Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging

N. Ahmad[a], M. G. M. Khan[b] and L. S. Rafi[b]

[a]University Department of Statistics and Computer Applications
T. M. Bhagalpur University, Bhagalpur-812007, India
[b]School of Computing Information and Mathematical Sciences
The University of the South pacific, Suva, Fiji Island

Abstract
Gokhale and Trivedi (1998) have proposed the Log-logistic software reliability growth model that can capture the increasing/ decreasing nature of the failure occurrence rate per fault. In this paper, we will first show that a Log-logistic testing-effort function (TEF) can be expressed as a software development/testing-effort expenditure curve. We investigate how to incorporate the Log-logistic TEF into inflection S-shaped software reliability growth models based on non-homogeneous Poisson process (NHPP). The models parameters are estimated by least square estimation (LSE) and maximum likelihood estimation (MLE) methods. The methods of data analysis and comparison criteria are presented. The experimental results from actual data applications show good fit. A comparative analysis to evaluate the effectiveness for the proposed model and other existing models are also performed. Results show that the proposed models can give fairly better predictions. Therefore, the Log-logistic TEF is suitable for incorporating into inflection S-shaped NHPP growth models. In addition, the proposed models are discussed under imperfect debugging environment.

Keywords:
Software reliability growth models, Testing-effort functions, Software testing, Imperfect debugging, Inflection S-shaped NHPP growth model, Estimation methods.

1. Introduction

The size and complexity of computer systems has grown significantly during the past decades. Computers are used in medical fields, businesses, chemical labs, air traffic control towers, ships, space ships, home appliances, communication, manufacture and many more. Software is a functioning element embedded in computers that plays vital role in the modern life. Errors are bound to happen as software is written by humans. Before, the focus was only on the design and reliability of the hardware. But, now increase in the demand of software has led to the study of the high quality reliable software development. Reliability is the most important aspect since it measures software failures during the process of software development. Software reliability is defined as the probability of failure-free operation of a computer program for a specified time in a specified environment (Musa et al., 1987). Many researches have been conducted over the past decades (Pham, 2000; Lyu, 1996; Musa et al. 1987) and still going on, to study the software reliability. A common approach for measuring software reliability is by using an analytical model whose parameters are generally estimated from available data on software failures (Lyu, 1996; Musa et al. 1987).

A software reliability growth model (SRGM) is a mathematical expression of the software error occurrence and the removal process. In early 1970's, many software reliability growth models (SRGMs) have been proposed (Lyu, 1996; Xie, 1991; Musa et al., 1987). A Non-homogeneous Poisson process (NHPP) as the stochastic process has been widely used in SRGM. In the past years, several SRGMs based on NHPP which incorporates the testing–effort functions (TEF) have been proposed by many authors (Yamada et al., 1986; 1987; 1993; Yamada and Ohtera, 1990; Kapur and Garg, 1996; Kapur and Younes, 1994; Huang et al., 1997; 2007; Kuo et al., 2001; Huang and Kuo, 2002; Huang, 2005; Bokhari and Ahmad, 2006; 2007; Quadri et al., 2006; 2008; Ahmad et al., 2008; 2009; 2010). The testing-effort can be represented as the number of CPU hours, the number of executed test cases, etc. (Yamada and Osaki, 1985; Yamada et al., 1986, 1993). Most of these works on SRGMs modified the exponential NHPP growth model (Goel and Okumoto, 1979) and incorporated the concept of testing-effort into an NHPP model to describe the software fault detection phenomenon. Recently, Bokhari and Ahmad (2006) also proposed a new SRGM with the Log-logistic testing-effort function to predict the behavior of failure and fault of software. However, the exponential NHPP growth model is sometimes insufficient and inaccurate to analyze real software failure data for reliability assessment.

In this paper we show how to integrate a Log-logistic testing-effort function into inflection S-shaped NHPP growth models (Ohba, 1984; 1984a) to get a better description of the software fault detection phenomenon. The parameters of the model are estimated by Least Square Estimation (LSE) and Maximum Likelihood Estimation

Manuscript received January 5, 2011
Manuscript revised January 20, 2011

(MLE) methods. The statistical methods of data analysis are presented and the experiments are performed based on real data sets and the results are compared with other existing models. The experimental results show that the proposed SRGM with Log-logistic testing-effort function can estimate the number of initial faults better than that of other models and that the Log-logistic testing-effort functions is suitable for incorporating into inflection S-shaped NHPP growth model. Further, the analyses of the proposed models under imperfect debugging environment are also discussed.

2. SRGM with TEF

A software reliability growth model (SRGM) explains the time dependent behavior of fault removal. The objective of software reliability testing is to determine probable problems with the software design and implementation as early as possible to assure that the system meets its reliability requirements. Numerous SRGMs have been developed during the last three decades and they can provide very useful information about how to improve reliability (Musa et al., 1987; Xie, 1991; Lyu, 1996). Among these models, exponential growth model and inflection S-shaped growth model have been shown to be very useful in fitting software failure data. Many authors incorporated the concept of testing-effort into exponential type SRGM based on the NHPP to get a better description of the fault detection phenomenon.

The testing-effort indicates how the errors are detected effectively in the software and can be modeled by different distributions (Musa et al., 1987; Yamada et al., 1986; 1993; Kapur et al., 1999). Gokhale and Trivedi (1998) proposed the Log-logistic SRGM that can capture the increasing/decreasing nature of the failure occurrence rate per fault. Recently, Bokhari and Ahmad (2006), and Ahmad et al. (2010a) also presented how to use the Log-logistic curve to describe the time-dependent behavior of testing-effort consumptions during testing.

The Cumulative testing-effort expenditure consumed in (0,t] is depicted in the following:

$$W(t) = \alpha[1-\{1+(\beta t)^{\delta}\}^{-1}] = \alpha[(\beta t)^{\delta}/(1+(\beta t)^{\delta})], \tag{1}$$

Therefore, the current testing-effort expenditure at testing t is given by:

$$w(t) = [\alpha\beta\delta(\beta t)^{\delta-1}]/[1+(\beta t)^{\delta}]^2\ ,\ t>0, \tag{2}$$

where α is the total amount of testing-effort consumption required by software testing, β is the scale parameter, and δ is the shape parameter. The testing-effort $w(t)$ reaches its maximum value at time

$$t_{\max} = \frac{1}{\beta}\left(\frac{\delta-1}{\delta+1}\right)^{\frac{1}{\delta}}$$

The inflection S-shaped NHPP software reliability growth model is known as one of the flexible SRGMs that can depict both exponential and S-shaped growth curves depending upon the parameter values (Ohba, 1984 & Kapur et al., 2004). The model has been shown to be useful in fitting software failure data. Ohba proposed that the fault removal rate increases with time and assumed the presences of two types of errors in the software. Later, Kapur et al. (2004), Khan et al. (2008) and Ahmad et al. (2010; 2010a) modified the inflection S-shaped model and incorporated the testing-effort in an NHPP model. Therefore, we show how to incorporate Log-logistic testing-effort function into inflection S-shaped NHPP model.

The extended inflection S-shaped SRGM with Log-logistic testing-effort function is formulated on the following assumptions (Ohba, 1984; 1984a; Yamada and Osaki, 1985; Kapur et al., 1999; Kuo et al., 2001; Huang and Lo, 2006; Kapur et al., 2004; Ahmad et al., 2010; 2010a):

1. The software system is subject to failures at random times caused by errors remaining in the system.
2. Error removal phenomenon in software testing is modeled by NHPP.
3. The mean number of errors detected in the time interval $(t, t+\Delta t]$ by the current testing-effort expenditures is proportional to the mean number of detectable errors in the software.
4. The proportionality increases linearly with each additional error removal.
5. Testing-effort expenditures are described by the Log-logistic TEF.
6. Each time a failure occurs, the error causing that failure is immediately removed and no new errors are introduced.
7. Errors present in the software are of two types: mutually independent and mutually dependent.

The mutually independent errors lie on different execution paths, and mutually dependent errors lie on the same execution path. Thus, the second type of errors is detectable if and only if errors of the first type have been removed. According to these assumptions, if the error detection rate with respect to current testing-effort expenditures is proportional to the number of detectable errors in the software and the proportionality increases linearly with each additional error removal, we obtain the following differential equation:

$$\frac{dm(t)}{dt}\times\frac{1}{w(t)} = \phi(t)\left(a-m(t)\right) \tag{3}$$

where

$$\phi(t) = b\left[r + (1-r)\frac{m(t)}{a} \right],$$

$r\ (>0)$ is the inflection rate and represents the proportion of independent errors present in the software, $m(t)$ be the mean value function (MVF) of the expected number of errors detected in time $(0,t]$, $w(t)$ is the current testing-effort expenditure at time t, a is the expected number of errors in the system, and b is the error detection rate per unit testing-effort at time t.

Solving (3) with the initial condition $t=0,\ W(t)=0,\ m(t)=0$, we obtain the MVF

$$m(t) = \frac{a\left[1-e^{-bW(t)}\right]}{1+((1-r)/r)e^{-bW(t)}} \quad (4)$$

If the inflection rate r = 1, the above NHPP model becomes equivalent to the exponential growth model.

The failure intensity at testing time t of the inflection S-shaped NHPP model with testing-effort is given by

$$\lambda(t) = \frac{dm(t)}{dt} = \frac{a \cdot b \cdot w(t) \cdot e^{-bW(t)}}{r\left[1+((1-r)/r)e^{-bW(t)}\right]^2}.$$

Furthermore, we describe a flexible SRGM with mean value function considering the Log-logistic testing-effort expenditure as

$$m(t) = \frac{a\left[1-e^{-b\alpha[(\beta t)^{\delta}/(1+(\beta t)^{\delta})]}\right]}{1+((1-r)/r)e^{-b\alpha[(\beta t)^{\delta}/(1+(\beta t)^{\delta})]}} \quad (5)$$

In addition, the expected number of errors to be detected eventually is

$$m(\infty) = \frac{a\left[1-e^{-b\alpha}\right]}{1+((1-r)/r)e^{-b\alpha}}.$$

2.1 Imperfect-Software Debugging Models

An NHPP model is said to have perfect debugging assumption when $a(t)$ is constant, i.e., no new faults are introduced during the debugging process. An NHPP SRGM subject to imperfect debugging was introduced by the authors with the assumption that if detected faults are removed, then there is a possibility that new faults with a constant rate γ are introduced (see Ahmad et al. 2010; Shyur, 2003; Lo and Huang, 2004; Pham, 2007; Pham et al., 1999; Yamada et al., 1992; Zhang et al., 2003; Xie and Yang, 2003).

Let $n(t)$ be the number of errors to be eventually detected plus the number of new errors introduced to the system by time t, we obtain the following system of differential equations:

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = \phi(t)\left(n(t)-m(t)\right), \quad (6)$$

and $\dfrac{dn(t)}{dt} = \gamma \dfrac{dm(t)}{dt}$, (7)

where $\phi(t) = b\left[r + (1-r)\dfrac{m(t)}{n(t)} \right]$

and $n(0) = a$.

Solving the above differential equations under the boundary conditions $m(0)=0$ and $W(0)=0$, we can obtain the following MVF of inflection S-shaped model with Log-logistic testing-effort under imperfect debugging.

$$m(t) = \frac{a\left[1-e^{-b(1-r\gamma)W(t)}\right]}{(1-\gamma)+\left[(1-r)/r\right]e^{-b(1-r\gamma)W(t)}} \quad (8)$$

We also have

$$n(t) = \frac{a\left[1+\left(((1-r)/r)-\gamma\right)e^{-b(1-r\gamma)W(t)}\right]}{\left(1-\gamma\right)+\left[(1-r)/r\right]e^{-b(1-r\gamma)W(t)}} \quad (9)$$

Thus, the failure intensity function $\lambda(t)$ is given by

$$\lambda(t) = \frac{ab\left[1-r\gamma\right]w(t)e^{-b(1-r\gamma)W(t)}}{(1-\gamma)\left[r(1-\gamma)+(1-r)e^{-b(1-r\gamma)W(t)}\right]} \quad (10)$$

The expected number of remaining errors after testing time t is

$$m_{remaning}(t) = \frac{a(1-r\gamma)e^{-b(1-r\gamma)W(t)}}{r(1-\gamma)+(1-r)e^{-b(1-r\gamma)W(t)}} \quad (11)$$

3. Estimation of Model Parameters

The parameters of the SRGM are estimated based upon the failure data collected by the MLE and LSE techniques (Musa et al., 1987; Musa, 1999; Lyu, 1996; Ahmad et al. 2008). The performance of the proposed model is then compared with other existing models. Experiments on three real software failure data are performed.

3.1 Least Square Method

The parameters α, β, and δ in the Log-logistic TEF can be estimated using the method of LSE. These parameters are determined for n observed data pairs in the form $(t_k, W_k)\ (k=1,2,....,n;\quad 0<t_1<...<t_n)$ where W_k is the cumulative testing-effort consumed in time $(0,t_k]$. The estimators $\hat{\alpha},\ \hat{\beta}$, and $\hat{\delta}$, which contribute the model with a greater fitting, can be obtained by minimizing:

$$S(\alpha,\beta,\delta) = \sum_{k=1}^{n}[\ln W_k - \ln\alpha - \delta\ln(\beta t_k) + \ln[1+(\beta t_k)^{\delta}]]^2 \quad (12)$$

Differentiating S with respect to α, β, and δ, setting the partial derivatives to zero, the set of nonlinear equations are obtained respectively.

$$\frac{\partial S}{\partial \alpha} = 2\sum_{k=1}^{n}[\ln W_k - \ln\alpha - \delta\ln(\beta t_k) + \ln[1+(\beta t_k)^{\delta}].\frac{-1}{\alpha} = 0$$

Thus, the LSE of α is given by

$$\hat{\alpha} = e^{\sum_{k=1}^{n}\ln W_k - \delta\sum_{k=1}^{n}\ln(\beta t_k) + \sum_{k=1}^{n}\ln(1+(\beta t_k)^{\delta})\Big/ n}$$

The LSE of β and δ can be obtained numerically by solving the following equations:

$$\frac{\partial S}{\partial \beta} = \sum_{k=1}^{n}[\ln W_k - \ln\alpha - \delta\ln(\beta t_k) + \frac{\ln[1+(\beta t_k)^{\delta}]}{(1+(\beta t_k)^{\delta})} = 0$$

and

$$\frac{\partial S}{\partial \delta} = \sum_{k=1}^{n}[\ln W_k - \ln\alpha - \delta\ln(\beta t_k) + \frac{\ln[1+(\beta t_k)^{\delta}]\ln(\beta t_k)}{1+(\beta t_k)^{\delta}} = 0$$

3.2 Maximum Likelihood Method

Once the estimates of α, β, and δ are known, the parameters of the SRGMs can be estimated through MLE method. The estimators of a, b, and r are determined for the n observed data pairs in the form (t_k, y_k) $(k=1,2,...,n;\ 0<t_1<t_2<......<t_n)$ where y_k is the cumulative number of software errors detected up to time t_k or $(0,t_k]$. Then the likelihood function for the unknown parameters a, b, and r in the NHPP model (5) is given by (Musa et al., 1987)

$$L'(a,b,r) = \prod_{k=1}^{n}\frac{\left[m(t_k)-m(t_{k-1})\right]^{(y_k-y_{k-1})}}{(y_k-y_{k-1})!}.e^{-[m(t_k)-m(t_{k-1})]}$$

where $t_0 \equiv 0$ and $y_0 \equiv 0$. (13)

The maximum likelihood estimates of SRGM parameters a, r and b can be obtained by solving the following three equations.

$$\hat{a} = \frac{y_n\left[1+\lambda\phi_n\right]}{1-\phi_n}$$

$$\frac{y_n}{r^2\left[1+\lambda\right]} + \frac{a\phi_n\left[1-\phi_n\right]}{\left[r^2\left[1+\lambda\phi_n\right]\right]^2} = \sum_{k=1}^{n}\frac{\left(y_k-y_{k-1}\right)\phi_k}{r^2\left[1+\lambda\phi_k\right]} + \sum_{k=1}^{n}\frac{\left(y_k-y_{k-1}\right)\phi_{k-1}}{r^2\left[1+\lambda\phi_{k-1}\right]}$$

$$\sum_{k=1}^{n}\frac{\left(y_k-y_{k-1}\right)\left[W\left(t_k\right)\phi_k - W\left(t_{-1}\right)\phi_{k-1}\right]}{\phi_{k-1}-\phi_k} + \sum_{k=1}^{n}\frac{\left(y_k-y_{k-1}\right)\left[\lambda W\left(t_k\right)\phi_k\right]}{1+\lambda\phi_k} + \sum_{k=1}^{n}\frac{\left(y_k-y_{k-1}\right)\left[\lambda W\left(t_{k-1}\right)\phi_{k-1}\right]}{1+\lambda\phi_{k-1}} = \frac{aW\left(t_n\right)\phi_n\left[1+\lambda\right]}{\left[1+\lambda\phi_{k-1}\right]^2}$$

where $\phi_k = e^{-bW(t_k)}$, $k = 1,2,...n$, and $\lambda = \frac{1-r}{r}$

The above can be solved by numerical methods to get the values of $\hat{a}$, $\hat{b}$ and $\hat{r}$.

4. Data Analysis and Experiments

4.1 Comparison Criteria

To check the performance of proposed SRGM with Log-logistic TEF, we use the following four criteria:

1. The Accuracy of Estimation (AE) is defined (Musa et al., 1987; Kuo et al., 2001) as
$$AE = \left|\frac{M_a - a}{M_a}\right|,$$
where M_a is the actual cumulative number of detected errors after the test, and a is the estimated number of initial errors. For practical purposes, M_a is obtained from software error tracking after software testing.
2. The Mean of Squared Errors (MSE) (Long-term predictions) is defined as
$$MSE = \frac{1}{k}\sum_{i=1}^{k}\left[m(t_i) - m_i\right]^2,$$
Where $m(t_i)$, is the expected number of errors at time t_i estimated by a model, and m_i is the expected number of errors at time t_i. MSE gives a quantitative comparison for long-term predictions. A smaller MSE indicates a minimum fitting error and better performance.
3. The coefficient of multiple determinations is defined (Musa et al., 1987; Musa, 1999) as
$$R^2 = \frac{S\left(\hat{\hat{\alpha}},0,1\right) - S\left(\alpha,\beta,\delta\right)}{S\left(\hat{\hat{\alpha}},0,1\right)},$$
where $\hat{\hat{\alpha}}$ is the LSE of α for the model with only a constant term, that is, $\beta = 0$, and $\delta = 1$ in (12). It is given by $\ln\hat{\hat{\alpha}} = \frac{1}{n}\sum_{k=1}^{n}\ln W_k$. Therefore, R^2 measures the percentage of total variation about the mean accounted for by the fitted model and tells us how well a curve fits the data. It is frequently employed to compare models and assess which model provides the best fit to the data. The best model is the one which

provides the higher R^2, that is, closer to 1 (Kumar et al., 2005; Ahmad et al., 2008). To investigate whether a significant trend exists in the estimated testing-effort, one could test the hypotheses $H_0 : \beta = 0$ and $\delta = 1$, against $H_1 : \beta \neq 0$ or at least $\delta \neq 1$ using F -test by merely forming the ratio

$$F = \frac{\left[S\left(\hat{\alpha},0,1\right) - S\left(\alpha,\hat{\beta},\hat{\delta}\right)\right]/2}{S\left(\hat{\alpha},0,1\right)/(n-3)}.$$

If the value of F is greater than $F_\alpha(2, n-3)$, which is the α percentile of the F distribution with degrees of freedom 2 and $n-3$, we can be $(1-\alpha)100$ percent confident that H_0 should be rejected, that is, there is a significant trend in the testing-effort curve.

4. The Predictive Validity is defined (Musa et al., 1987; Musa, 1999) as the capability of the model to predict future failure behavior from present and past failure behavior. Assume that we have observed q failures by the end of test time t_q. We use the failure data up to time t_e $(\leq t_q)$ to determine the parameters of $m(t)$. The ratio

$$\frac{\hat{m}(t_q) - q}{q}$$

is called the relative error. Values close to zero for relative error indicate more accurate prediction and hence a better model. We can visually check the predictive validity by plotting the relative error for normalized test time t_e/t_q.

4.2 Numerical Examples

Data Set1: The first set of actual data is from the study by Ohba (1984). The system is PL/1 data base application software, consisting of approximately 1,317,000 lines of code. During nineteen weeks of testing, 47.65 CPU hours were consumed and about 328 software errors were removed. Moreover, the total cumulative number of detected faults after a long time of testing was 358. The estimated parameters $\alpha, \beta, and\ \delta$ of the Log-logistic TEF are:

$$\hat{\alpha} = 1451.2265, \quad \hat{\beta} = 0.0026 \quad \delta = 1.1160$$

Figure 1, shows the fitting of the estimated testing-effort by using above estimates. The fitted curves and the actual software data are shown by solid and dotted lines, respectively. The estimated values of the parameters $a, b, \text{and}\ r$ in (5) are:

$$\hat{a} = 385.6254, \quad \hat{b} = 0.0622 \quad r = 0.3689$$

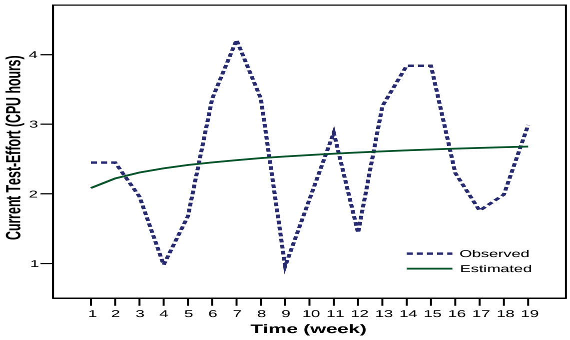

Figure 1: Observed/estimated current testing-effort function vs. time

Figure 2, illustrates a fitted curve of the estimated cumulative failure curve with the actual software data. The R^2 value for proposed Log-logistic TEF is 0.99574. Therefore, it can be said that the proposed curve is suitable for modeling the software reliability. Also, the calculated value $F\ (=4.9787)$ is greater than $F_{0.05}(2,16)$. Therefore, it can be concluded that the proposed model is suitable for modeling the software reliability and the fitted testing-effort curve is highly significant for this data set.

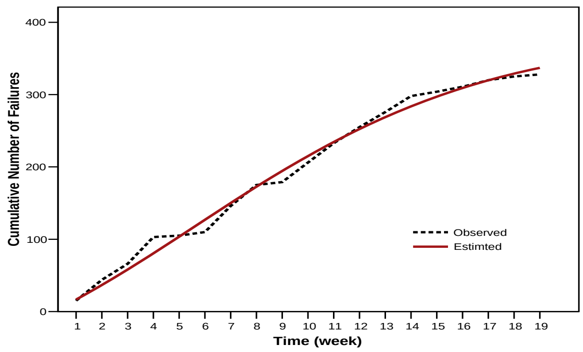

Figure 2: Observed/estimated cumulative number of failures vs. time

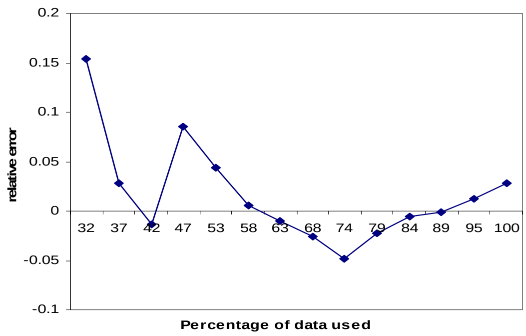

Figure 3: Predictive Relative Error Curve

Table I lists the comparisons of proposed model with different SRGMs which reveal that the proposed model has better performance. Kolmogorov Smirnov goodness-of-fit test shows that the proposed SRGM fits pretty well at the 5 percent level of significance.
Finally, the relative error in prediction of proposed model for this data set (DS) is calculated and illustrated by figure 3. It is observed that relative error approaches zero as t_e approaches t_q and the error curve is usually within ± 5 percent.

Table I: Comparison results of different SRGMs for DS1

Model	a	r	b	AE %	MSE
Proposed model	385.63	0.37	0.0622	7.54	87.69
Bokhari Log-logistic model	565.73		0.0196	58.02	116.74
Huang Logistic model	394.08		0.0427	10.06	118.59
Yamada Rayleigh model	459.08		0.0273	28.23	268.42
Yamada Weibull model	565.35		0.0196	57.91	122.09
Yamada delayed S-shaped model	374.05		0.1976	4.48	168.67
Delayed S-Shaped with Logistic TEF	346.55		0.0936	3.20	147.61
Inflection S-shaped model	389.1	0.2	0.0935	8.69	133.53
G-O model	760.0		0.0323	112.29	139.82

Therefore, Figures 1 to 3 and Table I reveals that the proposed model has better performance than the other models. This model fits the observed data better, and predicts the future behavior well.

Data set2: The second set of actual data relates to the Release of Tandem Computer Project cited in Wood (1996) from a subset of products for four separate software releases. In this research only Release 1 is used for illustrations. There were 10000 CPU hours consumed, over the 20 week of testing and 100 software errors were removed. The estimated parameters of the TEF are obtained as:

$$\hat{\alpha} = 15808.0743, \quad \hat{\beta} = 0.0704, \quad \delta = 1.5689$$

Figure 4 illustrates the comparisons between the observed failure data and the estimated Log-logistic testing-effort data. Here, the fitted curves are shown as a solid and dotted line represents actual software data. The estimated parameters of the SRGM (5) are:

$$\hat{a} = 197.4934, \quad \hat{b} = 1.05\times10^{-5}, \quad r = 9.934$$

Figure 5 illustrates a fitted curve of the estimated cumulative failure curve with the actual software data. The R^2 also known as the coefficient of determination, depicts how well a curve fits the data. A fit is more reliable when the value is closer to 1. The R^2 value for proposed Log-logistic TEF is 0.9976, which is very close to one. Moreover, the value of MSE is 18.04, which is very small compared to other SRGM. Table II lists the comparisons of proposed model with different SRGMs which reveal that the proposed model has better performance. It can therefore be observed that the Log-logistic TEF is suitable for modeling the proposed SRGM of this data set. Also the fitted testing-effort curve is significant since the calculated value $F(=5.3204)$ is greater than $F_{0.05}(2,17)$. Kolmogorov Smirnov goodness-of-fit test shows that the proposed SRGM fits pretty well at the 5 percent level of significance.

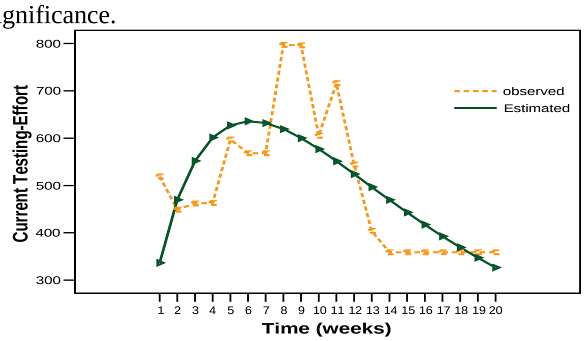

Figure 4: Observed/estimated current testing-effort vs. Time

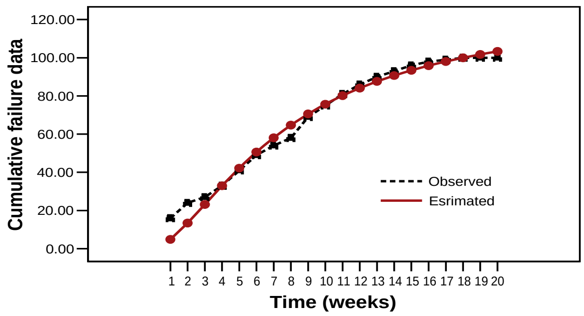

Figure 5: Observed/estimated cumulative number of failures vs. time

Table II. Comparison results of different SRGMs for DS2

Model	a	r	b	MSE
Proposed model	197.49	9.934	0.0000105	18.04
Bokhari Log-logistic model	135.91		0.000142	19.80
Huang Logistic model	107.66		0.000266	22.76
Yamada Rayleigh model	110.60		0.000226	39.69
Yamada Weibull model	135.74		1.42E-04	18.43
Yamada delayed S-shaped model	102.26		0.345	94.99

Delayed S-shaped model with Logistic TEF	101.86		0.00063	92.66
Inflection S-Shaped model	507.63	1.695	0.00814	87.45
G-O model	137.072		0.05154	25.33

Following the work of Musa et al. (1987), the relative error in prediction for this data set is computed and the results are plotted in Figure 6. Figures 4-6 and Table II show that the proposed model has better performance and predicts the future behavior well.

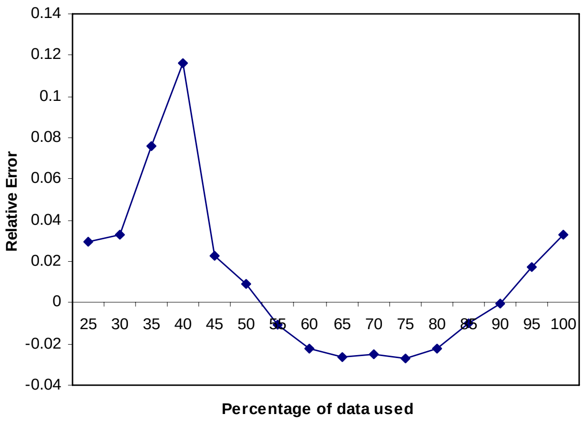

Figure 6: Predictive Relative Error Curve

Data set3: The third set of actual data in this research is the System T1 data of the Rome Air Development Center (RADC) projects and cited from Musa et al. (1987), Musa (1999). The number of object instructions in System T1 which is a real-time command and control application is 21,700. The software was tested by nine testers over the period of 21 weeks. Through the testing phase, about 25.3 CPU hours were consumed and 136 software errors are removed. The number of errors removed after 3.5 years of test was reported to be 188 (Huang, 2005). The estimated parameters $\alpha, \beta, \text{and}\, \delta$ of the Log-logistic testing-effort function are:

$$\hat{\alpha} = 33.110503,\ \hat{\beta} = 0.056547,\ \delta = 7.151126 .$$

Figure 7 shows the fitting of the estimated testing-effort by using above estimates. The fitted curve and the actual software data are shown by solid and dotted lines, respectively. The estimated values of the parameters $a, b, \text{and}\, r$ in (5) are:

$$\hat{a} = 161.024306,\ \hat{b} = .0010648,\ r = 168.36859 .$$

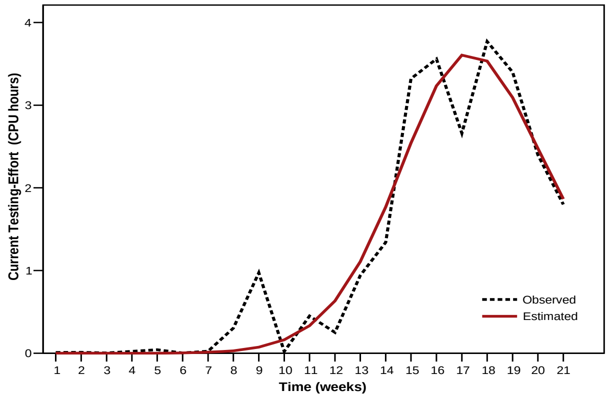

Figure 7: Observed/estimated current testing-effort function vs. time

Figure 8, illustrates a fitted curve of the estimated cumulative failure curve with the actual software data. The R^2 value for proposed Log-logistic testing-effort is 0.9973.

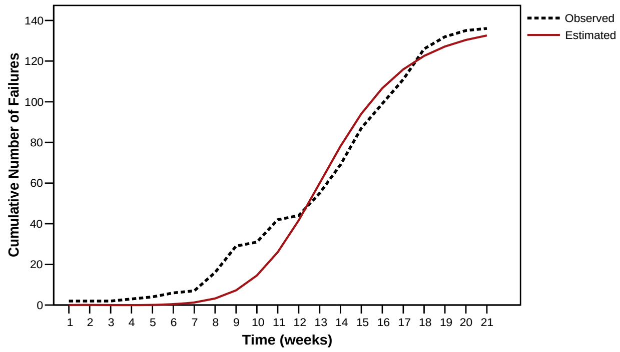

Figure 8: Observed/estimated cumulative number of failures vs. time

Therefore, it can be said that the proposed curve is suitable for modeling the software reliability. Also, the calculated value $F(=5.6512)$ is greater than $F_{0.05}(2,18)$ and $F_{0.01}(2,18)$, which concludes that the fitted testing-effort curve is highly for this data set. Table III lists the comparisons of proposed model with different SRGMs which reveal that the proposed model has better performance. Kolmogorov Smirnov goodness-of-fit test shows that the proposed SRGM fits pretty well at the 5 percent level of significance.

Lastly, the relative error in prediction of proposed model for this data set is calculated and Figure 9 shows the relative error plotted against the percentage of data used (that is, t_e / t_q). It is observed that relative error of the proposed model approaches zero as t_e approaches t_q and the error curve is usually within ± 5 percent.

Table III. Comparison results of different SRGMs for DS3

Model	a	r	b	AE (%)	MSE
Proposed model	161.02	168.37	0.0011	14.36	75.19
Bokhari Log-logistic model	133.28		0.1571	29.11	100.18
Huang Logistic model	138.03		0.1451	26.58	64.41
Yamada Rayleigh model	866.94		0.0096	25.11	89.241
Yamada Weibull model	133.71		0.155	28.88	81.51
Yamada delayed S-shaped model	237.19		0.0963	26.16	245.25
Delayed S-Shaped model with Logistic TEF (Huang)	124.11		0.411	33.98	180.02
Inflection S-shaped model	159.11		0.0765	15.36	118.3
G-O model	142.32		0.1246	24.29	2438.3

Consequently, from the Figures 7 to 9 and Table III discussed, it can be concluded that the proposed model gets reasonable prediction in estimating the number of software errors and fits the observed data better than the others.

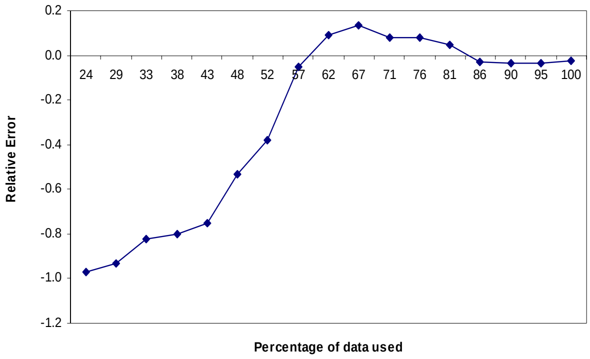

Figure 9: Predictive Relative Error Curve

4.3 Numerical Example on Imperfect Debugging

We consider the DS1 to discuss the issue of imperfect debugging for proposed SRGM. In order to validate the proposed SRGM under imperfect debugging, MSE is selected as the evaluation criterion. The parameters $a, b, r,$ and γ in (8) can be solved by the method of MLE.

Table IV. Comparison results of different SRGMs under Imperfect Debugging

Models	a	r	b	γ	MSE
Proposed model	385.63	0.37	0.0630	0.0336	87.69
Bokhari Log-logistic model	565.73		0.0196	58.02	116.74
Huang Logistic model	391.62		0.0420	0.0116	114.09
Yamada Rayleigh model	399.02		0.0316	0.123	268.55
Yamada Weibull model	565.35		0.0196	57.91	122.09
Yamada delayed S-shaped	374.87		.1968	0.962	168.71
Delayed S-Shaped model with Logistic TEF (Huang)	346.55		0.124	0.0115	147.60
Inflection S-Shaped model	387.95	0.34	0.201	0.330	133.54
G-O model	530.61		0.0463	0.287	222.09

Table IV shows the estimated parameters with MSE of the proposed SRGM and some selected models for comparison under imperfect debugging. We observed that the value of MSE of the proposed SRGM with Log-logistic testing-effort function is the lowest among all the models considered. Moreover, the estimated values γ of all the models is close to but not equal to zero, thus the fault removal phenomenon may not be pure perfect debugging process.

A fitted curve of the estimated cumulative number of failures with the actual software data and the RE curve for the proposed SRGM with Log-logistic testing-effort function under imperfect debugging is illustrated by Figure 10 and 11.

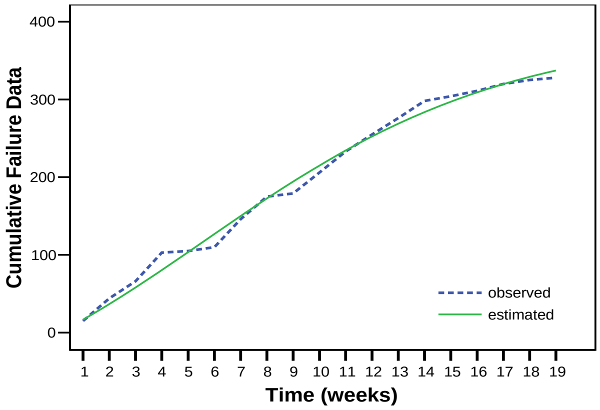

Figure 10: Observed/estimated cumulative number of failures vs. time

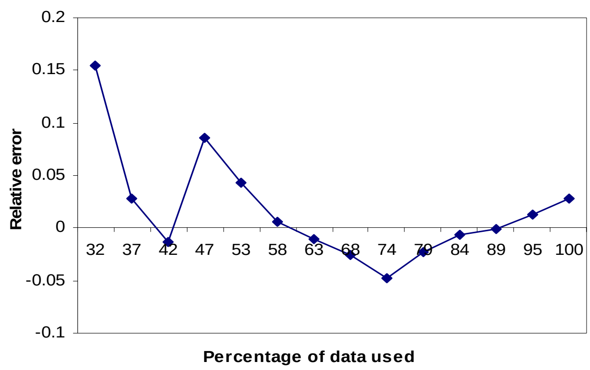

Figure 11: Predictive Relative Error Curve

5. Conclusion

In this paper, we have proposed a flexible SRGM based on NHPP model, which incorporates Log-logistic testing-effort function into inflection S-shaped model. The performance of the proposed SRGM is compared with other traditional SRGMs using different criteria. The results obtained show better fit and wider applicability of the proposed model on different types of real data applications. We conclude that the proposed flexible SRGM has better performance as compare to the other SRGMs and gives a reasonable predictive capability for the real failure data. We also conclude that the incorporated Log-logistic testing-effort function into inflection S-shaped model is a flexible and can be used to describe the actual expenditure patterns more faithfully during software development. In addition, the proposed models under imperfect debugging environment are also discussed.

Acknowledgements

The authors are thankful to the reviewers for his valuable comments and suggestions that helped us to improve the presentation of the paper.

References

[1] Ahmad, N., Bokhari, M.U., Quadri, S.M.K. and Khan, M.G.M. (2008), “The Exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis”, International Journal of Quality and Reliability Management, Vol. 25 (2), pp. 211-235.

[2] Ahmad, N., Khan, M. G. M., Quadri, S. M. K. and Kumar, M. (2009), “Modeling and analysis of software reliability with Burr type X testing-effort and release-time determination”, Journal of Modelling in Management, Vol. 4 (1), pp. 28-54.

[3] Ahmad, N., Khan, M. G. M and Rafi, L. S. (2010), “A Study of Testing-Effort Dependent Inflection S-Shaped Software Reliability Growth Models with Imperfect Debugging”, International Journal of Quality and Reliability Management, Vol. 27 (1), 89-110.

[4] Ahmad, N., Khan, M. G. M and Rafi, L. S. (2010a), “Software Reliability Modeling Incorporating Log-Logistic Testing-Effort with Imperfect Debugging”, in Proceedings of the International Conference on Modeling, Optimization and Computing (ICMOC-2010), Durgapur, India, Published by American Institute of Physics, pp. 651-657.

[5] Bokhari, M.U. and Ahmad, N. (2006), “Analysis of a software reliability growth models: the case of log-logistic test-effort function”, In: Proceedings of the 17th IASTED International Conference on Modeling and Simulation (MS’2006), Montreal, Canada, pp. 540-545.

[6] Bokhari, M.U. and Ahmad, N. (2007), “Software reliability growth modeling for Exponentiated Weibull functions with actual software failures data”, Advances in Computer Science and Engineering: Reports and Monographs, World Scientific Publishing Company, Singapore, Vol. 2, pp. 390-396.

[7] Goel, A.L. and Okumoto, K. (1979), “Time dependent error-detection rate model for software reliability and other performance measures”, IEEE Transactions on Reliability, Vol. R- 28, No. 3, pp. 206-211.

[8] Gokhale, S.S. and Trivedi, K.S. (1998), “Log-logistic software reliability growth model”, In: Proceedings of the 3rd IEEE International High Assurance Systems Engineering Symposium (HASE’ 98), Washington, USA, pp. 34-41.

[9] Huang, C.Y. (2005), “Performance analysis of software reliability growth models with testing-effort and change-point”, Journal of Systems and Software, Vol. 76, pp. 181-194.

[10] Huang, C.Y. and Kuo, S.Y. (2002), “ Analysis of incorporating logistic testing-effort function into software reliability modeling”, IEEE Transactions on Reliability, Vol. 51, no. 3, pp. 261-270.

[11] Huang, C.Y., Kuo, S.Y. and Chen, I.Y. (1997), “Analysis of software reliability growth model with logistic testing-effort function”, In: Proceeding of 8th International Symposium on

Software Reliability Engineering (ISSRE'1997), Albuquerque, New Mexico, pp. 378-388.
[12] Huang, C.Y., Kuo, S.Y. and Lyu, M.R. (2007), "An assessment of testing-effort dependent software reliability growth models", IEEE Transactions on Reliability, Vol. 56, no.2, pp. 198-211.
[13] Kapur, P.K. and Garg, R.B. (1996), "Modeling an imperfect debugging phenomenon in software reliability", Microelectronics and Reliability, Vol. 36, pp. 645-650.
[14] Kapur, P.K., Garg, R.B. and Kumar, S. (1999), Contributions to Hardware and Software Reliability, World Scientific, Singapore.
[15] Kapur, P.K., Jha, P.C., and Bardhan, A.K. (2004), "Optimal Allocation of Testing Resource for a Modular Software", Asia-Pacific Journal of Operational Research, Vol. 21, no. 3, pp. 333-354.
[16] Kapur, P.K. and Younes, S. (1994), "Modeling an imperfect debugging phenomenon with testing effort", In: Proceedings of 5th International Symposium on Software Reliability Engineering (ISSRE'1994), pp. 178-183.
[17] Khan, M. G. M., Ahmad, N., and Rafi, L. S. (2008), "Optimal Testing Resource Allocation for Modular Software Based on a Software Reliability Growth Model: a Dynamic Programming Approach", in: Proceedings of the International Conference on Computer Science and Software Engineering (CSSE-2008), Wuhan, China, IEEE Computer Society, pp. 759-762.
[18] Kumar, M., Ahmad, N. and Quadri, S.M.K. (2005), "Software reliability growth models and data analysis with a Pareto test-effort", RAU Journal of Research, Vol., 15 (1-2), pp. 124-128.
[19] Kuo, S.Y., Hung, C.Y. and Lyu, M.R. (2001), "Framework for modeling software reliability, using various testing-efforts and fault detection rates", IEEE Transactions on Reliability, Vol. 50, no.3, pp 310-320.
[20] Lo, J.H. and Huang, C.Y. (2004), "Incorporating imperfect debugging into software fault correction process", In: Proceeding of the IEEE Regional 10 Annual International Conference (TENCON'2004), Chiang Mai, Thailand, pp. 326-329.
[21] Lyu, M.R. (1996), Handbook of Software Reliability Engineering, McGraw- Hill.
[22] Musa J.D. (1999), Software Reliability Engineering: More Reliable Software, Faster Development and Testing, McGraw-Hill.
[23] Musa, J.D., Iannino, A. and Okumoto, K. (1987), Software Reliability: Measurement, Prediction and Application, McGraw-Hill.
[24] Ohba, M. (1984), "Software reliability analysis models" IBM Journal. Research Development, Vol. 28, no. 4, pp. 428-443.
[25] Ohba, M. (1984a), "Inflection S-shaped software reliability growth models", Stochastic Models in Reliability Theory (Osaki, S. and Hatoyama, Y. Editors), pp. 144-162, Springer-Verlag Merlin.
[26] Pham, H. (2000), Software Reliability, Springer-Verlag, New York.
[27] Pham, H. (2007), "An imperfect-debugging fault-detection dependent-parameter software", International Journal of Automation and Computing, Vol. 4, no. 4, pp. 325-328.
[28] Pham, H., Nordmann L. and Zhang, X. (1999), "A general imperfect software debugging model with S-shaped fault detection rate", IEEE Transactions on Reliability, Vol. R-48, no. 2, pp. 169-175.
[29] Quadri, S. M. K., Ahmad, N., and Peer, M. A. (2008), "Software Optimal Release Policy and Reliability Growth Modeling", In the Proceedings of 2nd National Conference on Computing for Nation Development, INDIACom-2008, India, pp. 423-431.
[30] Quadri, S.M.K., Ahmad, N., Peer, M.A. and Kumar, M. (2006), "Nonhomogeneous Poisson process software reliability growth model with generalized exponential testing effort function", RAU Journal of Research, Vol., 16 (1-2), pp. 159-163.
[31] Shyur, H.J. (2003), "A stochastic software reliability model with imperfect and change point", The Journal of Systems and Software, Vol. 66, pp. 135-141.
[32] Wood, A. (1996), "Predicting software reliability", IEEE Computers, 11, pp. 69-77.
[33] Xie, M. (1991), Software Reliability Modeling, World Scientific Publication, Singapore.
[34] Xie, M., and Yang, B. (2003), "A study of the effect of imperfect debugging on software development cost", IEEE Transaction on Software Engineering, Vol. SE-29, no. 5, pp. 471-473.
[35] Yamada, S., Hishitani J. and Osaki, S. (1993), "Software reliability growth model with Weibull testing-effort: a model and application", IEEE Transactions on Reliability, Vol. R-42, pp. 100-105.
[36] Yamada, S. and Ohtera, H. (1990), "Software reliability growth models for testing effort control", European Journal of Operational Research, Vol. 46, 3, pp. 343-349.
[37] Yamada, S., Ohtera, H. and Norihisa, H. (1986), "Software reliability growth model with testing-effort", IEEE Transactions on Reliability, Vol. R-35, no. 1, pp.19-23.
[38] Yamada, S., Ohtera, H. and Norihisa, H. (1987), "A testing-effort dependent software reliability model and its application", Microelectronics and Reliability, Vol. 27, no. 3, pp. 507-522.
[39] Yamada, S., and Osaki, S. (1985), "Software reliability growth modeling: models and applications", IEEE Transaction on Software Engineering, Vol. SE-11, no. 12, pp. 1431-1437.
[40] Yamada, S., Tokuno, K. and Osaki, S. (1992), "Imperfect debugging models with fault introduction rate for software reliability assessment", International Journal of Systems Science, Vol. 23, no. 12, pp. 2241-2252.
[41] Zhang, X., Teng, X. and Pham, H. (2003), "Considering fault removal efficiency in software reliability assessment", IEEE Transaction on System, Man, and Cybernetics - Part A, Vol. 33, no. 1, pp. 114-120.

Nesar Ahmad is a Sr. Lecturer in University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, India. He received the B. Sc. degree in Mathematics from Bihar University, India, in 1984 and the M. Sc., M. Phil., and Ph. D. degree in Statistics from Aligarh Muslim University, Aligarh, India, in 1987, 1990, and 1993, respectively. From 1995 to 1996 he was a research associate of UGC/CSIR at Aligarh Muslim University, Aligarh. He has been a Lecturer in Statistics from January 2006 to December 2009 at the University of the South Pacific, Fiji Islands. After working five months as a lecturer at Poona College, Pune, he joined the University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, India in December 1996. Presently he is Senior Lecturer since 2000. He has about 16 years of experience in teaching and research. His research interests include life testing, statistical modeling, reliability analysis, software reliability analysis and optimization. He is an author and coauthor of many journal papers and conference papers in these areas. Dr. Ahmad is life member of Indian Science Congress and Bihar Journal of Mathematical Society, and regular member of Aligarh Statistical Association.

Dr. MGM Khan is an Associate Professor in statistics at the University of the South Pacific, Suva, Fiji. He has been in the teaching profession since 1996 when he first joined as a Lecturer in Statistics at Poona College, India. Dr. Khan graduated from the University of Calcutta in 1985. He obtained his M.Sc. from the Aligarh Muslim University in 1989. In 1992 he awarded M.Phil. Degree in Statistics followed by Ph.D. Degree in 1996 from the Aligarh Muslim University. Since 1998, he has been a Lecturer at Mohamed Sathak College, Chennai, India. In 2001 he joined at the University of the South Pacific (USP) as a Lecturer in Statistics. Presently, he is also the acting Head of the School of Computing, Information and Mathematical Sciences at USP. His research interest is in the fields of mathematical programming, sampling, statistical modeling, reliability analysis and software reliability analysis. He has published numerous articles in statistical journals. He has also presented several research papers in international conferences. Dr. Khan is an elected member of the International Statistical Institute and also active member of International Association of Survey Statistician and American Statistical Association.

Loriza Shazmin Rafi is a Tutor in Mathematics and Statistics in the College of Foundation Studies at the University of the South Pacific, Suva, Fiji. She has been teaching Mathematics/Statistics as a tutor since 2004 at this university. She graduated from the University of the South Pacific with a B.Sc. in Mathematics and Chemistry in 2002, Post-graduate Diploma in Mathematics in 2007 and Masters in Mathematics in 2010. At present, she is pursuing her Masters in Mathematics at this university. Her research interest is in the fields of Software Reliability Analysis & Optimization. She is coauthor of a few journal articles and conference papers published in conjunction with her M.Sc. supervisors.