A Unified and Flexible Framework of Imperfect Debugging Dependent SRGMs with Testing-Effort

Ce Zhang*
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai, China
*Correspondence author, Email: zhangce@hitwh.edu.cn

Gang Cui and Hongwei Liu
School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China
Email: cg@fcl.hit.edu.cn, lhw@fcl.hit.edu.cn

Fanchao Meng and Shixiong Wu
School of Computer Science and Technology, Harbin Institute of Technology at Weihai, Weihai, China
Email: mengfanchao74@163.com, ggggokxwl@gmail.com

Abstract—In order to overcome the limitations of debugging process, insufficient consideration of imperfect debugging and testing-effort (TE) in software reliability modeling and analysis, a software reliability growth model (SRGM) explicitly incorporating imperfect debugging and TE is developed. From the point of view of incomplete debugging and introduction of new fault, software testing process is described and a relatively unified SRGM framework is presented considering TE. The proposed framework models are fairly general models that cover a variety of the previous works on SRGM with ID and TE. Furthermore, a special SRGM incorporating an improved Logistic testing-effort function (TEF) into imperfect debugging modeling is proposed. The effectiveness and reasonableness of the proposed model are verified by published failure data set. The proposed model closer to real software testing has better descriptive and predictive power than other models.

Index Terms—Software Reliability; Software Reliability Growth Model (SRGM); Imperfect Debugging; Testing-Effort

I. INTRODUCTION

Software reliability is important attribute and can be measured and predicted by software reliability growth models (SRGMs) which have already been extensively studied and applied [1-2]. SRGM usually views software testing as the unification of several stochastic processes. Once a failure occurs, testing-effort (TE) can be expended to carry out fault detection, isolation and correction. In general, with the removal of faults in software, software reliability continues to grow. SRGM has become a main approach to measure, predict and ensure software reliability during testing and operational stage.

As software reliability is closely related to TE, incorporating TE into software reliability model becomes normal and imperative, especially in imperfect debugging environment. As an important representative in sketching the testing resource expenditure in software testing, TE can be represented as the number of testing cases, CPU hours and man power, etc. In software testing, when a failure occurs, TE is used to support fault detection and correction. A considerable amount of research on TE applied in software reliability modeling has been done during the last decade [3-8]. TE which has different function expressions, can be used to describe the testing resource expenditure [4]. The available TEFs describing TE include constant, Weibull (further divided into Exponential, Rayleigh and Weibull, and so on) [4], log-logistic [5], Cobb-Douglas function (CDF) [7], etc. Besides, against the deficiency of TEF in existence, Huang presented Logistic TEF [3] and general Logistic TEF [6] to describe testing-effort expenditure. Finally, TE can also help software engineer to conduct optimal allocation of testing resources in component-based software [9].

In fact, software testing is very complicated stochastic process. Compared with perfect debugging, imperfect debugging can describe testing process in more detail. So, in recent years, imperfect debugging draws more and more attention [10-16]. Imperfect debugging is an abstraction and approximation of real testing process, considering incomplete debugging [12] and introduction of new faults [10, 11]. It can also be studied by the number of total faults in software [3, 4]. Reference [4] combined Exponentiated Weibull TEF with Inflection S-shaped SRGM to present a SRGM incorporating imperfect debugging described by setting fault detecting...
rate \( b(t) = b \left( r + (1-r) \frac{m(t)}{a} \right) \). Obviously, when \( r = 1 \), the proposed model has evolved to exponential SRGM. Likewise, Ahmad [13] also proposed an inflection \( S \)-shaped SRGM considering imperfect debugging and had Log-logistic TEF employed in his SRGM. Besides, there is also research that suggests incorporating imperfect debugging and TE into SRGM to describe software testing process from the view of the variation of \( a(t) \). For example, reference [14] presented \( a(t) = a e^{\alpha t} \), and [3] employed \( a(t) = a + \beta m(t) \). Considering the fact that so-called “peak phenomenon” occurs when \( m > 3 \) in EW TEF did not conform to real software testing [15], Huang introduced imperfect debugging environment into analysis by combining Logistic TEF with exponential and \( S \)-shaped SRGM to establish reliability model, finally obtaining a better effect. Kapur [16] proposed a unified SRGM framework considering TE and imperfect debugging, in which real testing process was divided into failure detection and fault correction, and convolution of probability distribution function was employed to represent the delay between fault detection and correction process. The imperfect debugging above is described by complete debugging probability \( p \) and by introducing new faults: \( a(W_t) = a + \alpha m(W_t) \). Compared to the others, the proposed imperfect debugging in [16] is relatively thorough. Actually these research efforts conducted from different views and contents, lack thorough and accurate description.

On the above basis, in the statistical literatures, some studies have involved imperfect debugging and TE. However, little research has been conducted to fully incorporate ID and TE into SRGM, failing to describe the real software testing. Thus, we come to know how important and imperative it is to incorporate ID and TE into software reliability modelling.

Obviously, in testing, the more real factors SRGM considered, the more accurate the software testing process would be. In this paper, a SRGM framework incorporating imperfect debugging and TE is presented and can be used to more accurately describe software testing process on the basis of the existing research. Unlike the earlier techniques, the proposed SRGM covers two types of imperfect debugging including incomplete debugging and introduction of new faults. It unifies contemporary approaches to describe the fault detection and correction process. Moreover, an improved Logistic TEF with Not Zero Initialization is presented and verified to illustrate testing resource consumption. Finally, a special SRGM: SRGM-GTEFID is established. The effectiveness of SRGM-GTEFID is demonstrated through a real failure data set. The results confirm that the proposed framework of imperfect debugging dependent SRGMs with TE is flexible, and enables efficient reliability analysis, achieving a desired level of software reliability.

The paper is structured as follows: Sec.2 presents a unified and flexible SRGM framework considering imperfect debugging and TE. Next, an improved Logistic TEF is illustrated to build a special SRGM in Sec.3. Sec.4 shows experimental studies for verifying the proposed model. Sec.5 contains some conclusions plus some ideas for future work.

II. THE UNIFIED SRGM FRAMEWORK CONSIDERING IMPERFECT DEBUGGING AND TE

A. Basic Assumptions

In subsequent analysis, the proposed model and study is formulated based on the following assumptions [3, 4, 17-21].

1. The fault removal process follows a non-homogeneous poisson process (NHPP);
2. Let \( \{N(t), t \geq 0\} \) denote a counting process representing the cumulative number of software failure detected by time \( t \), and \( N(t) \) is a NHPP with mean value function \( m(t) \) and failure intensity function \( \lambda(t) \) respectively;

\[
\Pr(N(t) = k) = \frac{(m(t))^k e^{-m(t)}}{k!}, k = 0, 1, 2, \ldots . \tag{1}
\]

\[
m(t) = \int_0^t \lambda(r) \, dr . \tag{2}\]

3. The cumulative number of faults detected is proportional to the number of faults not yet discovered in the time interval \((t, t+\Delta t)\) by the current TE expenditures, and the proportion function is \( b(t) \) hereinafter referred to as FDR;
4. The fault removal is not complete, that is fault correction rate function is \( p(t) \);
5. New faults can be introduced during debugging, fault introduction probability is proportional to the number of faults corrected, and the probability function is \( r(t) \) \((r(t) < p(t)) \).

B. General Imperfect Debugging Dependent Framework Model Considering TE

Based on the above assumptions, the following differential equations can be derived as:

\[
\begin{align*}
\frac{dm(t)}{dt} &= b(t) \left[ a(t) - c(t) \right] \\
\frac{dc(t)}{dt} &= p(t) \frac{dm(t)}{dt} \\
\frac{da(t)}{dt} &= r(t) \frac{dc(t)}{dt}
\end{align*}
\]

where \( a(t) \) denotes the total number of faults in software, \( c(t) \) the cumulative number of faults corrected in \([0, t]\) and \( w(t) \) TE consumption rate at \( t \), that is \( W(t) = \int_0^t w(x) \, dx \). Solving the differential equations above with the boundary condition of \( m(0) = 0 \), \( a(0) = a \), \( c(0) = 0 \) yields

\[
c(t) = a \int_0^t w(u) b(u) p(u) e^{-\int_0^u \alpha x \, dx} \left[ 1 - e^{-\int_0^u \lambda(x) \, dx} \right] du . \tag{4}\]
\[ a(t) = a \left[ 1 + \int_0^t w(u)b(u)p(u)r(u)e^{-\int_0^u w(r)b(r)p(r)(1-r)rdr} du \right] \] (5)

\[ m(t) = a \left[ 1 + \int_0^t w(v)b(v) \times \left\{ 1 + \frac{\int_0^t w(u)b(u)p(u)(1-r(u))e^{-\int_0^u w(r)b(r)p(r)(1-r)rdr} du}{dv} \right\} dv \right] \] (6)

Then the current failure intensity function \( \lambda(t) \) can be derived as:
\[ \lambda(t) = \frac{dm(t)}{dt} = aw(t)b(t) \times \left\{ 1 + \frac{\int_0^t w(u)b(u)p(u)(1-r(u))e^{-\int_0^u w(r)b(r)p(r)(1-r)rdr} du}{dv} \right\} \] (7)

Obviously, by setting the different values for \( b(t), p(t), r(t) \) and \( w(t) \), we can obtain the several available models.

(1) If \( p(t)=1, r(t)=0 \) and regardless of TE, then the proposed model has evolved into classical G-O model [17];

(2) If \( p(t)=1, r(t)=0 \) and TE is Yamada Weibull, Burr type X, Logistic, generalized Logistic or Log-Logistic respectively, then the proposed model has evolved into the models in references [5,22];

(3) If \( p(t)=1, r(t)=0, b(t)=b \frac{r+(1-r)m(t)/a}{(1+rb+m(t)/a)} \) and TE is Weibull, then the proposed model has evolved into the model in [4];

(4) In framework model, if \( p(t)=1, r(t)=1, b(t)=b^*\frac{1}{(1+rb+m(t)/a)} \) and TE is framework function, the proposed model has evolved into the model in [3];

(5) If \( p(t)=1, r(t)=0, a(t) \) is increasing function versus time \( t \), and TE is framework function, the proposed model has evolved into the model in [14];

(6) If \( p(t)=1, r(t)=0 \) and TE and \( b(t) \) are framework functions, the proposed model has evolved into the framework model in [15].

Thus, it can be seen that the proposed framework model is a generalization over the previous works on imperfect debugging and TE and is more flexible imperfect debugging framework model incorporating TE. In a practical application, \( w(t), b(t), p(t), r(t) \) and \( c(t) \) can be set to the proper functional forms as needed to accurately describe real debugging environment. The proposed model in this study incorporating imperfect debugging by the current TE expenditures is more flexible and referred to as SRGM-considering Generalized Testing-Effort and Imperfect Debugging Model (SRGM-GTEFID).

III. THE IMPERFECT DEBUGGING DEPENDENT SRGM WITH IMPROVED LOGISTIC TEF

Generally speaking, the most important factors affecting reliability are the number of total faults: \( a(t) \), fault detection rate (FDR): \( b(t) \) [21], and TE expenditure rate: \( w(t) \). Hereon, we have obtained the expression of \( a(t) \), and \( w(t) \) and \( b(t) \) will be discussed below.

Hereon, we present an improved Logistic TEF based on Logistic TEF [6, 15, 23, 24].

\[ W(t) = \left( \frac{1-e^{-\alpha t}}{1+e^{-\alpha t}} \right)_W \] (8)

where \( W \) represents total TE expectation, \( k \) and \( l \) denote the adjustment coefficient value, and \( \alpha \) is the consumption rate of TE expenditure. At some point, TE expenditure rate \( w(t) \) is:
\[ \frac{dW(t)}{dt} = W(t) = W \left( \frac{a(k+l)e^{-\alpha t}}{(1+ke^{-\alpha t})^2} \right) \] (9)

Obviously, \( W(0)=W \left( \frac{1-l}{1+k} \right) \neq 0 \) indicates that a certain amount of TE needs to be expended before the test begins. As \( w(t)>0, W(t) \) is an increasing function with testing time \( t \), and corresponds to the growing variation trend of TE expenditure. When \( t_{\text{max}} = \frac{\ln k}{\alpha} \), \( W(t) \) achieves maximum: \( W_{\text{max}}(t) = W \left( \frac{a(k+l)}{4k} \right) \). Obviously, \( W(t) \) first rises then falls.

In a considerable amount of research, many research studies suggest that \( b(t) \) is constant value [17], increasing function or decreasing function versus time \( t \). For example, \( b(t)=b^* \) [20], \( b(t)=b(0)+km(t)/a \) [15], \( b(0)= \frac{b}{1+\beta e^{-\alpha t}} \) and \( b(t)=b(0) \left[ 1-m(t)/a \right] \) [15]. Actually, these \( b(t) \) functions can only describe the varying of FDR at some stage of software testing. Hereon, we present a relatively flexible \( b(t) \) to comprehensively illustrate FDR.

\[ b(t) = \left( \frac{e^{-\alpha t}}{1+\beta e^{-\alpha t}} \right)_b \] (10)

In our previous study, (10) has been verified to describe the various changing trends of FDR.

For simplicity and tractability, let \( p(t)=p \) and \( r(t)=r \) is constant fault introduction rate due to \( r(t)<sp(t) \). If \( p\neq0 \) and \( r\neq0 \) obtained in experiment, the fault removal process is imperfect, namely, there exist incomplete debugging and introducing new faults phenomena. Below we elaborate the SRGM obtained when \( W(t) \) and \( b(t) \) are set to expressions in (8) and (10) respectively.

For convenience of exposition here, Let \( g(t)=w(t)b(t) \).

\[ f(v) = \int_0^v g(u)e^{-p(1-r)\int_u^v g(t)dt} du \] (11)

By integral transform, (11) can be converted to the following form:
\[ f(v) = -\frac{1}{p(1-r)} \left\{ e^{p(1-r)\int_u^v g(t)dt} - 1 \right\} \] (12)

Substitute (12) into (6), we can get:
\[ m(t) = a \int_0^v g(v)e^{-p(1-r)\int_0^v g(t)dt} dv \] (13)
By the similar integral transform above, we can obtain:

\[ m(t) = \frac{a}{p(1-r)} \left\{ \left[ -p(t) \sum_{k=0}^{\infty} \left( \frac{a}{1 + re^{-a}} \right)^k \right] \right\}^{-1} \]  
(14)

where

\[ G(t) = \int g(r) dr = \int w(r) b(r) dr = \int W \left( \frac{a(k+1)e^{-a}}{1 + ke^{-a}} \right) \left( \frac{b e^{-a}}{1 + be^{-a}} \right) dr \]

\[ = W a b (k+1) \left[ \frac{1}{1 + (k+1)e^{-a}} \right] W \left( \frac{a(k+1)e^{-a}}{1 + (k+1)e^{-a}} \right) \]

Substitute \( G(t) \) in (15) into (14), finally, \( m(t) \) is derived as:

\[ m(t) = \frac{a}{p(1-r)} \times \left\{ 1 - e^{-p(t) \sum_{k=0}^{\infty} \left( -\beta \right)^k \left( -\gamma \right)^k (n_k + 1) \left( -1 + e^{\left( \gamma + \beta \right) \left( n_k + 1 \right) a + \delta} \right) \right\} \]

(16)

Accordingly, \( a(t) \) and \( c(t) \) can also be solved as follows:

\[ c(t) = \frac{a}{1 - r} \times \left\{ 1 - e^{-p(t) \sum_{k=0}^{\infty} \left( -\beta \right)^k \left( -\gamma \right)^k (n_k + 1) \left( -1 + e^{\left( \gamma + \beta \right) \left( n_k + 1 \right) a + \delta} \right) \right\} \]

(17)

\[ a(t) = a \left( \frac{2r - 1}{1 - r} \right) \times \left\{ 1 - e^{-p(t) \sum_{k=0}^{\infty} \left( -\beta \right)^k \left( -\gamma \right)^k (n_k + 1) \left( -1 + e^{\left( \gamma + \beta \right) \left( n_k + 1 \right) a + \delta} \right) \right\} \]

(18)

IV. EXPERIMENTAL STUDIES AND PERFORMANCE COMPARISONS

A. Criteria for Model Comparisons

Here, to assess the models, \( MSE \), \( Variance \), \( RMS-PE \), \( BMMRE \) and \( R-square \) are used to measure the curve fitting effects and \( RE \) to measure the predictive abilities.

\[ MSE = \frac{1}{k} \sum_{i=1}^{k} \left( y_i - m(t_i) \right)^2 \]  
(19)

\[ R-square = \frac{\sum_{i=1}^{k} \left( m(t_i) - \bar{y} \right)^2}{\sum_{i=1}^{k} \left( y_i - \bar{y} \right)^2}, \quad \bar{y} = \frac{1}{k} \sum_{i=1}^{k} y_i \]  
(20)

\[ RE = \frac{m(t_e) - q}{q} \]  
(21)

\[ Variance = \sqrt{\frac{1}{k-1} \sum_{i=1}^{k} \left( y_i - m(t_i) - Bias \right)^2} \]  
(22)

\[ Bias = \frac{1}{k} \sum_{i=1}^{k} \left( m(t_i) - y_i \right) \]  
(23)

\[ RMS-PE = \sqrt{Bias^2 + Variance} \]  
(24)

\[ BMMRE = \frac{1}{k} \sum_{i=1}^{k} \left| \frac{m(t_i) - y_i}{\min\left( m(t_i), y_i \right)} \right| \]  
(25)

where \( y_i \) represents the cumulative number of faults detected, \( m(t_i) \) denotes the estimated value of faults by time \( t_i \) and \( k \) is the sample size the real failure data set. Obviously, the smaller the values of \( MSE \), \( Variance \), \( RMS-PE \), \( BMMRE \), the closer to 1 of \( R-square \), the quickly closer to 0 of \( RE \), which indicates better model than the others.

<table>
<thead>
<tr>
<th>Model</th>
<th>( m(t) )</th>
</tr>
</thead>
</table>
| SSRGM-EWTEFID [4]            | \[ m(t) = \frac{a}{p(1-r)} \times \left\{ 1 - e^{-p(t) \sum_{k=0}^{\infty} \left( -\beta \right)^k \left( -\gamma \right)^k (n_k + 1) \left( -1 + e^{\left( \gamma + \beta \right) \left( n_k + 1 \right) a + \delta} \right) \right\} \]
|                               | \[ W(t) = W \left( 1 - e^{-a} \right)^p \] |
| SSRGM-LTEFID [3]             | \[ m(t) = \frac{a}{1 - r} \times \left\{ 1 - [1 + bW(t)] e^{-a + \beta \left( n_k + 1 \right) a} \right\} \]
|                               | \[ W(t) = W \left( 1 + A \right)^{-\frac{W}{1+A}} \] |
| SSRGM-GTEFID (the proposed model) | \[ m(t) = \frac{a}{p(1-r)} \times \left\{ 1 - e^{-p(t) \sum_{k=0}^{\infty} \left( -\beta \right)^k \left( -\gamma \right)^k (n_k + 1) \left( -1 + e^{\left( \gamma + \beta \right) \left( n_k + 1 \right) a + \delta} \right) \right\} \]
|                               | \[ W(t) = W \left( 1 + A \right)^{-\frac{W}{1+A}} \] |

B. Failure Data Set and the Selected Models for Comparison

Hereon, in order to demonstrate the effectiveness and validity of proposed model, we designate a failure data set as an example which has been used and studied extensively to illustrate the performance of SRGM [25]. In the meanwhile, three pre-eminent models considering imperfect debugging and TE are also selected to be compared with TEID-SRGM.
C. Experimental Results and Comparative Studies

First, in order to verify the effectiveness of improved Logistic TEF, we compared the proposed \( W(t) \) to that of models in Table 1, Generalized Logistic TEF [6], Rayleigh TEF [4], and Weibull TEF [4]. The goodness of TE has been drawn to illustrate the fitting of TEFs in Fig.1. From Fig.1, we can see that the models fit the real TE well except Generalized Logistic TEF and Yamada Rayleigh TEF.

![Figure 1](https://via.placeholder.com/150)

Furthermore, here we give the criteria values of \( W(t) \) as shown in Table 2. As indicated in Table 2, the values of \( MSE \), Variance, \( RMS-PE \) and \( BMMRE \) for \( W(t) \) of SRGM-GTEFID are smallest, and the value of \( R-square \) is closest to 1. Obviously, the results provide better goodness of fit for failure data and proposed improved Logistic TEF is suitable for modeling the testing resources expenditure than the others.

| Table II. Comparison Results for Different TEFs |
|--------|---------|---------|---------|---------|---------|
| TEF Model | \( MSE \) | \( R-square \) | Variance | \( RMS-PE \) | \( BMMRE \) |
| Logistic TEF | 1.6271 | 0.9680 | 1.3221 | 1.3103 | 0.1066 |
| Generalized Logistic TEF | 1.3361 | 0.9784 | 1.1915 | 1.1875 | 0.0857 |
| Yamada Rayleigh TEF | 5.1476 | 1.1757 | 2.7599 | 2.3227 | 0.6374 |
| Yamada Weibull TEF | 0.9022 | 1.0126 | 0.9845 | 0.9757 | 0.0842 |
| Generalized Exponential TEF | 0.8502 | 1.0071 | 0.9473 | 0.9478 | 0.0496 |
| Improved Logistic TEF | .805117071 | .9945 | 3.279 | .9478 | 6124 |

By calculating, \( n_1 = 5 \) and \( n_2 = 2 \) in (15) can satisfy the requirements. The parameters of the models are estimated based upon the failure data set and estimation results are shown in Table 3.
As can be seen from Table 3, the estimated value \( p \) and \( r \) of SRGM-GTEFID are not equal to zero \((p=0.8304480, r=0.03087796, \text{and} \ r<0.5)\). Therefore we can conclude that the fault removal process is imperfect.

Next, the fitting curve of the estimated cumulative number \( m(t) \) of failures is graphically illustrated in Fig. 2.

As seen from Fig. 2, it can be found that the proposed model (SRGM-GTEFID) is very close to the real failure data and fits the data excellently well. Furthermore, we calculate comparison criteria results of all the models as presented in Table 4. It is clear from the Table 4 that the values of \( MSE, Variance, RMS-PE \) and \( BMMRE \) in SRGM-GTEFID are the lowest in comparison with models, and SRGM-GTEFID is followed by SSRGM-EWTEFID, and DSSRGMLTEFID. On the other hand, in the \( R-square \) comparison, SRGM-GTEFID and SSRGM-EWTEFID are the best, slightly differing in the fourth decimal points of \( R-square \) value and closely approximate to 1. Thus, \( R-square \) value of SRGM-GTEFID is excellent. Moreover, the values of \( MSE, Variance \) and \( BMMRE \) for SSRGM-EWTEFID are not very close to the proposed model. Therefore, SRGM-GTEFID provides better goodness of fit for failure data set than the other three models, and can almost be considered the best. The result can be explained in the following. DSSRGMLTEFID not only ignores incomplete debugging but also sets FDR to \( b(t)=b(1+b) \) form which are hard to describe different situations. Likewise, SSRGM-EWTEFID also thinks debugging is complete and sets \( b(t)=b'W(t) \) form which cannot show accurately the variation trend of FDR. In describing TE function \( W(t) \), SSRGM-EWTEFID employs complicated Exponentiated Weibull distribution TEF, but DSSRGMLTEFID employ Logistic TEF. These TEFs diverge from the real testing resources expenditures. Due to these insufficiencies, descriptive powers of these two models are inferior to that of the proposed one.

![Graph of Observed/Estimated Cumulative Number of Failures vs Time](image)

**Figure 2.** Observed/estimated cumulative number of failures vs time

<table>
<thead>
<tr>
<th>Model</th>
<th>Estimation of model parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSRGM-EWTEFID</td>
<td>( \hat{a}=392.41819765, \hat{b}=0.05845694, \hat{\phi}=0.39793805 ), ( \hat{\psi}=67.3168 ), ( \hat{\sigma}=4.8380 ), ( \hat{\theta}=0.231527 )</td>
</tr>
<tr>
<td>DSSRGMLTEFID</td>
<td>( \hat{a}=181.415525, \hat{b}=0.13939333, \hat{r}=0.5076305, \hat{\psi}=3.1658 ), ( \hat{\phi}=5.0814 ), ( \hat{\theta}=0.03087796 )</td>
</tr>
<tr>
<td>SRGM-GTEFID</td>
<td>( \hat{a}=265.81098261, \hat{b}=0.00002672, \hat{\phi}=0.8304480 ), ( \hat{\psi}=0.05845694 ), ( \hat{\theta}=0.00000017 )</td>
</tr>
</tbody>
</table>

**TABLE IV.** Comparison Criteria Results of the Models

<table>
<thead>
<tr>
<th>Model</th>
<th>( MSE )</th>
<th>( R-square )</th>
<th>( Variance )</th>
<th>( RMS-PE )</th>
<th>( BMMRE )</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSRGM-EWTEFID</td>
<td>85.963</td>
<td>1.0177</td>
<td>9.6015</td>
<td>9.5243</td>
<td>0.0642</td>
</tr>
<tr>
<td>DSSRGMLTEFID</td>
<td>477.39</td>
<td>1.2337</td>
<td>25.569</td>
<td>26.479</td>
<td>0.5938</td>
</tr>
<tr>
<td>SRGM-GTEFID</td>
<td>93.565</td>
<td>1.0181</td>
<td>8.627</td>
<td>8.5956</td>
<td>0.0640</td>
</tr>
</tbody>
</table>

In predictive capability, the relative error \( RE \) in prediction is calculated and the results are shown graphically in Fig. 3 respectively. It is noted that the \( RE \) of the models approximate fast to zero. Furthermore, we can see that SRGM-GTEFID is not close to zero most quickly in all the models in the beginning. And for this, we compute the \( RE \) in prediction for the models in Table 1 at the end of testing and the results are shown in Table 5. As indicated in Table 5, the minimums of \( RE \) in the final four testing time \((0.0625893076791, 0.02181151519274, 0.00866202271253 \) and \( 0.00502853969464, \) respectively) indicate the better prediction ability than the others. Thus, predictive capability of SRGM-GTEFID presents a gradually rising tendency. The reason for this is that, due to involving more parameters, predictive performance of SRGM-GTEFID is modest when the failure data set is small, and predictive performance is increasing and superior to the other models when larger failure data set is employed.
Altogether, from Fig. 1-3 and Table 2, 4 and 5, we conclude that the proposed model (SRGM-GTEFID) fits well the observed failure data than the others and gives a reasonable prediction capability in estimating the number of software failures. Moreover, from Table 2, it can be concluded that incorporating improved Logistic TEF into SRGM-GTEFID yields a better fitting and can be used to describe the real testing-effort expenditure.

V. CONCLUSIONS

A relatively unified and flexible SRGM framework considering TE and ID is presented in this paper. By incorporating the improved Logistic TEF into software reliability models, the modified SRGM become more powerful and more informative in the software reliability engineering process. By the experimentation, we can conclude that the proposed model is more flexible and fits the observed failure data better and predicts the future behavior better. Obviously, developing SRGM tailored to diverse testing environment is main research direction in view of imperfections in the real testing. Thus, change point (CP) problem, the delay between fault detection process (FDP) and fault correction process (FCP), and dependence of faults should be incorporated to enlarge the researching range of imperfect debugging. Further research on these topics would be worthwhile.

ACKNOWLEDGMENT

This research was supported in part by the National Key R&D Program of China (No.2013BA17F02), the National Nature Science Foundation of China (No.60503015), and the Shandong province Science and Technology Program of China (No.2011GGX10108, 2010GGX10104).

REFERENCES


Ce Zhang, born in 1978, received Bachelor and Master degrees of computer science and technology from Harbin Institute of Technology (HIT) and Northeast University (NEU), China in 2002 and 2005, respectively. He has been a Ph.D. candidate of HIT major in computer system structure since 2010. His research interests include software reliability modeling, Fault-Tolerant Computing (FTC) and Trusted Computing (TC).

Gang Cui was born in 1949 in China. He earned his M.S. degree in 1989 and B.S. degree in 1976, both in Computer Science and Technology from Harbin Institute of Technology at Harbin. He is currently a professor and Ph.D. supervisor in School of Computer Science and Technology at Harbin Institute of Technology. He is a member of technical committee of fault tolerant computing of the computer society of China. His main research interests include fault tolerance computing, wearable computing, software testing, and software reliability evaluation. Prof. Cui has implemented several projects from the National 863 High-Tech Project and has won 1 First Prize, 2 Second Prizes and 3 Third Prizes of the Ministry Science and Technology Progress. He has published over 50 papers and one book.

HongWei Liu was born in 1971 in China, is doctor, processor and doctoral supervisor of HIT. His research interests include software reliability modeling, FTC and mobile computing.

FanChao Meng was born in 1974 in China, is doctor and associate processor of HIT. His research interests include software architecture derived by model, software reliability modeling software reconstruction and reuse, and Enterprise Resource Planning (ERP).