Software Reliability Growth Models for the Safety Critical Software with Imperfect Debugging

O.Naga Raju,
Assist Professor,
Department of Computer Science &Engineerings,
Acharya Nagarjuna University, Nagarjuna Nagar-522510, India.
onrajunrt@gmail.com

Abstract. In this paper, we will investigate how to perform he log-logistic testing-effort function (TEF) into different 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 and the experimental results from actual data applications are analyzed. Results are compared with the other existing models to show that the proposed models can give fairly better predictions. It is shown that the log-logistic TEF is suitable for incorporating into inflection S-shaped NHPP growth models. In addition, the proposed models are also 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

Software reliability is defined as the probability of failure-free operation of a computer program for a specified time in a specified environment [1, 3, and 5]. A common approach for measuring software reliability is by using an analytical model whose parameters are generally estimated from available data on software failures [9], and [10]. Hence, software reliability is a key factor in software development process. SRGM is a mathematical expression of the software error occurrence and the removal process.

Since the early 1970's, many SRGMs have been proposed [8] and [10]. A NHPP as the stochastic process has been widely used in SRGM. In the past years, several SRGMs based on NHPP which incorporates the TEF have been proposed by many authors [2], [4], [6], [7], [8], [14], [15]. The testing-effort can be represented as the number of CPU hours, the number of executed test cases, etc. [14] and [15]. Recently, [1] and [3] also proposed a new SRGM with the Exponentiated Weibull (EW) testing-effort functions 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 TEF into inflection S-shaped NHPP growth models [11] and [12] to get a better description of the software fault detection phenomenon. The proposed framework is a generalization over the previous works on SRGMs with testing-efforts. The parameters of the model are estimated by LSE and 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. Further, the analyses of the proposed models under imperfect debugging environment are also discussed.

2. Log-Logistic Test Effort Function

The inflection S-shaped NHPP SRGM is known as one of the flexible SRGMs that can depict both exponential and S-shaped growth curves depending upon the parameter values [11]. 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, [3] modified the inflection S-shaped model and incorporated the TEF in an NHPP model. Therefore, we show how to incorporate Log-Logistic TEF into inflection S-shaped NHPP model.

The extended inflection S-shaped SRGM with Log-Logistic TEF is formulated on the following assumptions

[3], [4] and [11]:

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 + At] by the current testing-effort Expenditures are 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 [4].
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:

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = \emptyset(t)\big(n(t) - m(t)\big),$$

where $\emptyset(t) = \beta\left[r + (1-r)\frac{m(t)}{n(t)}\right]$------------------------(2.1)

r(> 0) is the inflection rate and represents the proportion of independent errors present in the Software, m(t)be the 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 equation (1) with the initial condition that, at t=0, w(t)=0, m(t)= 0, we obtain the MVF

$$m(t) = \frac{\alpha(t\beta)^{\delta}[1-e^{-\beta\delta W(t)}]}{(1+(t\beta)^{\delta})+[(1-r)/r]e^{-\beta(1-\delta)W(t)}} \text{ ----------------------- (2.2)}$$

The failure intensity at testing time *t* testing-effort is given by

$$\lambda(t) = \frac{\alpha\delta\beta w(t)e^{-\beta(1-r\delta)W(t)}}{(1-\delta\beta)[(r(1-\delta(t\beta)^2)+e^{-\beta(1-\delta)W(t)})]} \text{------------------------- (2.3)}$$

3. Imperfect-Software Debugging Models

In general software testing effort can be defined as the amount of effort spends during the software testing. Testing-effort can be described by following curves. Plenty of curves are proposed in literature to express the testing-effort.

3.1 Exponential curve:

Cumulative testing effort can described in (0, t]:

$$W(t) = \alpha\,(1 - e^{-\beta t}) \text{ ------------------- (3.1)}$$

Current testing effort $W(t) = \alpha\beta e^{-\beta t}$

Where α is the total amount of testing expenditure and β is the consumption rate of the testing-effort

3.2 Rayleigh Curve:

Cumulative testing-effort is described in (0, t]: Rayleigh curve is used by Yamada (1989) to describe the testing effort. Rayleigh curve increases to the maximum peak and decreases gradually. The Rayleigh distribution is a Weibull one with the shape factor set to two.

Cumulative testing-effort

$$W(t) = \alpha\left(1 - e^{-\beta r^2}\right) \text{---------- (3.2)}$$

Current testing-effort $w(t) = 2\alpha\beta t e^{-\beta t^2}$ --------------------- (3.3)

β is a scale parameter represents the consumption rate of the testing effort.

3.3 Weibull Curves: Cumulative testing-effort is described in (0, t]: Weibull curve is very flexible curve to model software testing effort in (0, t] (Yamada 1986): Weibull curve is flexible curve to model the reliability of the given system. Based on its nature it can take variety of forms based on the shape parameter. When m=2 its shows the Rayleigh curve and m=1 it describes the property of exponential curve.

Cumulative testing-effort $w(t) = \alpha\left(1 - e^{-\beta t^m}\right)$ ----------- (3.4)

Current testing-effort $w(t) = \alpha\beta\ t^{m-1}\ m e^{-\beta t^m}$ ----------- (3.5)

Where m is a shape parameter and β is a scale parameter

3.4 Logistic Curve: Cumulative testing-effort is described in (0, t] logistic curve has been used as the growth curve.

It is an S shaped curve, describing the first decreasing and then increasing phenomenon. The shape of the logistic distribution is similar to normal distribution.

Cumulative testing-effort

$$W(t) = \frac{\alpha}{1+Ae^{-\beta t}} \text{ -------------------------- (3.6)}$$

Current testing-effort

$$w(t) = \frac{\alpha A\beta e^{-\beta t}}{(1+Ae^{-\beta t})^2} \text{ ------------------------ (3.7)}$$

3.5 Log-Logistic curve:

The log-logistic distribution is the probability distribution of a random variable whose logarithm has a logistic distribution. It is similar in shape to the log-normal distribution but has heavier tails.

Cumulative testing-effort

$$W(t) = \frac{\alpha}{1+\left(\frac{t}{\lambda}\right)^{-\beta}} \text{ --------- (3.8)}$$

Current testing-effort

$$w(t) = \frac{\alpha\left(\frac{t}{\lambda}\right)^{-\beta}\beta}{\left(1+\left(\frac{t}{\lambda}\right)^{-\beta}\right)^{2} t} \quad \text{------------------------- (3.9)}$$

W (t) cumulative testing-effort function and w (t) is current testing effort function in (0,t] ‘α’ is total testing effort expenditure , $\lambda > 0$ scale parameter and $\beta > 0$ shape parameter.

An NHPP model is said to have imperfect debugging assumption when a (t) is constant, i.e., no new faults are introduced during the debugging process [13]. In this section we discuss the imperfect debugging of proposed SRGM.

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, w(t) obtain the following system of differential equations

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = \emptyset(t)\big(n(t) - m(t)\big),$$

and $\frac{dn(t)}{dt} = \gamma \frac{dm(t)}{dt}$

where $\emptyset(t) = b\left[r + (1-r)\frac{m(t)}{n(t)}\right]$ -----------(3.10)

Solving the above differential equations under the boundary conditions m(0)=0 and w(0) =0, we can obtain the following MVF of Log-Logistic testing-effort under imperfect debugging

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

We also have $n(t) = \frac{\alpha\left[1+e^{-\beta(1-r\delta)W(t)}\right]}{(1-\beta\delta)+[(1-r)/r]e^{-\beta(1-r\delta)W(t)}}$ ------ (3.11)

4. DATA ANALYSIS AND EXPERIMENTS

The set of actual data is from the study by [11]. 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. Through using the methods of MLE and LSE, the estimated

Time of observation (in week)	Current execution time (in CPU hr)	Cumulative execution time
1	2.45	15
2	4.90	44
3	6.86	66
4	7.84	103
5	9.52	105
6	12.89	110
7	17.10	146
8	20.47	175
9	21.43	179
10	23.35	206
11	26.23	233
12	27.67	255
13	30.93	276
14	34.77	298
15	38.61	304
16	40.91	311
17	42.67	320
18	44.66	325
19	47.65	328

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 α, β and δ of the Log-logistic TEF with imperfect debugging take δ=1 are:

α = 1201.7846 β=0.0047 δ=1

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 α, β, and δ in are:

α = 256.88 β=0.01017 δ=0.27

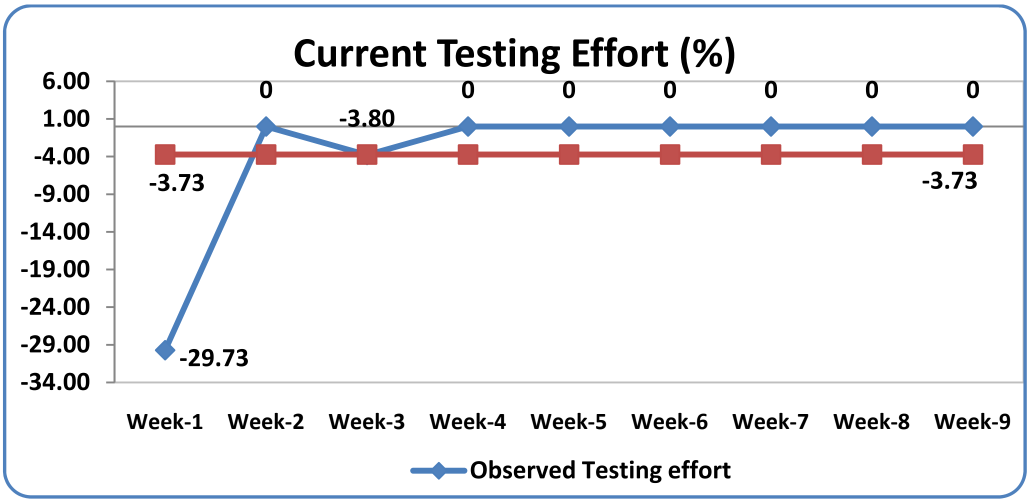

FIGURE 1: Observed/estimated current testing-effort function vs. time

Figure-2 illustrates a fitted curve of the estimated cumulative failure curve with the actual software data. The R2 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.

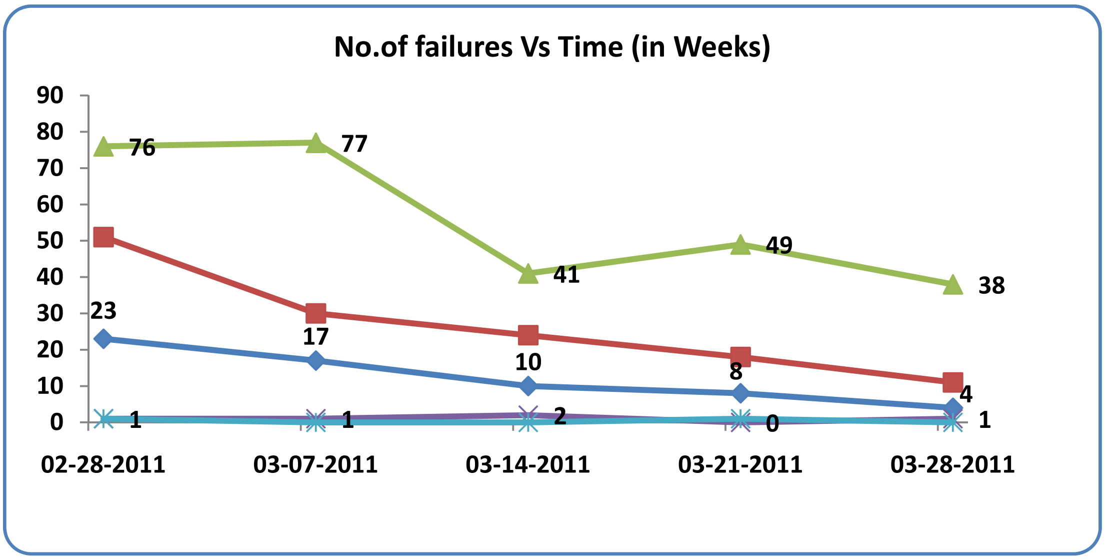

Figure 1: Observed/estimated cumulative number of failures vs. time

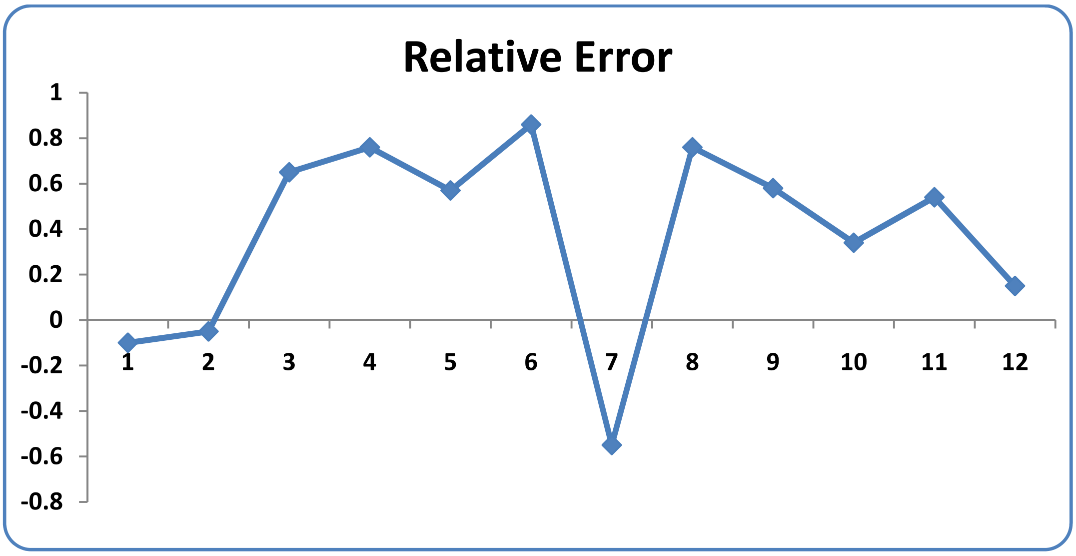

Figure 2: Predictive Relative Error Curve

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 under Imperfect Debugging

Model	α	δ	β	AE%	MSE
Proposed Model with Imperfect debugging	256.88	0.27	0.01017	6.23	67.87
Huang Logistic model	289.54		0.0252	9.68	135.87
Yamada Rayleigh model	363.18		0.0181	38.09	116.76
Yamada Weibull model	789.12		0.0925	47.67	120.34
Yamada delayed Sshaped model	276.56		0.0812	78.79	187.67
Delayed S-Shaped withLogistic TEF	365.91		0.0102	6.08	134.54
Inflection S-shaped model	467.34	1.7	0.01957	7.59	167.76
G-O Model	685.85		0.0858	18.07	112.83

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.

We observed that the value of MSE of the proposed SRGM with Log-Logistic TEF 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 TEF under imperfect debugging is illustrated by Figure 1, 3 and 2.

5. CONCLUSION

This paper, we have proposed a flexible SRGM based on NHPP model, which incorporates Log-Logistic testing-effort function with imperfect debugging (γ=1). 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 and compared with other imperfect debugging models.

6. REFERENCES

[1] N. Ahmad, M. U. Bokhari, S. M. K. Quadri, and M. G. M. Khan, "The Exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis",
[2] International Journal of Quality and Reliability Management, Vol. 25 (2), pp. 211-235 (2008).
[3] N. Ahmad, M. G. M. Khan, S. M. K. Quadri and M. Kumar, "Modeling and analysis of software reliability with Burr type X testing-effort and release-time determination", Journal of Modelling in Management, Vol. 4
[4] (1), pp. 28-54 (2009).
[5] N. Ahmad, M. G. M. Khan and L. S. Rafi, "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 (2010).
[6] M. U. Bokhari and N. Ahmad, "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
[7] (MS'2006), Montreal, Canada, 2006, pp. 540-545.
[8] S. S. Gokhale and K. S. Trivedi, "Log-logistic software reliability growth model", In: Proceedings of the 3rd IEEE International High Assurance Systems Engineering Symposium (HASE' 98), Washington, USA, 1998, pp. 34-41.
[9] C. Y. Huang and S. Y. Kuo, "Analysis of incorporating logistic testing-effort function into software reliability modeling", IEEE Transactions on Reliability, Vol. 51, no. 3, pp. 261-270 (2002).
[10] C. Y. Huang, S. Y. Kuo and M. R. Lyu, "An assessment of testing-effort dependent software reliability growth models", IEEE Transactions on Reliability, Vol. 56, no.2, pp. 198-211 (2007).
[11] S. Y. Kuo, C. Y. Hung and M. R. Lyu, "Framework for modeling software reliability, using various testing efforts and fault detection rates", IEEE Transactions on Reliability, Vol. 50, no.3, pp 310-320 (2001).
[12] M. R. Lyu, Handbook of Software Reliability Engineering, McGraw- Hill, 1996.
[13] J. D. Musa, A. Iannino and K. Okumoto, Software Reliability: Measurement, Prediction and Application, McGraw-Hill, 1987.
[14] M. Ohba, "Software reliability analysis models" IBM Journal. Research Development, Vol. 28, no. 4, pp. 428- 443 (1984).
[15] M. Ohba, "Inflection S-shaped software reliability growth models", Stochastic Models in Reliability Theory
[16] (Osaki, S. and Hatoyama, Y. Editors), 1984a, pp. 144-162, Springer-Verlag Merlin.
[17] H. Pham, L. Nordmann and X. Zhang, "A general imperfect software debugging model with S-shaped fault detection rate", IEEE Transactions on Reliability, Vol. R-48, no. 2, pp. 169-175 (1999).
[18] S. Yamada, J. Hishitani and S. Osaki, "Software reliability growth model with Weibull testing-effort: a model and application", IEEE Transactions on Reliability, Vol. R-42, pp. 100-105 (1993).
[19] S. Yamada, H. Ohtera and H. Norihisa, "Software reliability growth model with testing-effort", IEEE Transactions on Reliability, Vol. R-35, no. 1, pp.19-23 (1986).