Volume 5, Issue 3, March 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Software Reliability Growth Model Involving Non Linear Time Dependent Fault Removal Efficiency

Suneet Saxena
Department of Mathematics,
SRMS College of Engineering and Technology, Bareilly, India

Abstract— The paper presents variable fault removal efficiency (FRE) in NHPP Software Reliability Growth model. Various models have been developed with constant FRE. However, during testing phase it has been found that FRE depends on testing time as detection and removal of fault consume considerable length of time. Proposed model considered FRE as function of time and incorporates exponential order of time component. Parameters are estimated by MLM and goodness of fit is performed. The model compared with Delayed S-shaped model using statistical tools SSE , R^2 and AIC. Results show the model fits and predict faults better.

Keywords— Non Homogeneous Poisson Process, Maximum likelihood method(MLM), Akaike's Information Criterion(AIC), Coefficient of Determination (R^2), Software Reliability.

I. INTRODUCTION

In present world human beings are extremely depends on machine controlled by software for day to day tasks. Banking transactions, shopping, railway, aviation, car and hotel reservation can be carried out online through internet. High speed bullet train and aero plane running on autopilot mode and various advance surgeries are now being controlled by complex software system. Failure of software would not only result in financial loss but also loss of human lives. Software failures are costing the U.S. economy an estimated $59.5 billion each year, with more than half of the cost borne by end users and the remainder by developers and vendors. In Loss Angles air traffic controllers lost voice contact with 400 airplanes they were tracking over the south western United States and risking lives of large number of people.

The primary concern of software developers is to develop highly reliable software in order to minimize failures as far as possible. Software reliability is the probability of failure free software operation for a specified period of time in specified environment. Many mathematical models have been proposed for the estimation of software reliability and are referred as software reliability growth models(SRGM). These models establish mathematical relationship between cumulative faults detected and testing time. Models developed assuming variable failure rate are considered as non homogenous Poisson process (NHPP) models whereas homogenous Poisson process (HPP) models involve constant failure rate. NHPP SRGMs [1-8] are used extensively for reliability estimation of software.

Software would be more reliable if maximum number of faults detected and corrected at testing phase. Fault removal process is very complex process and time consuming. It involves detection and correction of faults. Fault removal efficiency is defined as the probability of perfectly removing a fault in first repair attempt[1]. Previously various authors developed SRGMs assuming value of Fault removal efficiency is one, means fault detected and completely corrected and no occurrence of that fault in future. However, Experience shows that this is not the case. Software fault occur multiple times before it is finally removed. Fault removal efficiency is function of testing time as detection and removal of fault consume considerable length of time. Model [8] suggested time to remove fault is constant and therefore FER not depends on time. NHPP Models [4] incorporated fault removal efficiency as function of time during development of model. Later on FER considered as constant in order to find the solution of model. Model [7] considered time dependent fault removal efficiency involving time component of degree one.

The paper presents NHPP SRGMs with FER as function of time and incorporates exponential order of time component. Section 2 proposes development of model and solution under certain assumptions. Section 3 discussed the parameter estimation and comparison. Finally conclusions highlighted in section 4.

A. *Notations*

a Total number of faults in the software.

b rate of fault detection.

m(t) Expected number of faults detected by time t.

$x(t)$ Expected number of faults removed by time t.

$p(t)$ Fault removal efficiency.

μ Expected time to remove a fault.

B. Assumptions

- The occurrence of faults follows Non Homogeneous Poisson Process.
- Fault detection rate (b) considered as constant.
- Fault removal efficiency is function of testing time.
- No introduction of new faults during debugging process.
- Total number of faults in the software is constant.

II. MODEL DEVELOPMENT

The rate of fault detected at time t is proportional to fault detection rate (b) and remaining faults $(a - x(t))$ in software at time t. $x(t)$ is expected number of faults removed by time t. Fault removal rate is proportional to fault removal efficiency (FRE) and fault detection rate. Model represented by following two differential equations.

$$\frac{dm}{dt} = b(a - x(t)) \tag{1}$$

$$\frac{dx}{dt} = p\frac{dm}{dt} \tag{2}$$

FRE (p) addresses the problem of multiple occurrences of fault before its final removal. In practice it has been observed that considerable length of time consumes in locating the faults that are detected later. FRE should decrease with testing time as more time is consumed to remove them. Model [1] supposed μ as expected time to remove a fault. The term μe^{bt} represents the expected number occurrences of fault before its final removal. Therefore, present model proposes FRE as function of time given by

$$p(t) = \frac{1}{1+\mu e^{bt}} \tag{3}$$

In view of above discussion equation (2) modified as

$$\frac{dx}{dt} = p(t)\frac{dm}{dt} \tag{4}$$

Using marginal conditions $m(0) = 0$, $x(0) = 0$ for differential equations (1) and (4), the solutions are:

$$x(t) = a\left[1 - \exp\left(-b\int_0^t p(u)du\right)\right] \tag{5}$$

$$m(t) = ab\int_0^t \exp\left(-b\int_0^v p(u)du\right)dv \tag{6}$$

Substituting (3) in (6), solution is :

$$m(t) = ab\int_0^t \exp\left(-b\int_0^v \frac{1}{1+\mu e^{bu}}\, du\right)dv \tag{7}$$

$$m(t) = \frac{a}{1+\mu}[1 + b\mu t - \exp(-bt)] \tag{8}$$

III. PARAMETER ESTIMATION AND COMPARISON

A. Parameter Estimation

Parameters of model are estimated by Maximum likelihood method using Musa's [2] SS1a failure dataset .The likelihood function for unknown parameters a, b and μ is given by

$$L(a,b,\mu) = \prod_{i=0}^{n} \frac{[m(t_i) - m(t_{i-1})]^{(y_i - y_{i-1})}}{(y_i - y_{i-1})!} \exp[-(m(t_i) - m(t_{i-1}))]$$

There are n observed data pairs (t_i, y_i) where y_i is observed cumulative faults at time t_i.
The parameters are estimated by maximizing likelihood function $L(a,b,\mu)$.

Table 1: Estimated Parameters.

	a	b	μ
MLE	70	0.0135	25

B. Goodness of Fit (GoF)

Measures of Gof are used to compare the models. Propose model is compared with Delayed S-shaped model .Some of the measures used to determine GoF are given below:

i) *Sum of Square Error (SSE)* : SSE is the sum of squares of residuals between observed value and estimated value. It can be expressed as

$$\mathrm{SSE} = \sum_{j=1}^{n}(y_j - E_J)^2$$

where y_j is observed cumulative faults at time j and E_J estimated cumulative faults at time j. Model with lower SSE fits better to given dataset.

ii) *Akaike's Information Criterion(AIC)* : AIC is used to compare the models. It can be evaluated as

AIC=-2*log(likelihood function at its maximum value)+2*N.

Where N represents the parameters in the model. The model with minimum AIC value is chosen as the best model to fit the data. In AIC, the compromise takes place between the maximized log likelihood and the number of free parameters estimated within the model (the penalty component) which is a measure of complexity or the compensation for the bias in the lack of fit when the maximum likelihood estimators are used.

iii) *Coefficient of Determination* (R^2) : Coefficient of Determination is also known as multiple correlation coefficient. It measures the correlation between the dependent and independent variables. Value of R^2 vary from 0 to 1. For $R^2 = 1$, fitting is perfect. $R^2 = 0$, no fitting and R^2 close to 1, good fitting.

R^2 is defined as : $R^2 = \frac{\sum_{j=1}^{n}(E_J - \bar{y})^2}{\sum_{j=1}^{n}(y_j - \bar{y})^2}$ $\bar{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$

where y_j is observed cumulative faults at time j and E_J estimated cumulative faults at time j. n is number of data points. Model fits better to given dataset if R^2 close to 1.

Table 2: Goodness of Fit.

	SSE	R^2	AIC
Delayed S-shaped	3342.31	1.36	133.46
Proposed Model	1274.72	0.96	121.59

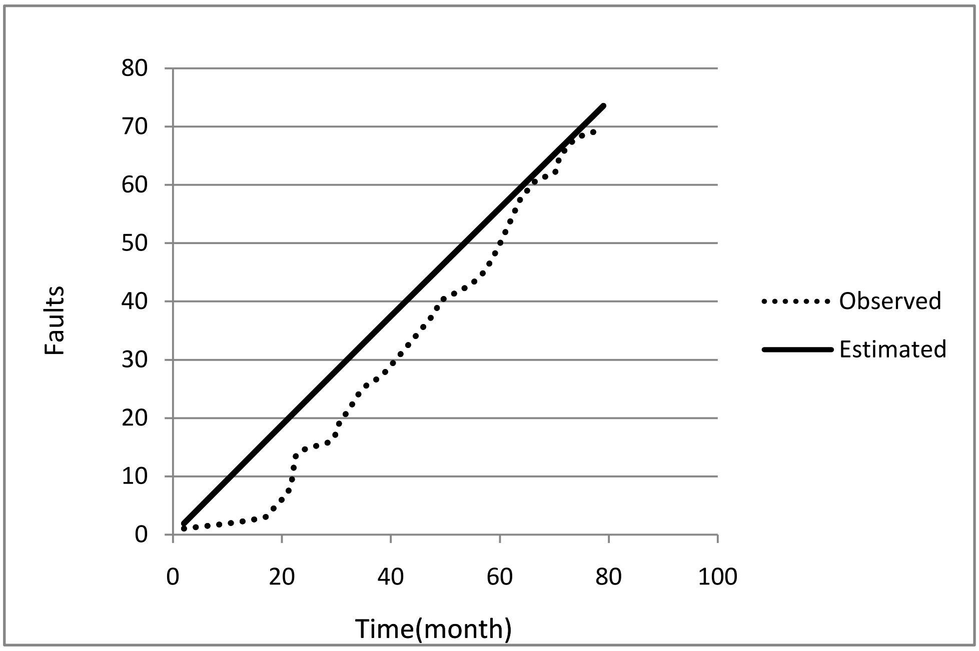

Figure 1: Observed and estimated faults for proposed model.

IV. CONCLUSION

The model presented in this manuscript considered fault removal efficiency (FRE) as function of time and incorporates exponential order of time component. Time dependent form of FRE addresses two problems effectively. One is multiple occurrences of fault before its final removal and other is time consumption during removal process. Model compared with delayed S-shaped model . The graph between faults detected and testing time indicates as testing time increases expected faults coincides with observed faults (after 60 months of testing). Proposed model effectively endorsed observed data set. Values of Statistical tools SSE, R^2 and AIC also indicates proposed model fits better for given dataset.

REFERENCES

[1] D. R. Jeske, X. M. Zhang, and L. Pham, “*Accounting for realities when estimating the field failure rate of software*”, in Proc. Of 2001 Int. Symp. Soft. Rel. Eng. (ISSRE’01), pp 332-339, Hong Kong, China, Nov. 2001.

[2] D.Musa, “*DACS software reliability dataset”*, Data & Analysis Center for Software, J.D. 980. www.dacs.dtic.mil/databases/sled/swrel.shtml. 2005.

[3] H. Pham, L. Nordmann, and X. Zhang, “*A general imperfect-software debugging model with S-shaped fault-detection rate”*, IEEE Transactions on Reliability, Vol.48, No.2, pp. 169-175. 1999.

[4] B. Purnaiah, K. V. Rama, and V. K. G. Bala, “*Fault removal efficiency in software reliability growth model”*, Advances in Computational Research, Vol. 4, Issue 1, pp . 74-77, 2012.

[5] R. Subburaj, G. Gopal, and P. K. Kapur, "*A software reliability growth model for vital quality metrics*", South African Journal of Industrial Engineering, Vol 18(2), pp 93-108, 2007.

[6] R. Subburaj, and G. Gopal, "*Generalized exponential poisson model for software reliability growth*", International Journal of Performability Engineering, Vol.2, No.3, pp. 291-301, 2006.

[7] X. Yang, N. Sang, and H. Lei, "*An improved NHPP model with time-varying fault removal delay"*, Journal of Electronic Science and Technology of China, Vol 6, No. 3, September 2008.

[8] X .M. Zhang, X. L. Teng, and H. Pham, "*Considering fault removal efficiency in software reliability assessment*", IEEE Trans. Systems, Man and Cybernetics-Part A: System and Humans,Vol. 33, no. 1, pp. 114-120, jan 2003.