IJASRM

International Journal of Advanced Scientific Research and Management, Volume 3 Issue 9, Sept 2018

www.ijasrm.com

ISSN 2455-6378

Framework for modelling software reliability growth for error detection with dynamic faults

Mohammad Altaf Dar[1], Showkat Ahmad Teeli[2], Fayaz Ahmad Bhat[3]

[1]School of Studies in Computer Science & Applications, Jiwaji University, Gwalior (M. P.)
[2,3]Govt. Degree College Dooru Anantnag (J&K)

Abstract

Software reliability assessment is important to evaluate and predict the reliability and performance of the software system. The models applicable to the assessment of the software reliability are called Software Reliability Growth Model. In the present study a new Software Reliability Growth Model is developed by integrating the dynamic fault with Delayed S-shaped NHPP growth model to get a better description of the software fault detection phenomenon. The parameters involved in the proposed model are estimated using least square estimation and the validation techniques are Sum of Squared Errors (SSE) and Akaike Information Criterion (AIC).From the comparative study, it has been conclude that the proposed model is able to fit a variety of reliability growth curves and the performance of the proposed Software Reliability Growth Model with dynamic fault was better than other existing models

Keywords

Software Reliability, Software Reliability Growth Models, Dynamic Fault, Error Detection.

1. Introduction

Software Reliability Growth Models based on NHPP have been quite successful tool in practical software reliability engineering. These models consider the debugging process as a counting process characterized by its mean value function. Software reliability can be estimated once the mean value function is obtained.

To consider the proposed problem, lot of Software Reliability Models are available in literature for example; Schneidewind (Schneidewind, 1975) proposed an error detection model. Goel and Okumoto (Goel, et al, 1979) proposed the time dependent failure rate model. Ohba (Ohba, 1975) proposed the inflection S-shaped model. Musa and Okumoto (Musa et al, 1984) (Yamada et al, 1983) proposed the basic execution time model and Log Poisson model respectively. Yamada, Ohba and Osaki (Yamada et al, 1983) proposed a model based on the concept of failure observation and the corresponding fault removal phenomenon. Goel (Goel, et al, 1985) modified his original model by introducing the test quality parameters. Littlewood (Littlewood, et al, 1981) proposed a modification of Dunae model based on NHPP. Yamada, Osaki and Narthisa (Yamada et al, 1985) proposed a model with two types of faults. Yamada and Osaki (Yamada et al, 1986) also proposed two classes of discrete time models. One class describes an error detection process in which the expected number of errors detected per test case is geometrically decreasing while the other class is proportional to the current error content. Yamada, Ohtera and Narihisa (Yamada et al, 1986) further proposed a testing-effort dependent model which assumes the testing-effort to follow either exponential, Weibull or Rayleigh distribution. Kapur, Garg and Kumar (Kapur et al, 1999) proposed a discrete time model based on the concept that the testing phase has two different processes namely, fault isolation and fault removal. Kapur and Younes (Kapur et al, 1995) further proposed a discrete time model based on the assumption that the software consists of different types of faults and on each type of fault a different strategy is required to remove the cause of the failure due to that fault. Ohba (Ohba et al, 1984) proposed the hyper-exponential model to describe the fault detection process in a module structured software. Khoshgoftaar (Khoshgoftaar, 1988) proposed the K-Stage Erlangian model. Kapur and Garg (Kapur et al, 1999) modified Goel-Okumoto model by introducing the concept of imperfect debugging. Kareer, Grover and Kapur (Kareer et al, 1990) proposed two types of faults models where each fault type is modelled by an S-shaped curve. Ahmad (Ahmad et al, 2010) proposed an inflection S-shaped Software Reliability Growth Model that incorporates exponentiated Weibull test effort function in both perfect and imperfect debugging environments. Rafi (Rafi et al, 2010) discussed module testing and resource allocation dependent release policy in imperfect debugging environment. Ahmada (Ahmada et al, 2011) proposed inflection S-shaped Software reliability Growth Model considering the log-logistic curve as testing effort function. Pachauri(Pachauri et al, 2014) proposed Software reliability Growth Model that incorporates generalized modified Weibull testing effort function in imperfect debugging environment with constant and time varying fault detection rates. Madhu (Madhu et al, 2014) proposed Software reliability Growth Model with imperfect debuging, Change points and a fault reduction factor.

The present paper is divided in different sections; each section describes its own purposes. The section 1 gives the brief introduction and literature related to the proposed work. Section 2 gives description about the model development, notations, assumptions and mean value function of the proposed model. In section 3 numerical results are obtained by using the goodness of fit criteria, such as SSE and AIC. Section 4 concludes the presented work. And the last section presents the citations that are used during the study.

1.1 Model Development

In the present study a new Software Reliability Growth Model is developed based on NHPP allowing both error introduction and error detection rate as a time dependent function. NHPP based Software Reliability Growth Model and an expression for mean value function is presented.

In the present study, the concept of dynamic fault is integrate with Delayed S-shaped NHPP growth models (Yamada et al, 1983) to get a better description of the software fault detection phenomenon. The result shows that the proposed Software Reliability Growth Model with dynamic fault gives better performance than other existing models

1.1.1 Notations used during the study

m(t): Mean value function

a(t): Total number of faults

r: Detection rate

μ: New faults introduction rate

1.1.2 Assumptions used during the study

Following assumptions used in proposed model:

Error removal process follows NHPP

Failure detection is proportional to the remaining number of faults in the software.

The initial fault content in the software are dynamic.

Failures at random, caused by the actual number of faults in the software.

When error caused in the software is removed, it is possible to introduce new error.

Based on the assumptions, we have the following differential equation for the mean value function m(t)

$$\frac{d}{dt}m(t) = r(t)[a(t) - m(t)] \quad (1)$$

$$\frac{d}{dt}a(t) = \mu\frac{d}{dt}m(t) \quad (2)$$

$$r(t) = \frac{r^2 t}{1+rt} \quad (3)$$

Solving the differential equations with initial conditions m(0) and a(0), we get the expected number of faults at time t denoted by mean value function m(t):

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

Where μ is the probability of introducing new fault in the software during testing. The mean value function m(t) is related to the other dependent functions a(t) and b(t) that by their definition, possesses actual meaning. By introducing the functional assumptions about a(t) and b(t), an expression for m(t) can be derived given in equation (4). The mean value function m(t) represents the expected number of software errors that have accumulated up to time t. In next step, this mean value function m(t) must be defined analytically, which is done usually by expressing the mean value expression as a function of other functions: the initial error content a(t) and the error detection rate r(t). By making assumptions about analytical behavior of these functions a(t) and r(t) are defined as function of time with one or more parameters. In the third step, actual failure dataset are analyzed, using statistical model or tool.

2 Numerical results

In this section numerical results are obtained using statistical tool such as SPSS to examine validity of the proposed model. The numerical results are presented to visualize the effects of different parameters. The parameter estimation are calculated using the dataset documented in (Kapur et al, 1995). The parameter estimation results and comparison criteria are listed in below table 1. After observing the goodness of fit the values of SSE and AIC are smaller in comparison to other existing models. Overall the performance of the proposed model is comparatively good. The figures 1 – 2 gives the pictorial representation of observed and estimated values.

2.1 Sum of Squared Errors (SSE)

The SSE is a mathematical approach to determining the scattering of data points; found by squaring the length between each data point and the line of best fit and then summing all of the squares. The sum of the squared errors (SSE) is defined as follows:

$$SSE = \left(\sum_{i=1}^{N}(Y_i - \hat{Y}_i)\right)^2 \qquad (5)$$

Where: Yi is the actual observations time series, $\hat{Y}_i$ is the estimated or forecasted time series

2.2 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.

Table 1: Comparison of Goodness of Fit using Dataset 3

Model	SSE	AIC
Proposed	44.62	79.72
Kapur [Kapur et al, 1995]	52.97	97.27
S-shaped [Yamada et, al 1983]	47.276	85.76

documented in (Kapur et al, 1990)

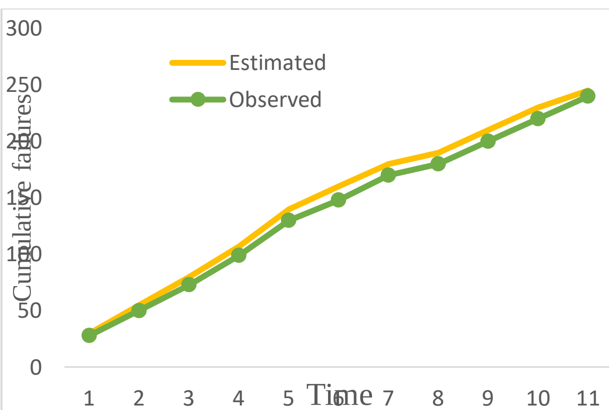

Figure 1: Goodness of fit between Observed and (Kapur) Estimated Curve

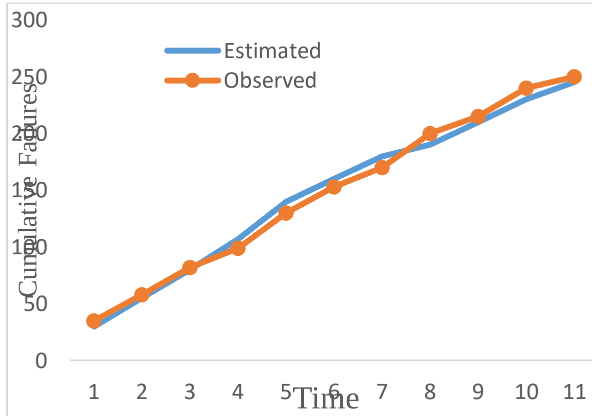

Figure 2: Goodness of fit between Observed and (S-shaped) Estimated Curve

3. Conclusion of proposed model

A new NHPP based software reliability growth model for error detection with dynamic faults is proposed during the study. In this study it has been assumed that failure detection is proportional to the remaining number of faults in the software, initial fault content in the software is dynamic and when error caused in the software is removed, it is possible to introduce new error. 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 experimental results of proposed model fits the dataset fairly well.

In future, it can be extended to incorporate the concept of fault reduction factor and change point.

References

[1] Schneidewind N. F “Analysis of Error Process In Computer Software”, Sigplan Notices, Vol. 10, pp. 337-346, 1975.

[2] Goel A , Okumoto K, “Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures”, IEEE Transactions on Reliability, Vol. 28, pp. 206–211, 1979.

[3] Ohba M, “Software Reliability Analysis Models”, IBM Journal of Research and Development,Vol. 28 pp. 428–443, 1984.

[4] Musa J, "A Theory of Software Reliability and Its Application," IEEE Transactions. on

Software Engineering, Vol. 1, pp 312-327, 1975.

[5] Musa J and Okumoto K "A Logarithmic Poisson Execution Time Model For Software Reliability Measurments," Proceeding of the 17th International Conference on Software Engineering, Orlando, Vol., pp. 230-237, 1984.

[6] Yamada S, Ohba M, Osaki S, "S-Shaped Software Reliability Growth Modeling for Software Error Detection", IEEE Transactions on Reliability, Vol. 10, pp 475-478, 1983.

[7] Goel A, "Software Reliability Models: Assumptions, Limitations, and Applicability", IEEE Transaction Software Engineering, Vol. 11, pp. 1411 – 1423, 1985.

[8] Littlewood B, "Stochastic Reliability-Growth: A Model for Fault removal in Computer Programs and Hardware designs", IEEE Transaction on Reliability, Vol.30, pp. 313-320, 1981.

[9] Yamada S, Osaki S and Narithisa N, "A software reliability growth model with two type of errors", RAIRO, Vol. 35, pp 87-104, 1985.

[10] Yamada S, Osaki S, "Software reliability growth modeling: models and applications", IEEE Transaction Software Engineering. Vol. 11, pp. 1431–1437, 1985.

[11] Yamada S, "Software Reliability Growth model with testing-effort", IEEE Transaction on Reliability, Vol. 35, pp.19–23, 1986.

[12] Kapur K, Garg R, and Kumar S, "Contributions to Hardware and Software Reliability", World Scientific, Singapore, 1999.

[13] Kapur, Younes, "Software Reliability Growth Model with Error Dependency", Microelectronics and Reliability, pp 273–278, 1995.

[14] Khoshgoftaar K, "Non-Homogenous Poisson Process for Software Reliability Growth Model", COMPSTAT'88, Copenhagen: Denmark, pp. 13-14, 1988.

[15] Kapur K, Garg R, "A Software Reliability Growth Model under Imperfect Debugging", RAIRO, Vol.24, pp. 295-305, 1990.

[16] Kareer N, Kapur P, Grover P, "S-shaped software reliability growth model with two types of errors", Microelectronics and reliability, Vol.13, pp 1085-1090, 1990.

[17] Ahmad N, Khan R, "A study of testing-effort dependent inflection S-shaped software reliability growth models with imperfect debugging", International Journal of Quality Reliability and Management, Vol. 27, pp. 89–110, 2010.

[18] Rafi K, Akthar S, "Imperfect-debugging SRGM with software module testing and resource allocation dependent release policy", International Journal of Computer Applications, Vol. 34, pp. 31–34, 2010.

[19] Ahmada N, Khan R, "Analysis of an inflection s-shaped software reliability model considering log-logistic testing-effort and imperfect debugging", International Journal Computer Science Network Security, Vol. 11, pp.161–171, 2011.

[20] Pachauri B, Kumar A, Dhar J, "Software reliability growth model with dynamic faults and optimal release time optimization using GA and MAUT", Applied Mathematics and computation, Vol.242, pp. 500-509, 2014.

[21] Jain M, Manjula T, Gulati T, "Cost optimization of a software reliability growth model with imperfect debugging and a fault reduction factor", in proceeding of the 11th Biennial Engineering Mathematics and Applications Conference, Vol. 55, pp. C182-C196, 2014.

[22] Pham H, Nordmann L and Zhang Z, "A general imperfect software debugging with S-shaped fault detection rate" IEEE Transactions on Reliability, Vol.48, pp. 169-175, 1999.