J Syst Sci Syst Eng (Feb 2017) 26(1):121-132 ISSN: 1004-3756 (Paper) 1861-9576 (Online)
DOI: 10.1007/s11518-016-5322-4 CN11-2983/N

A THREE-PARAMETER FAULT-DETECTION SOFTWARE RELIABILITY MODEL WITH THE UNCERTAINTY OF OPERATING ENVIRONMENTS

Kwang Yoon Song[1] **In Hong Chang**[1] **Hoang Pham**[2]

[1]*Department of Computer Science and Statistics, Chosun University, Gwangju, 62022, Korea*
skyjacker152@naver.com, ihchang@chosun.ac.kr (✉)
[2]*Department of Industrial and Systems Engineering, Rutgers University, NJ 08855-8018, USA*
hopham@rci.rutgers.edu

Abstract

As requirements for system quality have increased, the need for high system reliability is also increasing. Software systems are extremely important, in terms of enhanced reliability and stability, for providing high quality services to customers. However, because of the complexity of software systems, software development can be time-consuming and expensive. Many statistical models have been developed in the past years to estimate software reliability. In this paper, we propose a new three-parameter fault-detection software reliability model with the uncertainty of operating environments. The explicit mean value function solution for the proposed model is presented. Examples are presented to illustrate the goodness-of-fit of the proposed model and several existing non-homogeneous Poisson process (NHPP) models based on three sets of failure data collected from software applications. The results show that the proposed model fits significantly better than other existing NHPP models based on three criteria such as mean squared error (MSE), predictive ratio risk (PRR), and predictive power (PP).

Keywords: Nonhomogeneous Poisson process, software reliability, mean squared error, predictive ratio risk, predictive power, fault detection

1. Introduction

As requirements on system quality increase, the need for high system reliability is also increasing. Software systems are extremely important, in terms of enhanced reliability and stability, for providing high quality services to customers. Software systems improve the solution of immediate problems in a variety of industries, and continue offering customer convenience. However, because of the complexity of software systems, the software development process can be time-consuming and expensive. Enhancing the reliability of software systems and reducing cost to acceptable levels have become the main focus of the software industry (Grag 2010). Meanwhile, research has long been performed in software reliability engineering, and many software reliability growth models (SRGM) have been proposed. Many existing non-homogeneous Poisson

© Systems Engineering Society of China and Springer-Verlag Berlin Heidelberg 2017

process (NHPP) software reliability models have been conducted through the fault intensity rate and mean value $m(t)$ functions within a controlled testing environment in order to estimate reliability metrics, such as the number of residual faults, failure rate, and software reliability. In general, these models are applied to software testing data, and then used to make predictions on software failures and reliability in the field. The existing proposed models have the common assumption that the operating environments and the developing environment are about the same. For this reason, once software systems are introduced, the software product used in field environments are the same as, or close to, those used in the development - testing environment. However, such systems might be used in many different locations. The operating environments in the field for the software, in reality, are quite different. The uncertainty of the operating environments will affect the software failure and software reliability. During the system test, software developers execute test cases that mimic an ender user's operational profile. However, the operational profile of the test environment might not exactly match up with the operational profile of the operating environment (Zhang and Pham 2006). Zhu et al. (2015) recently revisited the 32 environmental factors that studied more than a decade ago by Zhang and Pham (2000) and reinvestigated the impact of software development environmental factors on software reliability assessment.

Some researchers, such as Yang and Xie (2000), Huang et al. (2000), and Zhang et al. (2002), proposed a method for predicting the fault detection rate to reflect changes in operating environments and attempted a methodology that modifies the software reliability model in operating environments by introducing a calibration factor. Teng and Pham (2006) discussed a generalized model that captures environment uncertainty and its effects on software failure rates. Recently, Pham (2013, 2014) developed a new software reliability model that incorporates the uncertainty of system fault-detection rate per unit of time subject to operating environments. Chang et al. (2014) also developed a new software reliability model in the software development process and related it to the error detection rate function with consideration of the uncertainty of operating environments.

In this paper, we present a new model with consideration of a three-parameter fault-detection rate in the software development process, and relate it to the error detection rate function with consideration of the uncertainty of operating environments. We examine the goodness-of-fit of the fault-detection rate software reliability model and other existing NHPP models based on several sets of software testing data. The explicit solution of the mean value function for the new model is derived in Section 2. Criteria for model comparisons and selecting the best model are discussed in Section 3. Model analysis and results are discussed in Section 4. Section 5 presents conclusions and remarks.

2. A Fault-detection Rate Software Reliability Model

In this section, an NHPP software reliability model with the uncertainty of operating environments is presented. The following are the assumptions for this model; 1) The occurrence of software failures follows an NHPP, 2) Software can fail during execution, caused by faults in the software, 3) The software-failure detection rate at any time is proportional to the number of remaining faults in the software at that time, 4) When a software failure occurs, a debugging effort removes the faults immediately, 5) For each debugging effort, whether the faults is successfully removed or not, some new faults may be introduced into the software system, 6) The environment affects the unit failure detection rate, $b(t)$, by multiplying a factor η. A generalized NHPP model that incorporates the uncertainty of operating environments can be formulated as follows (Pham 2014):

$$\frac{dm(t)}{dt} = \eta[b(t)][N - m(t)], \qquad (1)$$

where

- $m(t)$ expected number of errors detected by time t, or the mean value function
- N expected number of faults that exist in the software before testing
- $b(t)$ fault detection rate function, which also represents the average failure rate of a fault
- η random variable that represents the uncertainty of the system fault detection rate in the operating environments with probability density function g

The closed-form solution function $m(t)$ with the initial condition $m(0) = 0$ is given in terms of random variable η:

$$m(t) = N\left[1 - e^{-\eta\int_0^t b(x)dx}\right]. \qquad (2)$$

Some recent studies [1, 13, 18] have assumed that the uncertainty of the system operating environment random variable η follows a gamma distribution. Although the exponential distribution is a special case of gamma distribution, we believe that this exponential distribution can be of help in our study in order to keep the number of unknown parameters in the model as low as possible. With that in mind, in this paper we assume that η has an exponential distribution with parameter β, i.e., $\eta \sim \exp(\beta)$, where the probability density function of η is given by

$$g(x) = \beta e^{-\beta x}, \text{for } \beta > 0, x \geq 0,$$

then, from equation (2), we obtain the following:

$$m(t) = N\left[1 - \left(\frac{\beta}{\beta + \int_0^t b(x)dx}\right)\right]. \qquad (3)$$

In this paper, we consider a three-parameter fault-detection rate $b(t)$ is a non-decreasing function with inflexion S-shaped curve, which captures the learning process of the software developers. A three-parameter fault-detection rate $b(t)$ as follows:

$$b(t) = \frac{a}{1 + ce^{-bt}}, \quad a, b, c > 0. \qquad (4)$$

From equation (4), a new three-parameter fault-detection software reliability model, where the expected number of software failures detected by time t, $m(t)$, is subject to the uncertainty of the environments, can be obtained directly from equation (3):

$$m(t) = N\left[1 - \left(\frac{\beta}{\beta - \frac{a}{b}\ln\left(\frac{(1+c)e^{-bt}}{1+ce^{-bt}}\right)}\right)\right]. \qquad (5)$$

Example results show the estimated parameter and performance of the proposed model above (both equations (4) and (5)) based on several sets of real failure data and a set of criteria for model comparisons will be discussed in the next two sections.

3. Criteria for Model Comparisons

Once the analytical expression for the mean value function $m(t)$ is derived, the model parameters to be estimated for this function can be obtained with the help of a developed Visual C++ program and MS Office Excel 2010 based on the least square estimate (LSE) method.

Table 1 Software reliability models

No.	Model	$m(t)$
1	G-O Model [3]	$m(t)=a\left(1-e^{-bt}\right)$
2	Delayed S-Shaped SRGM [21]	$m(t)=a\left(1-(1+bt)e^{-bt}\right)$
3	Inflection S-Shaped SRGM [23]	$m(t)=\dfrac{a\left(1-e^{-bt}\right)}{1+\beta e^{-bt}}$
4	Yamada Imperfect Debugging Model [22]	$m(t)=a\left[1-e^{-bt}\right]\left[1-\dfrac{\alpha}{b}\right]+\alpha at$
5	PNZ Model [15]	$m(t)=\dfrac{a\left[1-e^{-bt}\right]\left[1-\dfrac{\alpha}{b}\right]+\alpha at}{1+\beta e^{-bt}}$
6	Pham-Zhang Model [16]	$m(t)=\dfrac{(c+a)\left[1-e^{-bt}\right]-\left[\dfrac{ab}{b-\alpha}\left(e^{-\alpha t}-e^{-bt}\right)\right]}{1+\beta e^{-bt}}$
7	Dependent-parameter Model 1 [11]	$m(t)=\alpha(1+\gamma t)\left(\gamma t+e^{-\gamma t}-1\right)$
8	Dependent-parameter Model 2 [11]	$m(t)=m_0\left(\dfrac{\gamma t+1}{\gamma t_0+1}\right)e^{-\gamma(t-t_0)}+\alpha(\gamma t+1)\left(\gamma t-1+(1-\gamma t_0)e^{-\gamma(t-t_0)}\right)$
9	Testing Coverage Model [1]	$m(t)=N\left[1-\left(\dfrac{\beta}{\beta+(at)^b}\right)^{\alpha}\right]$
10	Vtub-shaped Model [13]	$m(t)=N\left[1-\left(\dfrac{\beta}{\beta+a^{t^b}-1}\right)^{\alpha}\right]$
11	Proposed New Model	$m(t)=N\left[1-\left(\dfrac{\beta}{\beta-\dfrac{a}{b}\ln\left(\dfrac{(1+c)e^{-bt}}{1+ce^{-bt}}\right)}\right)\right]$

Three common criteria (Pham, 2006), such as the mean squared error (MSE), the predictive ratio risk (PRR), and the predictive power (PP), are used as criteria for the model estimation of the goodness-of-fit, and to compare the proposed model and other existing models as listed in Table 1. MSE, PRR and PP are given as follows:

$$MSE = \frac{\sum_{i=0}^{n}\left(m(t_i) - y_i\right)^2}{n-m},$$

$$PRR = \sum_{i=0}^{n}\left(\frac{\hat{m}(t_i) - y_i}{\hat{m}(t_i)}\right)^2,$$

$$PP = \sum_{i=0}^{n}\left(\frac{\hat{m}(t_i) - y_i}{y_i}\right)^2,$$

where y_i is total number of failures observed at time t_i; m is the number of unknown parameters in the model; and $m(t_i)$ is the estimated cumulative number of failures at t_i for $i = 1, 2, \cdots, n$. MSE measures the distance of a model estimate from the actual data with consideration of the number of observations, n, and the number of unknown parameters in the model, m. PRR measures the distance of model estimates from the actual data against the model estimate. PP measures the distance of model estimates from the actual data against the actual data. For all these three criteria (MSE, PRR, and PP) the smaller the value, the better the model fits relative to other models run on the same data set. Table 1 summarizes the proposed model and several existing well-known NHPP models with different mean value functions. Note that models 9 and 10 in Table 1 did consider environmental uncertainty.

4. Parameter Estimation and Model Comparison

Data set #1, listed in Table 2, was extracted from information on failures in the software development for the real-time multi-computer complex of the US Naval Fleet Computer Programming Center of the US Naval Tactical Data System (NTDS) (Goel 1979). The software consists of 38 different project modules. The time horizon is divided into four phases: production, test, user, and subsequent test. A total of 26 software failures were found during the production phase, five during the test phase, and the last failure was found on 4 January 1971. One failure was observed during the user phase, in September 1971, and two failures during the test phase in 1971. Data set #2, given in Table 3, is the data collected from testing system T at AT&T (Ehrlich 1993). AT&T's system T 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 operation for further action. The system has been tested and failure data has been collected. Table 3 lists failures and inter-failures, as well as cumulative failure times (in CPU units). Detailed information can be obtained from Ehrlich (1993) and Pham (2006). Data set #3, listed in Table 4, was reported by Jeske and Zhang (2005). The software in this case study runs on an element within a wireless network switching center. Its main function includes routing voice channels and signaling messages to relevant radio resources and processing entities. The cumulative field exposure time of the software was 58,633 system-days and a total of 33 failures were observed in the field, amongst which there were 19 unique failures. Table 4 shows the failure data, in grouped format, for each of the 13 months (Jeske and Zhang 2005).

Table 2 NTDS data set – data set #1

t	data	t	data	t	data
1	2	11	26	21	31
2	8	12	27	22	31
3	14	13	28	23	31
4	21	14	30	24	31
5	22	15	30	25	31
6	23	16	30	26	31
7	23	17	30	27	32
8	23	18	30	28	33
9	26	19	31	29	34
10	26	20	31		

Table 3 AT&T data set – data set #2

t	data	t	data
1	3	8	17
2	6	9	17
3	10	10	19
4	14	11	20
5	14	12	20
6	16	13	21
7	16	14	22

Table 4 WNSC data set – data set #3

Month Index	System Days	System Days (Cum.)	Failures	Cum. Failures
1	1249	1249	4	4
2	3472	4721	6	10
3	4065	8786	4	14
4	4883	13669	3	17
5	5425	19094	6	23
6	5656	24750	1	24
7	7549	32299	2	26
8	8295	40594	4	30
9	8882	49476	1	31
10	6120	55596	0	31
11	2465	58061	1	32
12	527	58588	1	33
13	45	58633	0	33

Tables 5, 6, 7 and 8 summarize the results of the estimated parameters for all 11 models in Table 1 using the LSE technique and the three common criteria (MSE, PRR, and PP) values. We obtained the three common criteria when $t = 1, 2, \cdots, 29$ from Data set #1 (Table 2) and when $t = 1, 2, \cdots, 14$ from Data set #2 (Table 3). In addition, we obtained when cumulative system days from Data set #3 (Table 4). As can be seen from Table 6, the PRR, and PP values for the proposed new model are the lowest. Furthermore, the PRR and PP values for the proposed new model are the lowest compared with all models in Table 7. Finally, the MSE value for the proposed new model is the lowest in Table 8. Figures 1, 2, and 3 show the graph of mean value functions for all 11 models for Data sets #1, #2, and #3, respectively.

Table 5 Parameter estimation in models

Model	Data Set #1	Data Set #2	Data Set #3
G-O Model [3]	$\hat{a}$ =31.717, $\hat{b}$ =0.190	$\hat{a}$ =22.760, $\hat{b}$ =0.186	$\hat{a}$ =33.010, $\hat{b}$ =0.000058
Delayed S-shaped SRGM [21]	$\hat{a}$ =30.413, $\hat{b}$ =0.458	$\hat{a}$ =20.002, $\hat{b}$ =0.520	$\hat{a}$ =31.013, $\hat{b}$ =0.000144
Inflection S-shaped SRGM [23]	$\hat{a}$ =31.717, $\hat{b}$ =0.190, $\hat{\beta}$ =0.00001	$\hat{a}$ =22.760 , $\hat{b}$ =0.186, $\hat{\beta}$ =0.001	$\hat{a}$ =33.009, $\hat{b}$ =0.000058, $\hat{\beta}$ =0.00000001
Yamada Imperfect Debugging Model [22]	$\hat{a}$ =28.234, $\hat{b}$ =0.234, $\hat{\alpha}$ =0.0063	$\hat{a}$ =17.425, $\hat{b}$ =0.265, $\hat{\alpha}$ =0.0250	$\hat{a}$ =33.009, $\hat{b}$ =0.000058, $\hat{\alpha}$ =0.00000001
PNZ Model [15]	$\hat{a}$ =22.829, $\hat{b}$ =1.015, $\hat{\alpha}$ =0.017, $\hat{\beta}$ =12.048	$\hat{a}$ =11.673, $\hat{b}$ =1.148, $\hat{\alpha}$ =0.066, $\hat{\beta}$ =7.799	$\hat{a}$ =32.740, $\hat{b}$ =0.00006, $\hat{\alpha}$ =0.00000001, $\hat{\beta}$ =0.002
Pham-Zhang Model [16]	$\hat{a}$ =31.420, $\hat{b}$ =0.211, $\hat{\alpha}$ =2.903, $\hat{\beta}$ =0.00, $\hat{c}$ =0.002	$\hat{a}$ =22.109, $\hat{b}$ =0.210, $\hat{\alpha}$ =4.033, $\hat{\beta}$ =0.00, $\hat{c}$ =0.001	$\hat{a}$ =33.008, $\hat{b}$ =0.000058, $\hat{\alpha}$ =10.00, $\hat{\beta}$ =0.00, $\hat{c}$ =0.001
Dependent-Parameter Model1 [11]	$\hat{\alpha}$ =0.000001, $\hat{\gamma}$ =243.500	$\hat{\alpha}$ =0.000002, $\hat{\gamma}$ =276.9	$\hat{\alpha}$ =2230.17, $\hat{\gamma}$ =0.000003
Dependent-Parameter Model2 [11]	$\hat{a}$ =286.14, $\hat{\gamma}$ =0.011, $\hat{t}_0$ =4.43, $\hat{m}_0$ =21.23	$\hat{\alpha}$ =218.657, $\hat{\gamma}$ =0.024 $\hat{t}_0$ =2.81, $\hat{m}_0$ =10.75	$\hat{\alpha}$ =632.23, $\hat{\gamma}$ =0.000004 $\hat{t}_0$ =0.00, $\hat{m}_0$ =15.20
Testing Coverage Model [1]	$\hat{a}$ =0.731, $\hat{b}$ =4.297, $\hat{\alpha}$ =0.124, $\hat{\beta}$ =0.736, $\hat{N}$ =39.99	$\hat{a}$ =2.021, $\hat{b}$ =2.4945, $\hat{\alpha}$ =0.062, $\hat{\beta}$ =5.888, $\hat{N}$ =63.682	$\hat{a}$ =0.0099, $\hat{b}$ =0.721, $\hat{\alpha}$ =1.001, $\hat{\beta}$ =61.10, $\hat{N}$ =52.24
Vtub-shaped Model [13]	$\hat{a}$ =1.300, $\hat{b}$ =0.001, $\hat{\alpha}$ =3.880, $\hat{\beta}$ =0.005, $\hat{N}$ =33.01	$\hat{a}$ =40.600, $\hat{b}$ =0.0001, $\hat{\alpha}$ =5.40, $\hat{\beta}$ =0.011, $\hat{N}$ =24.20	$\hat{a}$ =1.0001, $\hat{b}$ =0.0001, $\hat{\alpha}$ =1.1460, $\hat{\beta}$ =0.0002, $\hat{N}$ =40.77
Proposed New Model	$\hat{a}$ =0.175, $\hat{b}$ =3.755, $\hat{c}$ =0.497, $\hat{\beta}$ =49.680, $\hat{N}$ =35.389	$\hat{a}$ =0.072, $\hat{b}$ =2.323, $\hat{c}$ =3.520, $\hat{\beta}$ =0.302, $\hat{N}$ =27.684	$\hat{a}$ =0.0059, $\hat{b}$ =0.000001, $\hat{c}$ =34.60, $\hat{\beta}$ =2.870, $\hat{N}$ =41.54

Table 6 Comparison criteria from NTDS data – data set #1

Model	MSE	Rank	PRR	Rank	PP	Rank
G-O Model	2.2957	4	0.5504	7	3.1876	7
Delayed S-shaped SRGM	3.8051	9	0.2324	4	0.2179	4
Inflection S-shaped SRGM	2.384	5	0.5504	6	3.1875	6
Yamada Imperfect Debugging	2.5117	8	0.5758	8	4.0719	8

Model	MSE	Rank	PRR	Rank	PP	Rank
PNZ Model	1.401	3	0.1214	2	0.1863	3
Pham-Zhang Model	2.3916	6	0.3945	5	1.2342	5
Dependent-Parameter Model 1	227.7507	11	3639.2277	11	11.6177	10
Dependent-Parameter Model 2	32.7889	10	1.6766	10	92.6538	11
Testing Coverage Model	1.1342	1	0.191	3	0.132	2
Vtub-shaped Model	2.5222	7	0.5835	9	4.0556	9
Proposed New Model	1.2571	2	0.0708	1	0.0673	1

Table 7 Comparison criteria from AT&T data – data set #2

Model	MSE	Rank	PRR	Rank	PP	Rank
G-O Model	0.8969	4	0.1170	6	0.1517	6
Delayed S-shaped SRGM	1.6446	9	0.3841	9	0.1987	8
Inflection S-shaped SRGM	0.9786	6	0.1167	5	0.1509	5
Yamada Imperfect Debugging	0.8996	5	0.1430	8	0.2180	9
PNZ Model	0.3740	1	0.0853	4	0.0607	3
Pham-Zhang Model	1.1234	7	0.0617	2	0.0606	2
Dependent-Parameter Model 1	59.2875	11	495.6490	11	4.8606	10
Dependent-Parameter Model 2	12.4631	10	0.9502	10	6.6993	11
Testing Coverage Model	0.7993	2	0.0667	3	0.0666	4
Vtub-shaped Model	1.1637	8	0.1276	7	0.1764	7
Proposed New Model	0.8702	3	0.0490	1	0.0492	1

Table 8 Comparison criteria from WNSC data – data set #3

Model	MSE	Rank	PRR	Rank	PP	Rank
G-O Model	1.626	4	0.6277	6	0.24255	5
Delayed S-shaped SRGM	7.3713	9	65.1778	10	1.1664	9
Inflection S-shaped SRGM	1.7887	6	0.6279	6	0.24257	7
Yamada Imperfect Debugging	1.787	5	0.6278	5	0.24256	6
PNZ Model	2.0232	7	0.5604	4	0.2266	4
Pham-Zhang Model	2.2359	8	0.628	7	0.2426	8
Dependent-Parameter Model 1	152.13	11	66696.988	11	5.4583	10
Dependent-Parameter Model 2	33.593	10	1.0252	9	8.3639	11
Testing Coverage Model	1.3975	3	0.0546	1	0.0695	1
Vtub-shaped Model	1.2856	2	0.2462	3	0.1285	3
Proposed New Model	1.2405	1	0.2134	2	0.1165	2

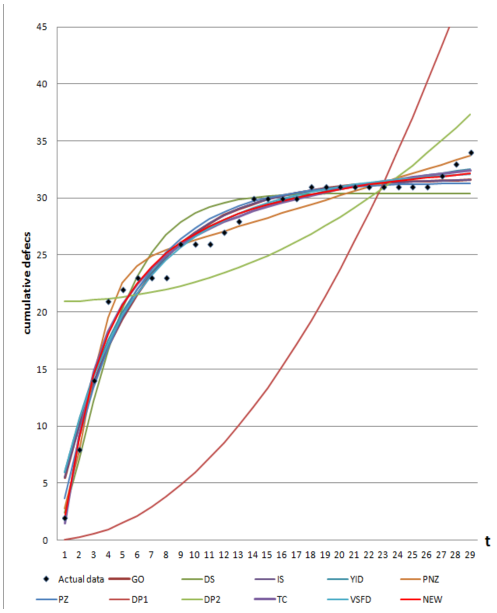

Figure 1 Mean value functions for all 11 models in Table 1 for data set #1

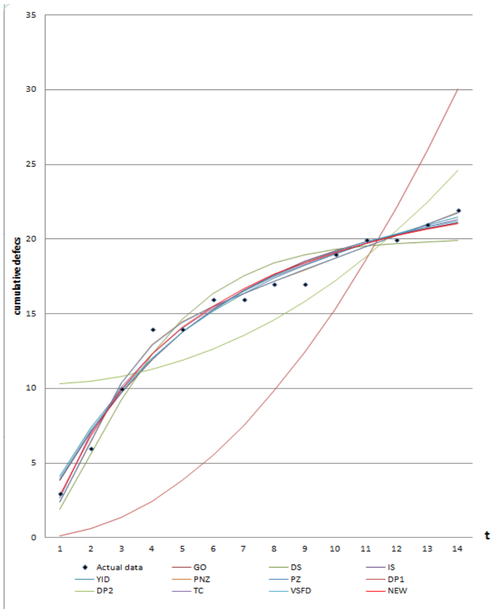

Figure 2 Mean value function for all 11 models in Table 1 for data set #2

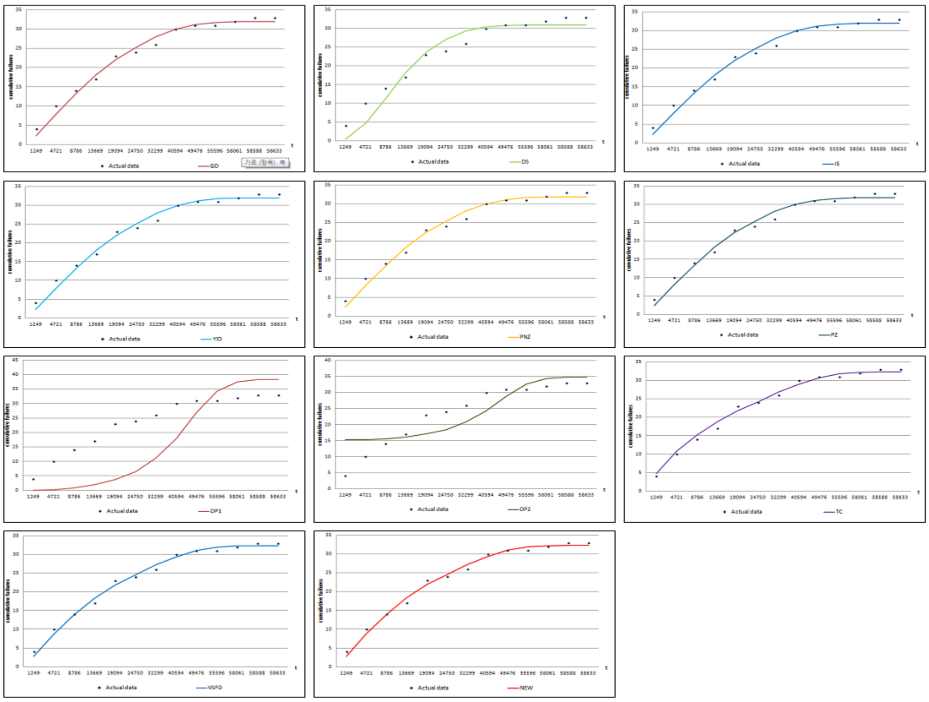

Figure 3 Mean value function for each 11 models in Table 1 for data set #3

5. Conclusions and Remarks

In general, existing proposed models have the common assumption that the testing and operating environment are the same as, or close to. However, systems might be used in many different locations. In this paper, we discussed a new software reliability model subject to the uncertainty of operating environments. The three sets of software failure data are used to illustrate the proposed new model. We summarized the results of the estimated parameters for all 11 models and the three common criteria (MSE, PRR, and PP) values. As can be seen from Table 6, the PRR, and PP values for the proposed new model are the lowest. Furthermore, the PRR and PP values for the proposed new model are the lowest compared with all models in Table 7, and the MSE value for the proposed new model is the lowest in Table 8. The results shown in Tables 6 – 8 show that the proposed new model fits significantly better than other ten existing NHPP models based on three common criteria. Future work in broader validation of this conclusion is needed based on recent data-sets.

Acknowledgments

The authors would like to thank the reviewers for their comments and suggestions. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2009277).

References

[1] Chang, I.H., Pham, H., Lee, S.W. & Song, K.Y. (2014). A testing-coverage software reliability model with the uncertainty of operation environments. International Journal of Systems Science: Operations & Logistics, 1 (4): 220-227.

[2] Ehrlich, W., Prasanna, B., Stampfel, J. & Wu, J. (1993). Determining the cost of a stop testing decision. IEEE Software, 10 (2): 33-42.

[3] Goel, A.L. & Okumoto, K. (1979). Time dependent error detection rate model for software reliability and other performance measures. IEEE Transactions on Reliability, R-28 (3): 206-211.

[4] Grag, R.P., Sharma, K., Kumar, R. & Grag, R.K. (2010). Performance analysis of software reliability models using matrix method. World Academy of Science, Engineering and Technology, 4 (11): 31-38.

[5] Huang, C.Y., Kuo, S.Y., Lyu, M.R. & Lo, J.H. (2000). Quantitative software reliability modeling from testing from testing to operation. Proceedings of the International Symposium on Software Reliability Engineering IEEE, Los Alamitos, CA, USA: 72-82.

[6] Jeske, D.R. & Zhang, X. (2005). Some successful approaches to software reliability modeling in industry. The Journal of Systems and Software, 74 (1): 85-99.

[7] Ohba, M. (1984). Inflexion s-shaped software reliability growth models. In: Osaki, S. & Hatoyama Y. (eds.), Stochastic Models in Reliability Theory, pp. 144-162. Berlin: Springer-Verlag.

[8] Pham, H. (1996). A software cost model with imperfect debugging, random life cycle and penalty cost. International Journal of Systems Science, 27 (5): 455-463.

[9] Pham, H. (2000). Software Reliability. Singapore: Springer.

[10] Pham, H. (2006). System Software Reliability. London: Springer.

[11] Pham, H. (2007). An imperfect-debugging fault-detection dependent-parameter software. International Journal of Automation and Computing, 04 (4): 325-328.

[12] Pham, H. (2013). A software reliability model with Vtub-shaped fault-detection rate subject to operating environments. Proceedings of the 19th ISSAT International Conference on Reliability and Quality in Design, August 5-7, Hawaii, USA, 33-37.

[13] Pham, H. (2014). A new software reliability model with Vtub-shaped fault detection rate and the uncertainty of operating environments. Optimization, 63 (10): 1481-1490.

[14] Pham, H. & Deng, C. (2003). Predictive-ratio risk criterion for selecting software reliability models. In: Proceedings of the 9th ISSAT International Conference on Reliability and Quality in Design.

[15] Pham, H., Nordmann, L. & Zhang, X. (1999). A general imperfect software debugging model with s-shaped fault detection rate. IEEE Transactions on Reliability, 48 (2): 169-175.

[16] Pham, H. & Zhang, X. (1997). An NHPP software reliability models and its comparison. International Journal of Reliability, Quality and Safety Engineering, 4 (3): 269-282.

[17] Pham, L. & Pham, H. (2000). Software reliability models with time dependent hazard function based on Bayesian approach. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 30 (1): 25-35.

[18] Sgarbossa, F., Persona, A. & Pham, H. (2015). Using systemability function for periodic replacement policy in real environments. Quality and Reliability Engineering International, 31 (4): 617-633.

[19] Teng, X. & Pham, H. (2004). Software cost model for quantifying the gain with considerations of random field environments. IEEE Transactions on Computers, 53 (3): 380-384.

[20] Teng, X. & Pham, H. (2006). A new methodology for predicting software reliability in the random field environments. IEEE Transactions on Reliability, 55 (3): 458-468.

[21] Yamada, S. & Osaki, S. (1985). Software reliability growth modeling: models and applications. IEEE Transactions on Software Engineering, 11 (12): 1431-1437.

[22] Yamada, S., Tokuno, K. & Osaki, S. (1992). Imperfect debugging models with fault introduction rate for software reliability assessment. International Journal of Systems Science, 23 (12): 2241-2252.

[23] Yamada, S., Ohba, M. & Osaki, S. (1983). S-shaped reliability growth modeling for software fault detection. IEEE Transactions on Reliability, 32 (5): 475-484.

[24] Yang, B. & Xie, M. (2000). A study of operational and testing reliability in software reliability analysis. Reliability Engineering and System Safety, 70 (3): 323-329.

[25] Zhang, X., Jeske, D. & Pham, H. (2002). Calibrating software reliability models when the test environment does not match the user environment. Applied Stochastic Models in Business and Industry, 18 (1): 87-99.

[26] Zhang, X. & Pham, H. (2000). An analysis of factors affecting software reliability. Journal of Systems and Software, 50 (1): 43-56.

[27] Zhang, X. & Pham, H. (2006). Software field failure rate prediction before software deployment. Journal of Systems and Software, 79 (3): 291-300.

[28] Zhang, X., Teng, X. & Pham. H. (2003). Considering fault removal efficiency in software reliability assessment. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 33 (1): 114-120.

[29] Zhu, M., Zhang, X. & Pham, H. (2015). A comparison analysis of environmental factors affecting software reliability. Journal of Systems and Software, 109: 150-160.

Kwang Yoon Song received the PhD degree in the Department of Computer Science and Statistics at Chosun University, Korea in 2016. He received the BS degrees from Chosun University, Korea in 2009. His research interests include reliability, cost model and Bayesian statistics.

In Hong Chang is a professor in the Department of Computer Science and Statistics at Chosun University, Korea. He received his MS and PhD degrees from Hanyang University, Korea. He has served as an editor and a board member of many journals relating to statistics and reliability. He is a member of the Korean Statistical Society and the Korean Reliability Society. His research interests include reliability, statistical inference and Bayesian statistics.

Hoang Pham is a Distinguished Professor and former Chairman (2007-2013) of the Department of Industrial and Systems Engineering at Rutgers University. He received his MS in statistics from the University of Illinois at Urbana-Champaign, and both MS and PhD in industrial engineering from State University of New York at Buffalo. His research interests include software reliability and system reliability modelling. He has been served as editor-in-chief and editor of several journals. He is the editor of Springer Book Series in Reliability Engineering and has served as Conference Chair and Program Chair of over 30 international conferences. He is the author or coauthor of 6 books and has published over 140 journal articles, and edited 10 books. His numerous awards include 2009 IEEE Reliability Society Engineer of the Year Award. He is a Fellow of the IEEE and IIE.