SRMS Journal of Mathematical Sciences, Vol-4, 2018, pp. 37-41
ISSN: 2394-725X

Logistic Test Effort Function Based Software Reliability Growth Model with Multi up Gradation

Ajay Kumar Gupta[1], Suneet Saxena[2] and Nancy Achrya[3]
[1]Bhagwant Institute of Technology, Muzaffarnagar U.P., India
[2]Department of Mathematics, SRMS College of Engineering & Technology, Bhojipura, Bareilly U.P, India
[3]Bhagwant Universtiy, Ajmer
e-mail: ajaydr_gupta@yahoo.co.in, Sunarn2012@gmail.com, nancykingdom@rediffmail.com

DOI: https://doi.org/10.29218/srmsjoms.v4i1.14676

Keywords: NHPP, Software Reliability, Multi up-gradation, logistic TEF

Abstract

Paper presents the development of the software reliability growth model under multi-up gradation process. In up gradation process new version of the software is released in the market with some new features. The process is repeated and referred to as multi-up gradation. The model has been developed using logistic test effort function. logistic distribution described increasing/decreasing phenomenon fairly. Three generations of the model have been developed and compared using statistical tools R^2 and MSE. Results validate good fitting of the third generation to a given dataset.

1. Introduction

For the last twenty year the primary concern for a software developer is how to ensure the reliability of software, that is, the software works without failure. Now a day's computers are being used to control safety-critical and civilian systems. In these areas, the demand for complex software system increased many times. As the complexity and size of software increases, the possibility of failures also increases. To improve the reliability of software by analyzing failures, many mathematical models have been developed. Such models are called Software Reliability Growth Model (SRGM). SRGM predicts the number of faults detected with respect to testing time. Reliability depends on faults detected and corrected. Non Homogenous Poisson Process (NHPP) are extensively used to study failure distribution of the software system.

Software reliability depends on several test-cases, testing coverage, CPU hours, Human resource collectively known as Test Effort Functions (TEF). TEF involves two types of constraints namely, time and resource. Time constraint is the limited time available for testing as the software has to be released in the market. Resource constraint (human resource, CPU hours, etc.) is the limited resources available for testing.. Reliability of software depends on TEF and it is necessary to effectively consume TEF to achieve optimum reliability of the software system.
Many authors have developed SRGM incorporating TEF. In 2002 Huang and Kuo [4] investigated a SRGM based on logistic TEF and predicted optimal software release policy involving cost-reliability criteria. In 2008 Ahmad, Bokhari, Quadri, and khan [3] proposed SRGM based on exponential and weibull distribution. They incorporated various TEF and estimated optimal release policy. In 2011, log-logistic TEF with imperfect debugging used by Ahmad, Khan, and Rafi [2]. They analyzed an inflection S- shaped SRGM. In 2012 Agarwal, Kapur, Kaur, and Kumar [1] incorporated various TEF in modular software. They categorized total faults as simple, hard and complex faults. These faults were considered as a function of TEF described by Weibull type distribution. An optimization problem has been formulated of maximizing total faults removed subject to budgetary and reliability constraints. Genetic algorithm has been used to solve the problem. Reddy and Raveendrababu [9] in the year 2012 incorporated generalized exponential TEF while developing SRGM. In 2013 Shinji Inoue and Shigeru Yamada [5] proposed SRGM based on continuous time model such as the lognormal process. They used Wiener process to represent fluctuations in fault detection and approximated fault detection rate with Weibull based test effort function.

In this dynamic world, requirements of end-user changes with time. To meet such requirements, working software has to be upgraded continuously. Railway system, Banking system, Health and Hospitality sector, etc. requires continuous up gradation of a software system. Multi-up gradation is one of the various factors on which software reliability depends. When new features added to working software, the possibility of failure increases. During testing phase faults detected and fixed. The process is termed as up gradation. In multi-up gradation software upgraded continuously with time.

Original software referred to as the first generation. The upgraded first generation software referred to as the Second generation and so on. In 2010 Kapur, Tandon and kaur [6] developed a multi-up gradation SRGM , considering that cumulative faults removed in a generation depend on all previous generation. Kapur et al. [10] in the year 2012 presented multi-up gradation SRGM involving Weibull testing effort function. The particular case discussed using two parameters exponential power distribution function. Recently Li and Yi [7] in year 2016 developed Multi-generation SRGM involving power law function as TEF.
ln this paper we propose Multi-generation SRGM in which fault detection rate approximated by logistic TEF. The Logistic TEF captures increasing/decreasing failure occurrence phenomenon effectively. After the introduction, there are five sections in this paper. Section 2 and 3 presents descriptions of some SRGM and test effort functions respectively. Section 4 discussed the development of a model and solution. Section 5 provides an estimation of the parameters and comparison using statistical tools. Finally, conclusions have been highlighted in section 6.

Notations

E	Total testing effort consumed.
β	Scale parameter in Exponential, Rayleigh and Weibull distribution.
γ	Shape parameter in Weibull and Generalized exponential distribution.
α	Consumption rate of testing effort expenditures in the Logistic TEF.
Λ	Constant parameter in the logistic TEF.
a	Total number of faults

2. Software Reliability Growth Model

2.1 Jelinsky- Moranda Model

J-M model is one of the oldest models. In this model the failure intensity is the product of constant hazard rate ($\emptyset$) of an individual fault and the number of expected faults remaining in the software. The elapsed time between failures follows exponential distribution. The failure intensity at time t is given by

$$\frac{dm}{dt} = \emptyset[a - m(t)].$$

2.2 Goel – Okumoto Model

G-O Model is the first non-homogeneous Poisson Process model that takes the number of faults per unit time as independent Poisson random variables. It is similar to J-M model except for that failure rate decreases with time. Parameters of the model have physical interpretation and can be estimated by various statistical methods using experimental data.

The mean value function is $m(t) = a[1 - \exp(-rt)]$.

2.3 Yamada S-Shaped Model

This NHPP model derived from G-O model. Mean value function of G-O model is of exponential nature. Yamada et al. reasoned that due to learning and skill improvements of the programmer during the debugging phase of the software development cycle, the error detection curve is not exponential but rather S-Shaped.

The mean value function is $m(t) = a[1 - (1 + rt)\exp(-rt)]$.

2.4 Log-Logistic Model

The NHPP models have constant, increasing or decreasing failure occurrence rates per fault. These models were inadequate to capture the failure processes underlying some of the failure data sets, which exhibit an increasing/decreasing failure occurrence rates per fault. The Log-Logistic model was proposed to capture increasing/decreasing failure occurrence rates per fault.

The mean value function is $m(t) = a\frac{(\gamma t)^k}{1+(\gamma t)^k}$. where k is constant.

3. Test Effort Function

3.1 Exponential TEF

It is a non increasing function. The PDF (current testing effort function at any time t) is given by $f(t) = E\beta \exp(-\beta t)$. CDF [cumulative testing effort function consumed in (0,t)] is given by $F(t) = \int_0^t f(t)\,dt = E[1 - \exp(-\beta t)]$.

3.2 Rayleigh TEF

This TEF exhibits both increasing and decreasing phenomenon. PDF is represented by $f(t) = 2E\beta t \exp(-\beta t^2)$ and CDF is $F(t) = E[1 - \exp(-\beta t^2)]$.

3.3 Weibull TEF

It is a generalized case of Exponential and Rayleigh TEF. Also exhibits peak phenomenon initially increasing and then decreasing. Its PDF is $f(t) = \lambda E\beta t^{\lambda-1} \exp(-\beta t^{\lambda})$ CDF is given by $F(t) = E[1 - \exp(-\beta t^{\lambda})]$.

3.4 Logistic TEF

It is a smooth bell-shaped function and represents TEF fairly accurate.PDF is given by $(\quad)ft = \frac{E\lambda\alpha \exp(-\alpha t)}{[1+\lambda \exp(-\alpha t)]^2}$. CDF is given by $F(t) = \frac{E}{[1+\lambda \exp(-\alpha t)]}$.

4. Model Development

The rate of fault detected at time t depends on instantaneous TEF and remaining fault in software at time t.

$$\frac{dm}{dt} = f(t)[a - m(t)] \tag{1}$$

Using condition m(0) = 0 and integrating we get

$$m(t) = a\left[1 - \exp\left(-\int_0^t f(t)\,dt\right)\right] = a[1 - \exp(-(F(t) - F(0)))] \tag{2}$$

Using logistic TEF , we get

$$m(t) = a\left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda \exp(-\alpha t)]} - \frac{1}{[1+\lambda]}\right)\right)\right] \tag{3}$$

$$m(t) = aT(t) \quad \text{where } T(t) = \left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda \exp(-\alpha t)]} - \frac{1}{[1+\lambda]}\right)\right)\right] \tag{4}$$

$T(t)$ is Test Effort Probability Distribution function.

4.1 First Generation

Let a_1 be the total number of faults and $m_1(t)$ number of faults at time t. Software released at time t_1, equation (4) becomes

$$m_1(t) = a_1\left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda \exp(-\alpha t)]} - \frac{1}{[1+\lambda]}\right)\right)\right] \quad 0 \le t \le t_1 \tag{5}$$

4.2 Second Generation

Let a_2 be the number of faults correspond to the second generation and remaining faults of the first generation are $a_1 - m_1(t_1)$. Total faults are $a_2 + a_1 - m_1(t_1)$. If software released at time t_2 then faults $m_2(t)$ detected at any time t given by

$$m_2(t) = [a_2 + a_1 - m_1(t_1)]T(t - t_1) \quad t_1 \le t \le t_2 \tag{6}$$

Where $[a_2 + a_1 - m_1(t_1)] = \left[a_2 + a_1 - a_1\left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda\exp(-\alpha t_1)]} - \frac{1}{[1+\lambda]}\right)\right)\right]\right]$

$$T(t - t_1) = \left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda\exp(-\alpha(t - t_1))]} - \frac{1}{[1+\lambda]}\right)\right)\right]$$

4.3 Third Generation

Let a_3 be the number of faults correspond to the third generation and remaining faults of the second generation are $a_2 - m_2(t_2 - t_1)$. Total faults are $a_3 + a_2 - m_2(t_2 - t_1)$. If software released at time t_3 then faults $m_3(t)$ detected at any time t given by

$$m_3(t) = [a_3 + a_2 - m_2(t_2 - t_1)]T(t - t_3) \qquad t_2 \le t \le t_3 \tag{7}$$

Where $[a_3 + a_2 - m_2(t_2 - t_1)] = [a_3 + a_2 - [a_2 + a_1 - m_1(t_1)]T(t_2 - 2t_1)]$

$$= \left[a_3 + a_2 - \left[a_2 + a_1 - a_1\left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda\exp(-\alpha t_1)]} - \frac{1}{[1+\lambda]}\right)\right)\right]\right]T(t_2 - 2t_1)\right]$$

Also $T(t - t_3) = \left[1 - \exp\left(-\mathrm{E}\left(\frac{1}{[1+\lambda\exp(-\alpha(t - t_3))]} - \frac{1}{[1+\lambda]}\right)\right)\right]$

5. Parameter Estimation and Model Comparisions

5.1 Parameter Estimation

Parameters of models are estimated by using non-linear Least Square Method. Data set used is Apache 2.0.39.

5.2 Model Comparisons

Three generations of models are compared using the following tools of Goodness of Fit (GoF).

5.2.1 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 varies from 0 to 1. If R^2 = 1 then perfect fitting, $R^2 = 0$ no fitting, and R^2 close to 1 good fitting. R^2 is defined as

$$\mathrm{R}^2 = \frac{\sum_{j=1}^{n}(\mathrm{Y_J} - \bar{\mathrm{y}})^2}{\sum_{j=1}^{n}(\mathrm{y_j} - \bar{\mathrm{y}})^2} \qquad \bar{\mathrm{y}} = \frac{1}{\mathrm{n}}\sum_{j=1}^{n}\mathrm{y_j}$$

where $\mathrm{y_j}$ is observed cumulative faults at time j and $\mathrm{Y_J}$ estimated cumulative faults at time j. n is a number of data points. Model fits better to the given dataset if R^2 close to 1.

5.2.2 Mean of Square Error MSE

MSE measures the mean of squares of the deviation between estimated and observed data. It is defined as

$\mathrm{MSE} = \frac{1}{\mathrm{n}}\sum_{j=1}^{n}(\mathrm{Y_J} - \bar{\mathrm{y}})^2$ if MSE close to 0 then model fits data better.

Comparative analysis of three generations are as follows:

Table 1. Goodness of Fit

	R^2	MSE
First Generation	0.799	1.23
Second Generation	0.823	0.78
Third Generation	0.896	0.65

6. Conclusion

Paper discussed multi-up gradation software reliability growth model. The model has been developed using logistic test effort function. logistic distribution described increasing/decreasing phenomenon fairly. Three generations of the model have been developed and compared using statistical tools R^2 and MSE. Table of goodness shows the value of R^2 is higher for the third generation up gradation, and also MSE is closer to zero. Results validate good fitting of the third generation to a given dataset.

References

[1] A.G. Agarwal, P.K. Kapur, G. Kaur, and R. Kumar, “Genetic Algorithm Based Optimal Testing Effort Allocation Problem for Modular Software”, BIJIT-BVICAM’s International Journal of Information Technology, 2012, 4,1, pp 445-451.

[2] N. Ahmad, M.G.M Khan, and L.S. Rafi, “Analysis of an Inflection S-Shaped Software Reliability Model Considering Log-Logistic Testing-Effort and Imperfect Debugging”, International Journal of Computer Science and Network Security,2011, 11, 1, pp 161-171.

[3] 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”, International Journal of Quality and Reliability Management, 2008, 25 (2), pp 211-235.

[4] C.Y. Huang, and S.Y. Kuo, “Analysis of Incorporating Logistic Testing-Effort Function into Software Reliability Modeling”, IEEE Transactions On Reliability,2002, 51, 3, pp 261-270.

[5] S. Inoue and S. Yamada,”Lognormal Process Software Reliability Modeling with Testing Effort”, Journal of Software Engineering and Applications, 2013, Vol 6, pp 8-14.

[6] P.K.Kapur, A.Tandon and G.Kaur, “Multi Up Gradation Software Reliability Model”, 2nd International Conference on Reliability, Safety and Hazard, 2010, IEEE, Mumbai. Pp 468-474.

[7] F. Li and Z.L.Yi, ”A New Software Reliability Growth Model: Multigeneration Faults and a Power-Law Testing Effort Function”, Mathematical Problems in Engineering, 2016, Volume 2016, pp 1-13.

[8] Q. Li and H.Pham, “A testing-coverage software reliability model considering fault removal efficiency and error generation”, PLoS ONE 12(7):e0181524. ttps://doi.org/10.1371/journal. pone.0181524, 2017 , pp 1/25- 25/25.

[9] K.V.S Reddy and B. Raveendrababu, “Software Reliability Growth Model with Testing-Effort by Failure Free Software”, International Journal of Engineering and Innovative Technology, 2012, Vol 2, 6, pp 103-107.

[10] O. Singh, P.K.Kapur and Jyotish N.P. Singh, ”Testing Effort Based Multi Up Gradation Software Reliability Growth Model”, Communications in Dependability and Management, 2012, Vol 15, pp 88-100.

[11] M. Zhu and H.Pham, “A software reliability model with time-dependent fault detection and fault removal”, Vietnam J Comput Sci (2016) Volume 3: pp-71–79.

Received Date: 15 October 2018
Revised Date: 14 May 2019
Published Date: 15 November 2019