R.Satya Prasad et al./ Elixir Comp. Sci. & Engg. 64 (2013) 19124-19129

Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Science and Engineering



Elixir Comp. Sci. & Engg. 64 (2013) 19124-19129

Pareto Type IV Software Reliability Growth Model

R.Satya Prasad^{1,*} and G.Sridevi²

¹Department of CSE, Acharya Nagarjuna University, Nagarjuna Nagar. ²Department of CSE, Nimra Women's College of Engineering, Vijayawada.

ARTICLE INFO

Article history: Received: 4 September 2013; Received in revised form: 28 October 2013; Accepted: 5 November 2013;

Keywords Software Reliability, NHPP, SRGM, Parameter estimation, Interval Domain Data,

ABSTRACT

Software quality depends upon Software Reliability as it is one of its most important characteristics. Software Reliability is an important attribute of software quality, together with functionality, usability, performance, serviceability, capability, installability, maintainability, and documentation. It is hard to be achieved, because the complexity of software tends to be high. The development of many Software Reliability Growth Models (SRGMs) over the last several decades have resulted in the improvement of software facilitating many engineers and managers in tracking and measuring the growth of reliability. This paper proposes Pareto type IV based Software Reliability growth model with interval domain data. The unknown parameters of the model are estimated using the maximum likelihood (ML) estimation approach. Reliability of a software system using Pareto type IV distribution, which is based on Non-Homogenous poisson process (NHPP) is presented through estimation procedures. Comparison is also made between the Pareto type II model and Proposed model. The performance of the SRGM is judged by its ability to fit the software failure data. How good does a mathematical model fit to the data is also being calculated. Besides, an analysis of two software failure data sets is included.

© 2013 Elixir All rights reserved

Introduction

ML estimation.

Software reliability is defined as the probability of failure free software operation for a specified period of time in a specified environment (Lyu, 1996) (Musa et al., 1987).SRGM is a mathematical model of how the software reliability improves as faults are detected and required (Quadri and Ahmad, 2010). Among all SRGMs developed so far a large family of stochastic reliability models based on a Non-Homogeneous poisson process known as NHPP reliability model has been widely used. Software Reliability is the most dynamic quality characteristic which can measure and predict the operational quality of the software system during its intended life cycle. To identify and eliminate human errors in software development process and also to improve software reliability, the Statistical Process Control concepts and methods are the best choice.

If the selected model does not fit the collected software testing data relatively well. We would expect a low prediction ability of this model and the decision makings based on the analysis of this model would be far from what is considered to be optimal decision (xie et al., 2001).

This paper presents Pareto type IV model to analyse the reliability of a software system using interval domain data and compares Pareto type II model with proposed model. In conclusion it is proved that the Reliability of the proposed model is very high for two failure data sets when compared to Pareto type II Model.

Literature Survey

NHPP Models

The NHPP based models are the most important models because of their simplicity, convenience and compatibility. The NHPP based software reliability growth models are proved quite successful in practical software reliability engineering [Musa et al., 1987].

The main issue in the NHPP model is to determine an appropriate mean value function to denote the expected number of failures experienced up to a certain time point. Model parameters can be estimated by using maximum likelihood estimate (MLE).Parameter values can be obtained using Newton Raphson Method. SPC is a powerful tool to optimize the amount of information needed for use in making management decisions. Statistical techniques provide an understanding of the business baselines, insides for process improvements, communication of value and results of processes, and active and visible involvement.

According to probabilistic assumptions there are numerous software reliability growth models available for use. The Non Homogenous Poisson Process (NHPP) based software reliability growth models are proved to be quite successful in practical software reliability engineering. NHPP model formulation is described in the following lines.

A software system is subjected to failures at random times caused by errors present in the system. Let $\{N(t), t > 0\}$ be a counting process representing the cumulative number of failures by time t. Since there are no failures at t=0 we have

$$N(\mathbf{0}) = \mathbf{0}$$

It is reasonable to assume that the number of software failures during non-overlapping time intervals do not affect each other. In other words, for any finite collection of times $t_1 < t_2 < \cdots t_n$. The n random variables $(t_1), \{N(t_2) - N(t_1)\}, \dots, \{N(t_n - N(t_{n-1}))\}$ are independent. This implies that the counting process $\{N(t), t>0\}$ has independent increments.

Let m(t) represent the expected number of software failures by time's'. The mean value function m(t) is finite valued, nondecreasing, non-negative and bounded with the boundary conditions.

$$m(t)=0, t=0$$

$$= a, t \rightarrow \infty$$

Where a is the expected number of software errors to be eventually detected.

Suppose N (t) is known to have a Poisson probability mass function with parameters m(t) i.e.,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, n = 0, 1, 2 \dots \infty$$

Then N(t) is called an NHPP. Thus the stochastic behaviour of software failure phenomena can be described through the N(t) process. Various time domain models have appeared in the literature (Kantam and Subbarao, 2009) which describe the stochastic failure process by an NHPP which differ in the mean value functions m (t).

The Pareto Type II SRGM

The Mean value function is given by

$$m(t) = a \left[1 - \frac{c^b}{(t+c)^b}\right]$$

By using this mean value function, the Reliability of the software is calculated (Satya Prasad et al., 2011 [4]).

The Proposed Pareto Type IV SRGM

In this paper we consider m(t) as given by

$$m(t) = a \left[1 - \left[1 + \left(\frac{t}{c} \right) \right]^{-b} \right]$$

Where [m(t)/a] is the cumulative distribution function of Pareto type IV distribution (Johnson et al, 2004) for the present choice.

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}$$
$$\lim_{n \to \infty} P\{N(t) = n\} = \frac{e^{-a} \cdot a^n}{n!}$$

This is also a Poisson model with mean 'a'.

Let N(t) be the number of errors remaining in the system at time 't'

$$N(t) = N(\infty) - N(t)$$
$$E[N(t)] = E[N(\infty)] - E[N(t)]$$
$$= a - m(t)$$
$$= a - a \left[1 - \left[1 + \left(\frac{t}{c}\right)\right]^{-b}\right]$$

Let S_k be the time between $(k-1)^{th}$ and k^{th} failure of the software product. Let X_k be the time up to the k^{th} failure. Let us find out the probability in the time between $(k-1)^{th}$ and k^{th} failures, i.e. S_k exceeds a real number 's' given that the total time up to the $(k-1)^{th}$ failure is equal to x.

i.e.
$$P[S_k > s/X_{k-1} = x]$$

 $RS_k/X_{k-1} (s/x) = e^{-[m(x+s)-m(s)]}$

Parameter Estimation Based on Interval Domain Data

In this section we develop expressions to estimate the parameters of the Pareto type IV model based on interval domain data. Parameter estimation is of primary importance in software reliability prediction.

A set of failure data is usually collected in one of two common ways, time domain data and interval domain data. In this paper parameters are estimated from the interval domain data.

The mean value function of Pareto type IV model is given by

$$m(t) = a \left[1 - \left[1 + \left(\frac{t}{c} \right) \right]^{-b} \right]$$

In order to have an assessment of the software reliability, a, b and c are to be known or they are to be estimated from software failure data. Expressions are now delivered for estimating 'a', 'b' and 'c' for the Pareto type IV model.

Assuming the given data are given for the cumulative number of detected errors n_i in a given time interval $(0, t_i)$ where i=1,2, n and $0 < t_1 < t_2 < ...t_n$, then the logarithmic likelihood function (LLF) for interval domain data [8] is given by

$$Log L = \sum_{i=1}^{k} (n_i - n_{i-1}) \cdot log[m(t_i) - m(t_{i-1})] - m(t_k)$$

$$Log L = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[log \left[a \left[1 - \left[1 + \left(\frac{t_i}{c} \right) \right]^{-b} \right] \right] - \left[a \left[1 - \left[1 + \left(\frac{t_{i-1}}{c} \right) \right]^{-b} \right] \right] \right] - \left[a \left[1 - \left[1 + \left(\frac{t_k}{c} \right) \right]^{-b} \right] \right]$$
$$= \sum_{i=1}^{k} (n_i - n_{i-1}) \ Log \left[\left(\frac{at_i}{c} \right)^{-b} - \left(\frac{at_{i-1}}{c} \right)^{-b} \right] - \left(\frac{at_k}{c} \right)^{-b}$$
$$\frac{\partial \ Log \ L}{d} = 0$$

$$= \sum_{i=1}^{k} (n_i - n_{i-1}) \left[\left[\frac{-bt_i}{ac} + \frac{bt_{i-1}}{ac} \right] - \left(\frac{t_k}{c} \right)^{-b} \right] = \mathbf{0}$$

$$a = \sum_{i=1}^{k} (n_i - n_{i-1}) + \left[\frac{b(t_{i-1} - t_i)}{c} \right] \left(\frac{c}{t_k} \right)^{b}$$

The parameter 'b' is estimated by iterative Newton Raphson Method using

 $b_{n+1} = b_n - \frac{g(b_n)}{g'(b_n)}$, where g(b) and g'(b) are expressed as follows. $g(b) = \frac{\partial \log L}{\partial b} = 0$

$$g(b) = \frac{\partial \log L}{\partial b} = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[-Log(t_{i-1} + 1) - Log(t_i + 1) + \frac{1}{(t_i + 1)^{b^{-(t_{i-1} + 1)^{b}}} \left[(t_i + 1)^{b} Log(t_i + 1) - (t_{i-1} + 1)^{b} Log(t_{i-1} + 1) \right] \right] \\ + \left[\sum_{i=1}^{k} (n_i - n_{i-1}) + b(t_{i-1} - t_i) Log\left(\frac{1}{t_k + 1}\right) \right] = 0 \\ g'(b) = \frac{\partial^{2 \log L}}{\partial b^2} = 0 \\ g'(b) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[\frac{2(t_{i-1} + 1)^{b} (t_i + 1)^{b} Log(t_i + 1) Log\frac{(t_{i-1} + 1)}{(t_i + 1)^{b} - (t_{i-1} + 1)^{b}]^2}}{\left[(t_i + 1)^{b} - (t_{i-1} + 1)^{b} \right]^2} + \sum_{i=1}^{k} (n_i - n_{i-1}) + b(t_{i-1} - t_i) Log\left(\frac{1}{t_k + 1}\right) \right]$$

∂LoaL

 $C_{n+1=C_n - \frac{g(c_n)}{g'(c_n)}}$ The parameter 'c' is estimated by iterative Newton Raphson Method using

Where g(c) and $g^{1}(c)$ are expressed as follows.

$$g(c) = \frac{-s}{\partial c} = 0$$

$$g(c) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[\left[\frac{1}{c} - \frac{1}{(t_{i-1} + c)} - \frac{1}{(t_i + c)} + \left(\frac{t_{i-1} - t_i}{c}\right) \left(\frac{c}{t_k}\right) \left[\frac{t_k}{(t_k + c)^2}\right] \right] \right] = 0$$

$$g'(c) = \frac{\partial^2 LogL}{\partial c^2} = 0$$

$$g'(c) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[\frac{-1}{c^2} + \frac{1}{(t_{i-1} + c)^2} + \frac{1}{(t_i + c)^2} \right] + \sum_{i=1}^{k} (n_i - n_{i-1}) \left[\frac{t_{i-1} - t_i}{c^2} \left(\frac{1}{(t_k)}\right) \left(\frac{t_k}{(t_k + c)^3}\right) \right]$$

The values of 'b' and 'c' in the above equations can be obtained using Newton Raphson Method. Solving the above equations, simultaneously yields the point estimates of the parameters b and c. These equations are to be solved iteratively and their solutions in turn when substituted gives value of 'a'.

Data Analysis

In this section, we present the analysis of two software failure data sets. The set of software errors analysed here is borrowed from a real software development project as published in Pham (2005), which in turn referred to Pham(2005) as Zhang et al.,(2000). The data are named as Release 2 and Release 3 test data. The release 2 test data are summarized in the below table.

Table 1: Release 2 Test Data (Wood 1996)				
Test Week	CPU Hours	Percent CPU Hours	Defects found	Predicted total defects
1	384	-	13	-
2	1186	-	18	-
3	1471	-	26	-
4	2236	-	34	-
5	2772	-	40	-
6	2967	-	48	-
7	3812	-	61	-
8	4880	-	75	-
9	6104	-	84	-
10	6634	65	89	203
11	7229	70	95	192
12	8072	79	100	179
13	8484	83	104	178
14	8847	86	110	184
15	9253	90	112	184
16	9712	95	114	183
17	10083	98	117	182
18	10174	99	118	183
19	10272	100	120	184
20	-	-	-	-

Solving equations in section 4 by Newton Raphson Method(N-R) method for the Release 2 test data, the iterative solutions for MLEs of a, b and c are

$$a = 107.3145$$

 $b = 0.979778$
 $c = 8.483201$

Hence, we may accept these three values as MLSs of a, b, c. The estimator of the reliability function from the equation at any time x beyond 10272 hours is given by

$${}^{RS_k}/X_{k-1}$$
 (s/x) = e^{-[m(x+s)-m(s)]}

 $\frac{\text{R S}_{21}}{X_{20}}$ (10272 + 8847) = $e^{-[m(8847+10272) - m(10272)]}$ = 0.954471

Table 2: Release3 Test Data(Wood 1996)		
CPU hours	Percent CPU Hours	Defects found
162	-	6
499	-	9
715	-	13
1137	-	20
1799	-	28
2438	-	40
2818	-	48
3574	71	54
4234	84	57
4680	93	59
4955	98	60
5053	100	61

Solving equations by (N-R) method for the Release 3 test data, the iterative solutions for MLSs of a, b and c are

$$a = 55.496875$$

 $b = 0.976831$

c = 5.718852

Hence, we may accept these three values as MLSs of a, b, c. The estimator of the reliability function from the equation at any time x beyond 5053 hours is given by

$$\frac{RS_k}{X_{k-1}} \left(\frac{s}{x}\right) = e^{-[m(x+s)-m(s)]}$$

$$\frac{RS_{21}}{X_{20}} \left(5053 + 4234\right) = e^{-[m(4234+5053)-m(5053)]}$$

$$= 0.967650262$$

The Reliabilities of 2 datasets for two different models are shown in table 3 and the Log values for two models are calculated in table 4.

Table 3: Reliabilities of different data sets			
Dataset	Reliability Pareto Type IV	Reliability Pareto Type II	
(No)			
Release 2	0.954471	0.930823755	
Release 3	0.9676502	0.94798376	

Release 2	0.954471	0.930823755
Release 3	0.9676502	0.94798376

Table 4: Log likelihood on different data sets		
ta set (No)	Pareto Type IV	Pareto Type II

Data set (No)	Pareto Type IV Log L (MLE)	Pareto Type II Log L (MLE)
Release2	-59.31448664	-85.01402669
Release 3	-30.26832376	-43.96251223

Conclusion

The aforesaid proposed model is primarily useful in estimating and monitoring software reliability, viewed as a measure of software quality. Equations to obtain the maximum likelihood estimates of the parameters based on interval domain data are developed. By comparing the Reliabilities of both Pareto type IV model and Pareto type II model, it is proved that the Pareto type IV model has the highest Reliability for all the data sets. Out of the datasets collected, the Release 2 dataset has the highest Negative value for the Log likelihood. Hence Release 2 dataset best fits for both the models. Besides, when Pareto Type IV and Pareto type II are compared, Pareto type II has the highest –ve value. This is a simple method for model validation and is very convenient for practitioners of software reliability.

References

[1] A.L.Goel, K. Okumoto, "Time-dependent Error Detection rate model for software reliability and other performance measures", IEEE Trans. Reliab. 1979, R-28, pp. 206-211.

[2] Pham. H.. 2003. "Handbook of Reliability Engineering", Springer.

[3] Musa J.D., Software Reliability Engineering MCGraw-Hill, 1998

[4] Satya Prasad R and Geetha Rani N(2011), "Pareto type II Software Reliability Growth model", International Journal of Software Engineering, volume 2, Issue(4) 81-86.

[5] Wood, A(1996), "Predicting Software Reliability", IEEE Computer, 2253-2264.

[6] R.R.L. Kantam and R.Subba Rao, 2009, "Pareto Distribution: A Software Reliability Growth model", International Journal of Performability Engineering. Volume 5, Number 3, April 2009, Paper 9, PP: 275-281

[7] Satya Prasad R and Geetha Rani N (2011) "Software Reliability Growth model using Interval Domain Data"", International Journal Of Computer Applications (0975-8887), Volume 34-No 9, November 2011.

[8] R.satya Prasad et al., "A two step approach for parameter estimation of software reliability ",, elixir comp.Sci & Engg (2013).

[9] Wood A., (1996). "Software Reliability Growth Models", Tandem Computers, Technical report 96-1.

[10] J.D.Musa, A.Iannino, K.Okumoto, "Software Reliability: Measurement Prediction Application". MCGraw-Hill, New York. 1987[11] Lyu. M.R. (1996). "Handbook of Software Reliability Engineering", MCGraw-Hill, New York.

[12] S.M.K Quadri, N.Ahmed "Software reliability growth modeling with new modified weibull testing effort and optimal release policy 2010." International journal of computer applications, volume 6 no 12, September 2010.

[13] Xie. M. Goh. T.N. Ranjan P.. (2002). "Some effective control chart procedures for Reliability monitoring", Reliability Engineering and System Safety 77 143-150.

Authors Profile



Dr. R. Satya Prasad received Ph.D.degree in computer science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Guntur, Andhra Pradesh. He have a satisfactory consistent academic track of record and received Gold medal from Acharya Nagarjuna University for his outstanding performance in master's degree. He is currently

working as Associate Professor in the department of Computer Science & Engg., Acharya Nagarjuna University. He has occupied various academic responsibilities like practical examiner, project adjudicator, external member of board of Examiners for various Universities and colleges in and around in Andhra Pradesh. His current research is focused on Software engineering. He has published several papers in National & International Journals.



Mrs. G. Sridevi received M.Sc. and M.Tech degree from Acharya Nagarjuna University. She is currently pursing Ph.D at Department of Computer Science and Engineering ,Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. She is currently working as a Vice-Principal and Associate professor in the Department of Computer Science, Nimra Women's

College of Engineering, Jupudi, Ibrahimpatnam, Vijayawada, Andhra Pradesh. Her research interests lies in Data Mining and Software Engineering.