment by applying different testing strategies and testing tools.

With respect to the mean value function, according to assumptions, the mean value of cumulative failures after the change-point of testing,  $m^{af}(t)$ , is

$$\frac{\mathrm{d}m^{\mathrm{af}}(t)}{\mathrm{d}t} = b_{\mathrm{af}}(t)[a - m(\tau) - m^{\mathrm{af}}(t)] \quad t > \tau \tag{18}$$

where  $b_{af}(t)$  can be approximately represented by  $\bar{b}_{af}(t)$  in Eqs. (2) and (15)

$$b_{\rm af}(t) = \frac{b_{\rm bf}}{\bar{k}(t)} = \frac{b_{\rm bf}}{Be^{-\partial t}}$$
(19)

Thus,  $m^{\text{af}}(t)$ , is

$$m^{\mathrm{af}}(t) = (a - m(\tau))(1 - \exp(-B^*(t))) \quad t > \tau$$
 (20)

where,

$$B^*(t) = \int_{\tau}^{t_{\text{end}}} b_{\text{af}}(t) dt \quad t > \tau$$
(21)

Constituting Eq. (19) into Eq. (21), we have

$$B^{*}(t) = \int_{\tau}^{t_{end}} \frac{\bar{b}_{bf} * e^{\partial * t}}{B} dt = \frac{\bar{b}_{bf}}{B\partial} \left( e^{\partial t_{end}} - e^{\partial \tau} \right)$$
$$= \frac{b}{B\partial} \left( e^{\partial t_{end}} - e^{\partial \tau} \right) \quad t > \tau$$
(22)

Consequently, the mean value functions before and after the change-point of testing are as follows:

$$m(t) = \begin{cases} \frac{a}{1+A*\exp(-b*t)} & t \leq \tau\\ (a-m(\tau))\left(1-e^{-\frac{b}{B\partial}(e^{\partial t_{end}}-e^{\partial \tau})}\right) + m(\tau) & t > \tau \end{cases}$$
(23)

The model addresses the FDR of change-point and environmental factors. We call the model CE-SRGM.

#### 4. Evaluation of the proposed model

In this section, we will evaluate the performance of CE-SRGM by using real failure data sets (Ohba, 1984; Musa et al., 1989) as described in Section 2. Since the proposed model is new to software reliability prediction/estimation, here we will compare its accuracy with those of some well-known SRGMs, such as the model proposed by Huang (2005), Goel–Okumoto (G–O) model (Goel and Okumoto, 1979), Yamada delayed S-shaped model (Yamada et al., 1983), logistic growth curve model (Yamada et al., 1983), and Goel generalized NHPP model (Goel, 1985).

Comparison criteria	LSE	SSE	R-square	AE (%)
CE-SRGM	$a = 374, b = 0.94, A = 16.72, B = 15.29, \partial = 1e-07$	127.2	0.9951	7.76
Model proposed by Huang	$a = 398.114, r_1 = 0.0473895, r_2 = 0.0408960$	136.8	0.9879	11.20
Yamada delayed S-shape (Yamada et al., 1983)	a = 374.1, b = 0.1977,	320.5	0.9837	17.76
G–O model (Goel and Okumoto, 1979)	a = 760.5, b = 0.03227	265.6	0.9865	34.36

Table 3 Comparison results of different SRGMs for the first data set

#### 4.1. Comparison criteria

- (1) The SSE and *R*-squares can show us the goodness-offit of a model for the given failure data set, and are described in Section 2.
- (2) The accuracy of estimation (AE), is defined as follows (Musa et al., 1989):

$$AE = \left| \frac{m_a - a}{m_a} \right| \tag{24}$$

where  $m_a$  is the actual cumulative number of detected faults after the test, and *a* is the estimated number of initial faults.

The capability of the model to predict failure behavior from present and past failure behavior is called predictive validity, which can be represented by computing the relative error (RE) for a data set (Musa et al., 1989):

Relative error(RE) = 
$$\frac{m(t_q) - q}{q}$$
 (25)

Assuming that we have observed q failures by the end of test time  $t_q$ , we use the failure data up to time  $t_e$  ( $t_e \le t_q$ ) to estimate the parameters of m(t). Substituting the estimates of these parameters in the mean value function yields the estimate of the number of failures  $m(t_q)$  by  $t_q$ . The estimate is compared with the actual number q. The procedure is repeated for various values of  $t_e$ . We can check the predictive validity by plotting the relative error for different values of  $t_q$ .

# 4.2. Model comparison with real applications

In this section, we examine the goodness-of-fit and predictive power of the proposed model and compare it with the existing SRGMs. For both sets of failure data, the least squares estimation (LSE) is used to estimate the parameters. We consider the parameter  $\tau$  is given since the testing strategy and testing resource allocation can be well tracked all the time during the software development (Zhao, 1993; Musa, 1998; Zou, 2003; Shyur, 2003).

# 4.2.1. DS1

For the first real failure data set (Ohba, 1984),  $\tau$  maybe located around the sixth week (Huang, 2005). Thus, the parameters of the proposed model and compared models can be estimated and listed in Table 3. Table 3 also shows the three comparison criteria: SSE, *R*-squares and AE. The values of SSE, *R*-squares and AE of our proposed model are better than those of the model proposed by Huang with k = 2.63326 and  $\tau = 6$ , G–O model and Yamada delayed S-shaped model.

Generally speaking, confidence intervals represent a range of values within which parameters are expected to lie with a certain confidence (Musa et al., 1989). Fig. 6(a)–(c) shows the cumulative faults, fitted faults, and the 95% global confidence bound of the proposed model, G–O model and Yamada delayed S-shape model.

The failure data of the first data set and the fitted curves of the proposed model and other famous NHPP SRGMs are shown in Fig. 6(d). Fig. 6(e)–(g) depicts the relative errors for G–O model, Yamada delayed S-shaped model and the proposed model for the first data set, respectively. It is noted that the relative error of the proposed model approached zero faster compared with G–O model and Yamada S-shaped model. From these figures and tables, it can be concluded that the proposed model provides a better prediction than G–O model and Yamada S-shaped model.

# 4.2.2. DS2

In this section, the predictive power of the proposed model and other SRGMs are tested using the second failure data set (Musa et al., 1989). For this data set,  $\tau$  maybe located around the eighth week (Zou, 2003; Huang, 2005). Thus, the parameters of the proposed model and compared

Table 4

Comparison results of different SRGMs for the first data s	se
--	----

Comparison criteria	LSE	SSE	R-square	AE (%)
CE-SRGM	$a = 140, b = 0.4737, A = 378.9, B = 13.45, \partial = 0.2713$	142	0.9931	17.8
Model proposed by Huang	$a = 139.665, r_1 = 0.757705, r_2 = 0.231555$	156	0.9928	25.71
Yamada delayed S-shape (Yamada et al., 1983)	a = 200, b = 0.09659	567	0.9042	47.76
Logistic growth curve model (Yamada et al., 1983)	a = 154.2, A = 146.1, b = 0.3483	235	0.9954	21.3
Goel generalized NHPP model (Goel, 1985)	a = 200, b = 0.0001391, c = 3.024	25.7	0.8256	56.8



Fig. 6. (a) The proposed model and the 95% global confidence bound vs. time for the first data set. (b) G–O model and the 95% global confidence bound vs. time for the first data set. (c) Yamada delayed S-shape model and the 95% global confidence bound vs. time for the first data set. (d) The failure data of the first data set, and the fitted curves of the proposed model and other famous NHPP SRGMs. (e) RE curve for the G–O model. (f) RE curve for the Yamada delayed S-shaped model.



Fig. 7. (a) The proposed model and the 95% global confidence bound vs. time for the second data set. (b) Yamada delayed S-shape model and the 95% global confidence bound vs. time for the second data set. (c) Goel generalized NHPP model and the 95% global confidence bound vs. time for the second data set. (d) Logistic growth curve model and the 95% global confidence bound vs. time for the second data set. (e) RE curve for the proposed model. (f) RE curve for the Yamada delayed S-shape model. (g) RE curve for the Goel generalized NHPP model. (h) RE curve for the logistic growth curve model.

models can be estimated and listed in Table 4. Besides, Table 4 also shows the three comparison criteria: SSE, *R*-

squares and AE. The values of SSE, *R*-squares and AE of our proposed model are better than those of model

proposed by Huang with k = 1.27171 and  $\tau = 8$ , Yamada S-shaped model, Goel generalized NHPP model and logistic growth curve model.

Fig. 7(a)–(d) shows the cumulative faults, fitted faults, and the 95% global confidence bound of the proposed model, Yamada delayed S-shape model, Goel generalized NHPP model and logistic growth curve model for the second data set. Fig. 7(e)–(h) depicts the relative errors for the proposed model, Yamada delayed S-shape model, Goel generalized NHPP model and logistic growth curve model for the second data set.

It is noted that the relative error of the proposed model approaches zero faster compared with Yamada delayed Sshaped model, Goel generalized NHPP model and logistic growth curve model.

From these figures and tables, it can be concluded that the proposed model provides a better prediction than Yamada delayed S-shaped model, Goel generalized NHPP model and logistic growth curve model.

### 4.3. Future work

- (1) Analyze the environmental factors affecting software testing comprehensively, and perform modeling and quantitative analyses on the environment factors.
- (2) Research the influence of the quantitized environmental factors on the effectiveness of software testing.
- (3) Introduce the model of the environmental factors into the SRGM so as to make the modeling process of software reliability more reasonable.

# 5. Conclusions

In this paper, the changing trend of the time-varying environmental factor is analyzed by applying actual failure data of testing, which can fairly quantify the mismatch between the testing environments before and after the change-point. The FDR after the change-point of testing is computed from the time-varying environmental factor and FDR before the change-point of testing, and A NHPP model called CE-SRGM is proposed which incorporates an environmental function and the change-point. Finally, the CE-SRGM is compared with other SRGMs, and the comparison results shows that predictive power of CE-SRGM is better than those of other SRGMs.

# Acknowledgements

The authors would like to thank the editor and referees for their valuable comments and suggestions that have improved the paper. This work is supported by the National Nature Science Foundation of China (60503015) and Ph.D. Programs Foundation of Ministry of Education of China under Grant No. 20020213017.

#### References

- Beizer, B., 1990. Software Testing Techniques. Van Nostrand Reinhold, 240–500.
- Chang, Y.P., 2001. Estimation of parameters for nonhomogeneous Poisson process software reliability with change-point model. Communications in Statistics: Simulation and Computation 30, 623–635.
- Goel, A.L., 1985. Software reliability models: assumptions, limitations, and applicability. IEEE Transactions on Software Reliability 11 (12), 1411–1423.
- Goel, A.L., Okumoto, K., 1979. Time-dependent error-detection rate model for software and other performance measures. IEEE Transactions on Reliability 28 (3), 206–211.
- Huang, C.Y., 2005. Performance analysis of software reliability growth models with test-effort and change-point. Journal of Systems and Software 76, 181–194.
- Huang, C.Y., Kuo, S.Y., 2002. Analysis of incorporating logistic testingeffort function into software reliability modeling. IEEE Transactions on Reliability 51, 261–270.
- Huang, C.Y., Kuo, S.Y., Chen, I.Y., 1997. Analysis of a software reliability growth model with logistic testing-effort function. In: Proceedings of Software Reliability Engineering (ISSRE'97), Albuquerque, NM, November, pp. 378–388.
- Kuo, S.Y., Huang, C.Y., Lyu, M.R., 2001. Framework modeling software reliability, using various testing-efforts and fault detection rates. IEEE Transactions on Reliability 50 (3), 310–320.
- Liu, Hong-wei, Yang, Xiao-Zong, Qu, Feng, Dong, Jiang, 2005. A software reliability growth model with bell-shaped fault detection rate function. Chinese Journal of Computers 28 (5), 908–913.
- Lyu, M.R., 1996. Handbook of Software Reliability Engineering. McGraw-Hill and IEEE Computer Society, New York.
- Malaiya, Y.K., Li, M.N., Bieman, J.M., Karcich, R., 2002. Software reliability growth with test coverage. IEEE Transactions on Reliability 51, 420–426.
- Musa, John D., 1998. Software Reliability Engineering: Software, Faster Development and Testing. McGraw-Hill.
- Musa, J.D., Iannino, A., Okumoto, K., 1989. Software Reliability: Measurement, Prediction, Application. McGraw-Hill, New York.
- Ohba, M., 1984. Software reliability analysis models. IBM Journal of Research and Development 28 (4), 428–443.
- Pham, H., Nordmann, L., Zhang, X.M., 1999. A general imperfectsoftware-debugging model with S-shaped fault-detection rate. IEEE Transactions on Reliability 48 (2), 169–175.
- Shyur, H.J., 2003. A stochastic software reliability model with imperfectdebugging and change-point. Journal of Systems and Software 66, 135–141.
- Yamada, S., Ohba, M., Osaki, S., 1983. S-shaped software reliability growth modeling for software error detection. IEEE Transactions on Reliability 32 (5), 475–484.
- Zhang, Xuemei, Pham, Hong, 2000. An analysis of factors affecting software reliability. Journal of Systems and Software 50, 43–56.
- Zhang, Xuemei, Shin, Mi-Young, Pham, Hoang, 2001. Exploratory analysis of environmental factors for enhancing the software reliability assessment. Journal of Systems and Software 57, 73–78.
- Zhang, X.M., Jeske, Daniel R., Pham, Hoang, 2002. Calibrating software reliability models when the test environment does not match the user environment. Applied Stochastic Models in Business and Industry 18, 87–99.
- Zhao, M., 1993. Change-point problems in software and hardware reliability. Communications in Statistics-Theory and Methods 22 (3), 757–768.
- Zou, F.Z., 2003. A change-point perspective on the software failure process. Software Testing, Verification and Reliability 13, 85–93.