

Predictions of MLE and LSE in NHPP Software Reliability Model

Kwang-Yoon Song, In-Hong Chang[†], and Seung-Woo Lee

Abstract

We propose a mean value function for software failures in NHPP software reliability model. And we deal with the maximum likelihood estimation and the least squares estimation in the proposed mean value function. The explicit mean value function solution for the proposed model is presented by MLE and LSE in two data sets. The values of SSE and MSE is presented in two data sets by MLE and LSE. We compare the predicted number of faults with the actual two data sets using the proposed mean value function.

Key words: NHPP, Software Reliability Model, MLE, LSE

1. Introduction

A general software reliability model based on the Non-homogeneous Poisson process (NHPP) is used to derive a model that integrates imperfect debugging with the learning phenomenon. Learning occurs if testing appears to improve dynamically in efficiency as one progress through a testing phase. Learning usually manifests itself as a changing fault-detection rate. Published models and empirical data suggest that efficiency growth due to learning can follow many growth-curves, from linear to that described by the logistic function^[1,2]. The software development process becomes increasingly time-consuming and expensive due to the complexity of software systems. In the mean time, the need for the highly reliable system is ever increasing. How to enhance the reliability of the software systems and reduce the cost to an acceptable level becomes the main focus of the software industry^[3]. The pioneering attempt in non-homeogenous Poisson process based on software reliability growth models (SRGM) was made by Goel and Okumoto^[4]. The model describes the failure observation phenomenon by an exponential curve. There are also SRGM that describe either S-shaped curves, or a mixture of exponential and S-shaped curves (flexible). Some of the important contributions of these types of

models were achieved by Yamada^[5] and others^[6-10]. In the last few decades, most SRGM were developed based on the assumptions that faults detected in the testing phase are removed immediately with no debugging time delay, and no new fault introduction. The traditional SRGM of Goel-Okumoto^[4] was based on the common assumption that the faults are removed immediately after a software failure is observed. The time from observation to removal should not be neglected in a practical software testing process. Further, Yamada et al.^[5] described the testing process as a two-stage process. The first stage consists of observing software failure (desired output is not attained) when software is tested. It is the second stage of the testing process in which an effort is made to remove the cause of that failure^[11].

Research activities in software reliability engineering have been conducted over the past 30 years, and many statistical models have been developed for the estimation of software reliability^[12,13]. Software reliability is a measure of how closely user requirements are met by a software system in actual operation. Most existing models for quantifying software reliability are based purely upon observation of failures during the system test of the software product^[4-6,13-21].

This paper is organized as follows: In Section 2, the explicit solution of the mean value function for the NHPP model is presented and new mean value function is proposed. The mean square error as goodness-of-fit criteria is presented for model estimation from actual data. In Section 3, we deal with the maximum likeli-

Department of computer science and statistics, Chosun University, Gwangju, Korea

[†]Corresponding author : ihchang@chosun.ac.kr
(Received : April 17, 2013, Revised : June 20, 2013, Accepted : June 24, 2013)

hood estimation and the least squares estimation in the proposed mean value function. In Section 4, the values of SSE and MSE is presented in two data sets by MLE and LSE. We compare the predicted number of faults with the actual two data sets using the proposed mean value function. Section 5 presents conclusions in this paper and further research direction in software reliability modeling and challenging issues are also discussed.

2. Model Description

Many NHPP models assume that failure intensity is proportional to the residual fault content. A general class of NHPP SRGMs can be obtained by solving the following differential equation^[1,2]:

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

The general solution of the above differential equation is given by

$$m(t) = e^{-B(t)}[m_0 + \int_{t_0}^t a(\tau)b(\tau)e^{B(\tau)}d\tau],$$

where $B(t) = \int_{t_0}^t b(\tau)d\tau$,

and $m(t_0) = m_0$ is the marginal condition of the above differential equation with t_0 representing the starting time of the debugging process^[20].

An increasing function $a(t)$ implies an increasing total number of faults (note that this includes those faults already detected and removed, and those inserted during the debugging process) and reflects imperfect debugging. An increasing $b(t)$ implies an increasing fault detection rate, which could be either attributed to a learning curve phenomenon, or to software process fluctuations, or a combination of both. Different $a(t)$ and $b(t)$ functions also reflect different assumptions on the software testing processes^[20]. $a(t)$ is time-dependent fault-content function: total number of faults in the software, including the initial and introduced faults. $b(t)$ is time-dependent fault-detection rate function per fault per unit time. $m(t)$ is expected number of error detected by time t .

In this paper, the mean value function is given by;

$$m(t) = \alpha(1 + \gamma)(\gamma + e^{-\gamma t} - 1) \tag{1}$$

where α is the expected total number of software faults that exist in a system and γ is a detective rate. This mean value function is special case in Pham model^[22].

Once the analytical expression for the mean value function $m(t)$ is derived, in this paper, the model parameters to be estimated in the mean value function can then be obtained with the help of a developed Matlab program. The term goodness of fit is used in two different contexts. There are common criteria, such as the sum of squared errors (SSE), the mean squared error (MSE), the Akaike's information criterion (AIC) and the predictive-ratio risk (PRR) that are commonly used for model comparison of goodness-of-fit and predictive power and we will use them for model illustrations^[10].

The Sum of squared errors and the mean squared error are given by

$$SSE = \sum_{i=1}^n (m(t_i) - y_i)^2, \quad MSE = \frac{\sum_{i=1}^n (m(t_i) - y_i)^2}{n - N}$$

where y_i is total number of failure observed at time t_i and according to the actual data and $m(t_i)$ is the estimated cumulative number of failure at t_i for $i = 1, 2, \dots, n$.

The MSE measures the distance of a model estimate from the actual data with the consideration of the number n of observations and the number N of parameters in the model. The lower MSE indicates less fitting error, the lower value of MSE is the better the model fits, relative to other models run on the same data set.

The Akaike's information criterion is given by

$$AIC = -2 \cdot \log(\max \text{ of likelihood function}) + 2N,$$

where N is number of parameters in the model. The AIC measures the ability of a model to maximize the likelihood function that is directly related to the degrees of freedom during fitting, increasing the number of parameter will usually result in better fit.

The predictive-ratio risk is given by

$$PRR = \sum_{i=1}^n \frac{\hat{m}(t_i) - y_i}{\hat{m}(t_i)}$$

which measures the distance of model estimates from the actual data against the model estimate.

For all these four criteria the smaller the value, the better model the model fits, relative to other models run on the same data set

3. Maximum Likelihood Estimation (MLE) and Least Squares Estimation (LSE)

In this section, we briefly discuss the maximum likelihood estimators and least squares estimators of parameters in the proposed model.

To obtain MLE from model (1), the probability density function is given by

$$f(t; \alpha, \gamma) = m'(t)e^{-m(t)} \\ = \alpha[2\gamma^2 t - \gamma^2 t e^{-\gamma t}]e^{-[\alpha(1+\gamma)(\gamma t + e^{-\gamma t} - 1)]}$$

Then, the likelihood function of α, γ is given by

$$L(\alpha, \gamma) = \prod_{i=1}^n f_i(x_i) = \left(\prod_{i=1}^n [\alpha(2\gamma^2 x_i - \gamma^2 x_i e^{-\gamma x_i})] \right) \\ \exp \left[-\alpha \sum_{i=1}^n [(1 + \gamma x_i)(\gamma x_i + e^{-\gamma x_i} - 1)] \right]$$

The log-likelihood function of α, γ is given by

$$l(\alpha, \gamma) = n \ln \alpha + \sum_{i=1}^n \ln(2\gamma^2 x_i - \gamma^2 x_i e^{-\gamma x_i}) \\ - \alpha \sum_{i=1}^n [(1 + \gamma x_i)(\gamma x_i + e^{-\gamma x_i} - 1)]$$

The first derivative of the log likelihood function with respect to α is

$$\frac{\partial \ln l}{\partial \alpha} = \frac{n}{\alpha} - \sum_{i=1}^n [(1 + \gamma x_i)(\gamma x_i + e^{-\gamma x_i} - 1)] \quad (2)$$

Setting equation (2) equal to zero, we can obtain the MLE of α as follows

$$\hat{\alpha} = n \left[\sum_{i=1}^n (1 + \gamma x_i)(\gamma x_i + e^{-\gamma x_i} - 1) \right]^{-1}$$

Now, we consider the least square estimation to obtain the least squares estimators, α and γ are unknown is (1), we can use the information provided by the observations in two data sets to give us estimates $\hat{\alpha}$ and $\hat{\gamma}$ of α and γ ; thus we can write

$$\hat{m}(t) = \hat{\alpha}(1 + \hat{\gamma}t)(\hat{\gamma}t + e^{-\hat{\gamma}t} - 1)$$

where $\hat{m}(t)$ denotes the predicted values of mean value function $m(t)$ for a given $\hat{\alpha}$ and $\hat{\gamma}$. Then $\hat{\alpha}$ and $\hat{\gamma}$ is called the least square estimator(LSE), estimators of estimates α and γ that minimize the function;

$$\sum_{t=1}^n (m(t) - \hat{m}(t))^2$$

We will present values of least square estimators of α and γ by numerical method in next section.

4. Numerical Examples

In this section, we present the analysis of two real data sets to compare MLE and LSE the AT&T's System T Table 1(data 1) is a network-management system developed by AT&T that receives data from telemetry events, such as alarms, facility-performance information, and diagnostic messages, and forwards them to operators for further action. The system has been tested and failure data has been collected^[23]. Table 1 shows the failures and the inter-failure as well ad cumulative failure times (in CPU units)^[10].

Table 1. Data 1

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--------|---|---|----|----|----|----|----|----|----|----|
| $x(t)$ | 3 | 6 | 10 | 14 | 14 | 16 | 16 | 17 | 17 | 19 |

We obtain MSE using MLE, $MSE = 20.158$ at $\hat{\alpha} = 5.1297, \hat{\gamma} = 0.17$ when $t = 1, \dots, t = 10$ from Table 1(data 1). Also We obtain MSE using LSE, $MSE = 16.3588$ at $\hat{\alpha} = 0.0001, \hat{\gamma} = 35.86$ when $t = 1, \dots, t = 10$ from Table 1(data 1). These estimated parameters provide minimum MSE in prediction. And then the mean value functions respectively becomes;

Table 2. SSE in data 1 by LSE

| α | γ | SSE |
|----------|----------|----------|
| 0.0001 | 35.86 | 32.718 |
| 0.0001 | 40.45 | 158.363 |
| 0.0001 | 50.33 | 1625.024 |
| 0.0010 | 13.26 | 261.120 |
| 0.0010 | 15.91 | 1620.223 |
| 0.0030 | 9.19 | 1626.046 |
| 0.0050 | 5.08 | 32.755 |
| 0.0100 | 3.00 | 185.614 |
| 0.0100 | 4.00 | 133.387 |
| 0.0100 | 5.03 | 1615.813 |
| 0.0200 | 2.50 | 34.203 |
| 0.0300 | 0.78 | 1282.959 |
| 0.1000 | 1.04 | 78.421 |

$$m(t) = 5.1297(1+0.17t)(0.17t+e^{-0.17t}-1),$$

$$m(t) = 0.0001(1+35.86t)(35.86t+e^{-35.86t}-1)$$

Table 2 is SSE in data 1 using LSE and Table 3 is SSE in data 1 using MLE and Table 4 is predicted values

Table 3. SSE in data 1 by MLE

| α | γ | SSE |
|----------|----------|------------|
| 0.0001 | 0.001 | 1725.000 |
| 0.0109 | 0.010 | 1724.984 |
| 0.3474 | 0.050 | 1709.740 |
| 1.6111 | 0.100 | 1412.308 |
| 1.9874 | 0.110 | 1261.385 |
| 3.3689 | 0.140 | 589.441 |
| 3.9138 | 0.150 | 337.460 |
| 4.5008 | 0.160 | 130.778 |
| 5.1297 | 0.170 | 40.316 |
| 5.8005 | 0.180 | 172.479 |
| 7.2672 | 0.200 | 1776.818 |
| 8.0629 | 0.210 | 3751.247 |
| 17.07027 | 0.300 | 176886.906 |

Table 4. Predicted value from data 1

| T | Real Data | LSE | MLE |
|-----|-----------|----------|----------|
| 1 | 3 | | |
| 2 | 6 | | |
| 3 | 10 | | |
| 4 | 14 | | |
| 5 | 14 | | |
| 6 | 16 | | |
| 7 | 16 | | |
| 8 | 17 | | |
| 9 | 17 | | |
| 10 | 19 | | |
| 11 | 20 | 15.55977 | 15.07741 |
| 12 | 20 | 18.51743 | 18.24578 |
| 13 | 21 | 21.73228 | 21.73066 |
| 14 | 22 | 25.20432 | 25.53168 |
| MSE | | 16.3588 | 20.158 |

Table 5. Data 2

| t | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| $x(t)$ | 1 | 1 | 2 | 3 | 5 | 5 | 5 | 8 | 9 | 11 | 13 | 15 | 19 | 19 | 22 |

Table 6. SSE in data 2 by LSE

| α | γ | SSE |
|----------|----------|-----------|
| 0.0001 | 25.960 | 62.333 |
| 0.0001 | 32.280 | 1063.787 |
| 0.0200 | 1.820 | 63.518 |
| 0.0200 | 2.300 | 1149.705 |
| 0.1000 | 0.790 | 83.414 |
| 0.1000 | 0.950 | 439.943 |
| 0.1000 | 1.000 | 835.590 |
| 0.1520 | 1.020 | 6092.133 |
| 0.2000 | 1.200 | 35841.631 |
| 0.3000 | 0.500 | 98.339 |
| 0.5000 | 0.400 | 158.805 |
| 0.5000 | 0.300 | 482.209 |
| 1.0000 | 0.200 | 726.324 |

from data 1 by LSE and MLE.

The data set in Table 5 (data 2) was reported by Zhang et al^[24]. based on system test data for a telecommunication system. System test data consisting of release is shown in Table 5. In both tests, automated test and human-involved tests are executed on multiple test beds^[10].

We obtain MSE using MLE, $MSE = 5.7003$ at $\hat{\alpha} = 2.9106$, $\hat{\gamma} = 0.159$ when $t = 1, \dots, t = 15$ from Table 5 (data 2). Also We obtain MSE using LSE, $MSE = 4.7948$ at $\hat{\alpha} = 0.0001$, $\hat{\gamma} = 25.96$ when $t = 1, \dots, t = 15$ from Table 5 (data 2). These estimated parameters provide minimum MSE in prediction. And then the mean value functions respectively becomes;

$$m(t) = 2.9106(1+0.159t)(0.159t+e^{-0.159t}-1),$$

$$m(t) = 0.0001(1+25.96t)(25.96t+e^{-25.96t}-1)$$

Table 6 is SSE in data 2 using LSE and Table 7 is SSE in data 2 using MLE and Table 8 is predicted values from data 2 by LSE and MLE.

Fig. 1 and Fig. 2 show MSE values in respect of

Table 7. SSE in data 2 by MLE

| α | γ | SSE |
|----------|----------|----------|
| 0.000066 | 0.001 | 3417.000 |
| 0.007204 | 0.010 | 3416.960 |
| 0.229722 | 0.050 | 3375.599 |
| 0.648945 | 0.080 | 3092.778 |
| 1.060839 | 0.100 | 2983.152 |
| 1.307440 | 0.110 | 2203.781 |
| 2.211073 | 0.140 | 674.377 |
| 2.871297 | 0.158 | 77.002 |
| 2.910568 | 0.159 | 74.103 |
| 2.950111 | 0.160 | 76.039 |
| 3.753122 | 0.179 | 1508.624 |
| 3.843353 | 0.181 | 1881.584 |
| 4.754580 | 0.200 | 9190.350 |

Table 8. Predicted value of data 2

| T | Real Data | LSE | MLE |
|-----|-----------|---------|---------|
| 1 | 1 | | |
| 2 | 1 | | |
| 3 | 2 | | |
| 4 | 3 | | |
| 5 | 5 | | |
| 6 | 5 | | |
| 7 | 5 | | |
| 8 | 8 | | |
| 9 | 9 | | |
| 10 | 11 | | |
| 11 | 13 | | |
| 12 | 15 | | |
| 13 | 19 | | |
| 14 | 19 | | |
| 15 | 22 | | |
| 16 | 22 | 17.2523 | 16.7367 |
| 17 | 23 | 19.4762 | 19.0768 |
| 18 | 24 | 21.8350 | 21.5725 |
| 19 | 24 | 24.3285 | 24.2231 |
| 20 | 24 | 26.9568 | 27.0282 |
| 21 | 26 | 29.7198 | 29.9871 |
| MSE | | 4.7948 | 5.7003 |

parameter α and γ from data 1 by LSE and MLE, Fig. 3 and Fig. 4 show MSE values in respect of parameter α and γ from data 2 by LSE and MLE.

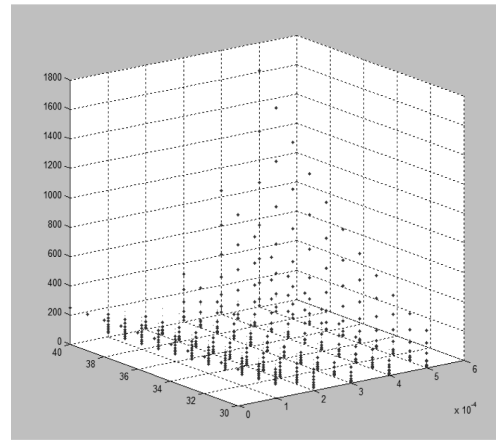


Fig. 1. MSE of LSE in data 1.

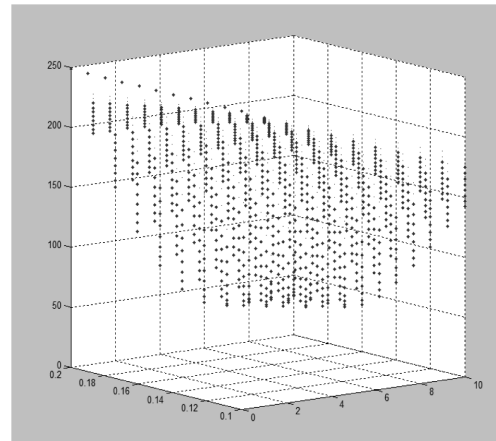


Fig. 2. MSE of MLE in data 1.

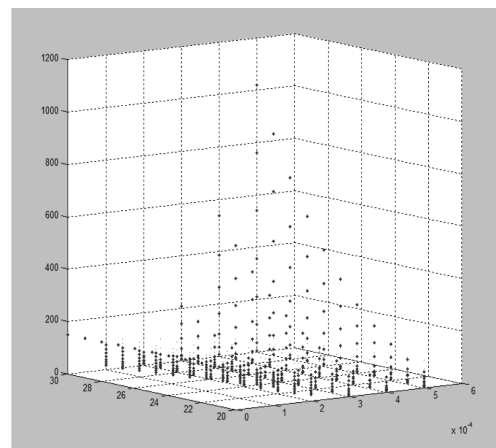


Fig. 3. MSE of LSE in data 2.

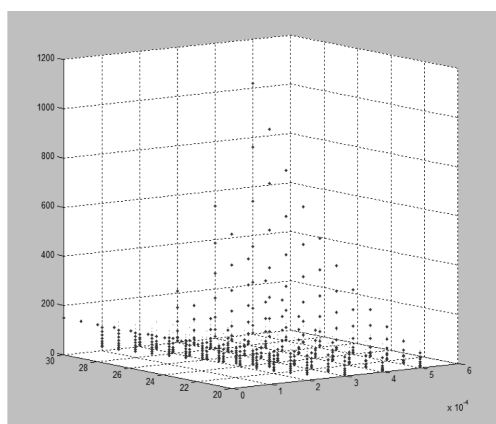


Fig. 4. MSE of MLE in data 2.

5. Conclusion

We considered a mean value function for software failures in NHPP software reliability model as a special case in Pham's model^[22]. And we obtained the maximum likelihood estimator and the least squares estimator in the proposed mean value function. The explicit mean value function solution for the proposed model was presented by MLE and LSE in two data sets. The values of SSE and MSE was presented in two data sets by MLE and LSE. We compared the predicted number of faults with the actual two data sets using the proposed mean value function. Finally we confirmed the predicted values of faults presented by MLE and LSE are very similar values.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012-0006750).

References

- [1] H. Pham and X. Zhang, "A software cost model with warranty and risk costs", *IEEE T. Comput.*, Vol. 48, pp. 71-75, 1999.
- [2] H. Pham and X. Zhang, "Software release policies with gain in reliability justifying the costs", *Ann. Softw. Eng.*, Vol. 8, pp. 147-166, 1999.
- [3] R. P. Grag, K. Sharma, R. Kumar, and R. K. Grag, "Performance analysis of software reliability models using matrix method", *World Academy of Science, Engineering and Technology*, Vol. 47, pp. 31-38, 2010.
- [4] A. L. Goel and K. Okumoto, "Time dependent error detection rate model for software reliability and other performance measures", *IEEE T. Reliab.*, Vol. R-28, pp. 206-211, 1979.
- [5] S. Yamada, M. Ohba, and S. Osaki, "S-shaped reliability growth modeling for software fault detection", *IEEE T. Reliab.*, Vol. 32, pp. 475-484, 1983.
- [6] M. Ohba, "Software reliability analysis models", *IBM J. Res. Dev.*, Vol. 28, pp. 428-443, 1984.
- [7] M. Ohba, "Inflection S-shaped software reliability growth models", *Stochastic models in reliability theory*, S. Osaki and Y. Hatoyama Eds., Springer-Verlag, Berlin, Vol. 235, pp. 144-162, 1984.
- [8] P. K. Kapur and R. B. Garg, "A software reliability growth model for an error removal phenomenon", *Software Engineering Journal*, Vol. 7, pp. 291-294, 1992.
- [9] P. K. Kapur, R. B. Garg, and S. Kumar, "Contributions to hardware and software reliability", *World Scientific Publishing Co. Ltd.*, Singapore, 1999.
- [10] H. Pham, "System Software Reliability", Springer, Berlin, pp. 149-149, 2006.
- [11] P. K. Kapur and H. Pham, "A unified approach for developing software reliability growth models in the presence of imperfect debugging and error generation", *IEEE T. Reliab.*, Vol. 60, pp. 331-340, 2011.
- [12] A. Wood, "Predicting software reliability". *IEEE Computer*, Vol. 11, pp. 69-77, 1996.
- [13] L. Pham and H. Pham, "Software reliability models with time dependent hazard function based on Bayesian approach", *IEEE T. Syst. Man. Cy. A*, Vol. 30, pp. 25-35, 2000.
- [14] Z. Jelinski and P. B. Moranda, "Software reliability research", *Statistical computer performance evaluation*, W. Freiberger Ed., Academic Press, New York, pp. 465-497, 1972.
- [15] H. Pham, "Software reliability assessment: imperfect debugging and multiple failure types in software development", *EGandG-RAAM-10737*, Idaho National Engineering Laboratory, 1993.
- [16] H. Pham, "A software cost model with imperfect debugging, random life cycle and penalty cost", *Int. J. Syst. Sci.*, Vol. 27, pp. 455-463, 1996.
- [17] H. Pham and X. Zhang, "An NHPP software reliability models and its comparison", *Int. J. Rel. Qual. Saf. Eng.*, Vol. 4, pp. 269-282, 1997.
- [18] H. Pham, L. Nordmann, and X. Zhang, "A general

- imperfect software debugging model with S-shaped fault detection rate”, IEEE T. Reliab., Vol. 48, pp. 169-175, 1999.
- [19] S. Yamada and S. Osaki, “Software reliability growth modeling: Models and applications”, IEEE T. Software Eng., Vol. 11, pp. 1431-1437, 1985.
- [20] H. Pham, “Software reliability and cost models: perspectives, comparison, and practice”, Eur. J. Oper. Res., Vol. 149, pp. 475-789, 2003.
- [21] H. Pham, “Software Reliability”, Springer-Verlag, 2000.
- [22] H. Pham, “An Imperfect-debugging fault-detection dependent-parameter software”, International Journal of Automation and Computing, Vol. 4, pp. 325-328, 2007.
- [23] W. Ehrlich, B. Prasanna, J. Stampfel, and J. Wu, “Determining the cost of a stop-testing decision”, IEEE Software, Vol, 10, pp. 33-42, 1993.
- [24] X. Zhang, D. R. Jeske and H. Pham, “Calibrating software reliability models when the test environment does not match the user environment”, Applied Stochastic Models in Business and Industry, Vol. 18, pp. 87-99, 2002.