International Journal of Computer Applications (0975 – 8887)
Volume 113 – No. 11, March 2015

Testing Effort Dependent Software Reliability Growth Model with Dynamic Faults for Debugging Process

Mohammad Altaf Dar
School of Studies in Computer Science and Applications, Jiwaji University
Gwalior (M. P.)

D N Gowsami
School of Studies in Computer Science and Applications, Jiwaji University
Gwalior (M. P.)

Anshu Chaturvedi
Madhav Institute of Technology and Science, Gwalior (M. P.)

ABSTRACT
In present era people depend on both hardware and software system. As software system is engrafted in every aspect of computer system, the desired quality of software is an essential concern for many critical system. From last few decades, many software reliability growth models were developed to analyze the growth of reliability. For improving the quality of software, SRGM plays an essential role. The present study proposed a Software Reliability Growth Model with testing effort and dynamic fault. The parameters involved in the proposed model are estimated using least square estimation. The performance of the proposed model is validated using Mean Square Error (MSE), Akaike Information Criterion (AIC) and R Squared Error (R^2). A proposed Model is compared with existing models reported in literature, and it has been observed that proposed model performed better.

Keywords
Software Reliability, Software Reliability Growth Models, Test effort, Fault

1. INTRODUCTION
Software became critical part of our society, plays a key role in controlling resources in different areas like banks, telecommunications, nuclear plants, mills as well as defense system. Even in households many of the appliances, automobiles are controlled by software. Companies, industries, educational institutions as well as business hubs had become extraordinary dependent on software. With the dawn of the computer age, computers as well as software running on them are playing a catalytic role in human life. However computer, computer-based appliances have invaded every area of human activity. As more and more organizations are being computerized, human dependency on computer is increasing vigorously, i.e., the human life became software dependent. As this revolution of technology made our lives easier, corporation of safety and security has become incredible. There are lot of instances where the failure of computer-based appliances are responsible for the loss of both human health and wealth. The impression of software on our society continue to be enlightened. Though increasing growth, the software engineers as well as software companies continuously endeavor to acquire technologies that will render it more comfortable, faster, and less expensive to develop high quality software. From last few decades, many reliability growth models were developed to analyze the progress of reliability. As far as the quality of software is concerned, reliability models plays a vital role. The present study considered a growth model incorporating test effort and dynamic faults. A number of Software Reliability Growth Models have been constructed with or without testing effort [1-12]. Kapur et al [11] proposed Software reliability growth model with testing effort dependent learning function. Pachauri et al. [12] proposed Software reliability growth model with dynamic faults and optimal release time. Quadri et al. [13] used generalized exponential curve as testing-effort function in Software Reliability Growth Model. Huang et al. [14], described how to incorporate the logistic testing-effort function into both exponential type, and S-shaped software reliability models. Ahmad et al. [15, 16] proposed Software Reliability Growth Model with optimal release time with exponentiated Weibull and Burr-type X testing-effort functions. Rafi et al. [17, 18], Rafi and Akthar [19] proposed three Software Reliability Growth Models with generalized modified Weibull (GMW), Gompertz and logistic exponential curves as testing-effort functions, respectively, with optimal release policy.

Considering the work of Kapur et al. [11], Pachauri et al. [12] the focus is to develop a more efficient Software Reliability Growth Model in debugging environment using Generalized Modified Weibull Distribution.

2. GENERALIZED MODIFIED WEIBULL DISTRIBUTION
In this study, the Generalized Modified Weibull Distribution is used as a testing effort function. The Generalized Modified Weibull Distribution density function is mathematical expressed as in [12]

$$G(t) = \alpha(1 - e^{-\beta t^m} e^{\lambda t})^{\theta} \quad --1$$

Where m and θrepresents the shape parameter, α, β and λ represents total effort expenditure, scale parameter and accelerating factor respectively. Therefore the Cumulative Testing effort function is:

$$g(t) = \frac{dG(t)}{dt} = \frac{\alpha\beta\theta e^{\lambda t}\left[e^{-\beta t^m e^{\lambda t}}\left(e^{\beta t^m e^{\lambda t}}-1\right)(mt^m+\lambda t^m)\right]}{e^{\beta t^m}e^{\lambda t}-1} \quad --2$$

The above distribution has greater flexibility with all forms of hazard rate function. However, it can be used with number of models. One of the importance of this distribution is that it is a generalization of many existing Software Reliability Growth Models. Few of them are enlisted in below given table:

Table 1: Generalized Modified Weibull Distribution

Distribution	Parameters	Expression
Yamada Weibull curve	λ=0, θ=1	$G(t) = \alpha(1 - e^{-\beta t^m})$
Modified Weibull distribution [22]	θ=1	$G(t) = \alpha(1 - e^{-\beta t^m} e^{\lambda t})$
Exponentiated Weibull curve [15]	λ=0	$G(t) = \alpha(1 - e^{-\beta t^m})^{\theta}$
Generalized exponential curve [13]	λ=0, m=1	$G(t) = \alpha(1 - e^{-\beta t})^{\theta}$

Yamada exponential curve [10]	λ=0, θ=1, m=1	$G(t) = \alpha(1 - e^{-\beta t})$
Yamada Rayleigh curve [10]	λ=0, θ=1, m=2	$G(t) = \alpha(1 - e^{-\beta t^2})$
Burr type X curve [16]	λ=0, m=2	$G(t) = \alpha(1 - e^{-\beta t^2})^{\theta}$
Extreme value distribution (log–gamma)	θ=1, m=0	$G(t) = \alpha(1 - e^{-\beta e^{\lambda t}})$

3. MODEL DEMONSTRATION

The proposed model is motivated by the work of Kapur et al [11] and Pachauri et al. [12]. In this section, a Software Reliability Growth Model based on Non Homogenous Poisson Process (NHPP) is developed by incorporating two things, one is the testing effort and second one is dynamic fault.

3.1. Notations used

m(t): Mean value function

G(t): Testing effort function

a(t): Total number of faults

b: Detection rate

μ: New faults introduction rate

3.2. Assumptions

Following assumptions used in proposed model are given in [11, 12]:

1. Total number of faults detected/removed follows Poisson distribution.

2. Failure rate of software is equally affected by remaining faults in software.

3. The fault removal with respect to test effort is proportional to mean number of remaining faults in the software.

4. Generalized Modified Weibull Distribution is used to model the Test Effort Function (TEF).

In this study, it is assumed that when a failure occurring in a software is removed, new faults can be introduced during debugging process with a probability of μ. Based on the assumptions, the following differential equation for the mean value function m(t) are :

$$\frac{d}{dt}m(t) \times \frac{1}{G(t)} = b(t)[a(t) - m(t)] \qquad \text{--3}$$

$$\frac{d}{dt}a(t) = \mu \frac{d}{dt}m(t) \qquad \text{--4}$$

$$b(t) = \frac{b^2 t}{1+bt} \qquad \text{--5}$$

Solving the differential equations (3) and (4), the expected number of faults at time t denoted by mean value function m(t) is obtained as:

$$m(t) = \frac{a}{1-\mu}\left[1 - \{1 + b(1-\mu)G(t)\}e^{-b(1-\mu)G(t)}\right] \qquad \text{--6}$$

4. APPLICATIONS

In this section, applicability of the proposed model is shown by validating it on software failure data set obtained from different real software development projects. The datasets derived from different time-periods are illustrative of industrial software processes prevalent in that period. The procedure is as follows:

First, we fit the proposed model into the data i.e., parameter estimation, and obtain mean value function m(t). Secondly, the proposed model is compared with the existing models available in literature within a dataset using the MSE, AIC and R^2.

5. PARAMETER ESTIMATION

To support the model applicability both the parameter estimation and model validation are the necessary aspects. The mathematical equation of the proposed SRGM is non-linear. Nevertheless, it is hard to discover the answer for a nonlinear model using Least Square Method and requires numerical algorithm to resolve it. To overcome this problem. Statistical Software Package such as SPSS is used. SPSS is a statistical package for social sciences. To estimate the parameters of the proposed model, a Least Square method (Non-linear regression method) is used. Non-linear regression method finds the relationship between the dependent and independent variable. Non-linear regression can estimate models with arbitrary relationships between autonomous and dependent variable.

5.1. Goodness of Fit

The term goodness of fit is used in two different contexts, in one context it denotes the question if sample of information comes from a population with a specific distribution. In another context it denotes the question of "How good does mathematical model fit to the data."

5.1.1. Mean Square Fitting Error

The model under the comparison is used to simulate the fault data, the difference between the expected values $m(t_i)$, and the observed data Y_i is measured by MSE as follows:

$$\text{MSE} = \sum_{i=1}^{k} \frac{(m(t_i) - y_i)^2}{k} \qquad \text{--7}$$

Where "k" is the number of observations. The lower the MSE indicates a less fitting error, thus better goodness of fit

5.1.2. Akaike Information Criterion (AIC)

It is defined as AIC = -2(The value of the maximum log likelihood function) + 2(The number of the parameters used in the model).This index takes into account both the statistical goodness of fit and the number of parameters that are estimated. Lower values of AIC indicate the preferred model

5.1.3. R Square (R^2)

The model under this comparison is used to examine whether a significant trend exists in the observed failure intensity. It is defined as the ratio of Sum of Squares resulting from the constant model subtract from 1, that is

$$R^2 = 1 - \frac{\text{residual sum of square}}{\text{corrected sum of square}} \qquad \text{--8}$$

Its value ranges from 0 to 1. The low value indicates that the model does fit easily to the data set.

5.2. Model Validation

To validate as well to determine the software reliability growth of a proposed model, it has been tried out on two testing datasets, which are documented in [20] [21] respectively. The proposed model has been compared with the NHPP of Kapur et al [11]. The results are shown in tables below. After observing the goodness-of fit, the values of MSE, AIC and R^2 are smaller than the values of other models. Overall, the proposed model perform better. The plot of observed values and estimated values of the datasets are illustrated in figures 1-4.

Table 1: Goodness-of fit criteria based on dataset documented in [20]

Model	MSE	AIC	R^2
Kapur	3260.677	429.1246	0.982
Proposed	2967.211	371.2412	0.957

Table 2: Goodness-of fit criteria based on dataset documented in [21]

Model	MSE	AIC	R^2
Kapur	663.3415	215.717	0.995
Proposed	598.1002	198.311	0.966

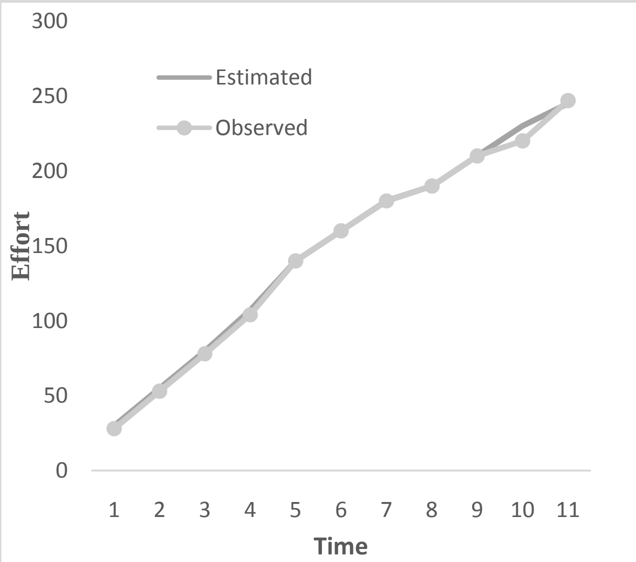

Figure 1: Test effort between observed and estimated curve on Dataset –I

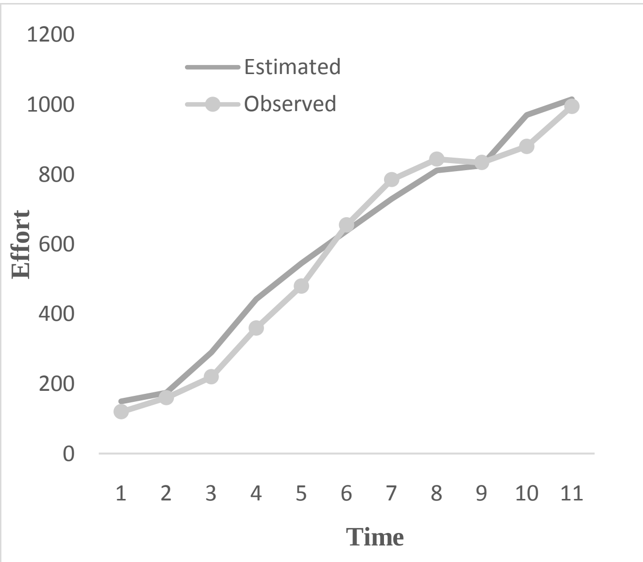

Figure 2: Goodness-of fit between observed and estimated curve on Dataset -I

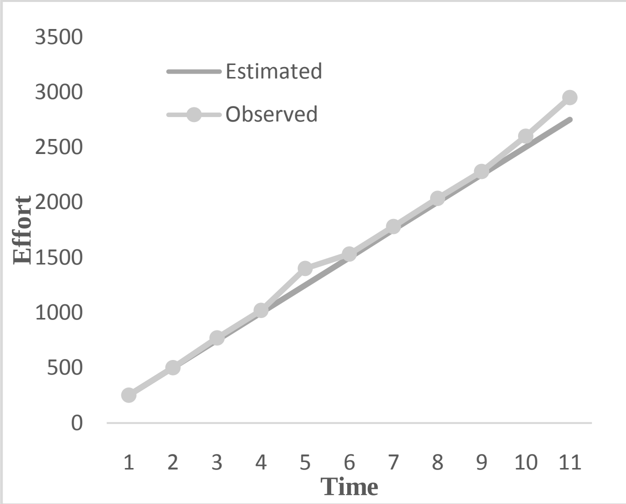

Figure 3: Test effort between observed and estimated curve on Dataset -II

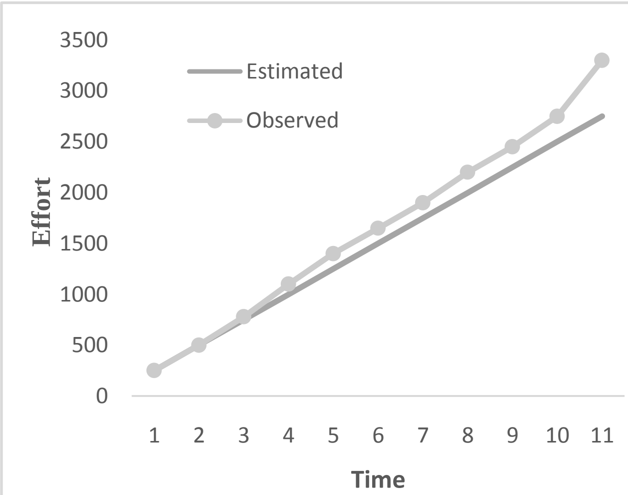

Figure 4: Goodness-of fit between observed and estimated curve on Dataset -II

6. CONCLUSION

In this piece of research work, a new Software Reliability Growth Model is constructed based on NHPP. This model incorporates more novel features namely GMW testing effort function and dynamic faults and is suitable for describing the fault detection/removal process during debugging. The proposed Software Reliability Growth Model was validated on the failure datasets available in literature, which demonstrates the applicability of the proposed model. From the comparative study it has been concluded that the proposed model fits the data fairly well.

7. REFERENCES

[1] Goel and Okumoto, 1979 "Time dependent error-detection rate model for software reliability and other performance measures," IEEE Trans. Reliab. pp. 206–211

[2] Kimura, Yamada and Osaki, 1992 "Software reliability assessment for an exponential S-shaped reliability growth phenomenon," Comput. Math. Appl., pp. 71–78

[3] Huang, Kuo and Lyu, 1999, Optimal software release policy based on cost and reliability with testing efficiency, 23rd International Conference on Computer Software and Applications, pp. 468–473

[4] Huang and Kuo, 2002, Analysis of incorporating logistic testing-effort function into software reliability modelling, IEEE Trans. Reliab. pp. 261-270

[5] Popstojanova and Trivedi, 2003, Architecture-based approaches to software reliability prediction, Comput. Math. Appl. pp.1023–1036

[6] Lo, Huang, Chen, Kuo and Lyu, 2005, Reliability assessment and sensitivity analysis of software reliability growth modeling based on software module structure," J. Syst. Softw. pp. 3–13

[7] Huang, 2005, Performance analysis of software reliability growth models with testing-effort and change-point, J. Syst. Softw. pp.181–194

[8] Huang, 2005, Cost-reliability-optimal release policy for software reliability models incorporating improvements in testing efficiency," J. Syst. Softw. pp 139–155

[9] Yamada and Osaki, 1985, Software reliability growth modeling: models and applications, IEEE Trans. Softw. Eng. pp. 1431–1437

[10] Yamada, Ohtera and Norihisa, 1986, Software reliability growth model with testing-effort, IEEE Trans. Reliab.pp.19–23

[11] Kapur, Goswami, and Gupta, 2004, A Software reliability growth model with testing effort dependent learning function for distributed systems," Int. J. of Reliab., Qual., Safety Eng.. pp. 365-377

[12] Pachauri, Kumar and Dhar, 2014, Software reliability growth model with dynamic faults and optimal release time optimization using GA and MAUNT," Applied Math., and comput., pp 500-509

[13] Quadri, Ahmad, Peer and Kumar, 2006, Non-homogeneous Poisson process software reliability growth model with generalized exponential testing effort function, RAU J. Res. pp.159–163

[14] Huang, Kuo and Lyu, 2007, An assessment of testing-effort dependent software reliability growth models, IEEE Trans. Reliab. pp. 198–211

[15] Ahmad, Bokhari, Quadri, and Khan, 2008, The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy, Int. J. Qual. Reliab. Manag. pp. 211–235

[16] Ahmad, Khan, Quadri and Kumar, 2009, Modeling and analysis of software reliability with Burr type-X testing-effort and release-time determination," J. Model. Manag. pp. 28–54

[17] Rafi, Rao and Akhtar, 2010, Incorporating generalized modified Weibull TEF in to software reliability growth model and analysis of optimal release policy," Comput. Inf. Sci. pp.145–162

[18] Rafi, Rao, and Akhtar, 2010, Software reliability growth model with logistic exponential test-effort function and analysis of software release policy," Int. J. Comput. Sci. Eng. pp. 387–399

[19] Rafi and Akthar, 2010, Software reliability growth model with Gompertz TEF and optimal release time determination by improving the test efficiency, Int. J. Comput. Appl. pp. 34–43

[20] Brooks and Motley, 1980, Analysis of discrete software reliability models, RADC-TR, 80-84

[21] Misra,1983, Software reliability analysis," IBM Sys. J., pp. 262-270

[22] Xie and Murthy, 2003, A modified Weibull distribution," IEEE Trans. Reliab. pp. 33–37