American Journal of Software Engineering and Applications
2013; 2(3): 92-104
Published online July 20, 2013 (http://www.sciencepublishinggroup.com/j/ajsea)
doi: 10.11648/j.ajsea.20130203.12

A discussion of software reliability growth models with time-varying learning effects

Chiu, Kuei-Chen

Institute of Allied Health Sciences, College of Medicine, National Cheng Kung University, Tainan, Taiwan

Email address:
ckj0214@ms24.hinet.net

To cite this article:
Chiu, Kuei-Chen, A Discussion of Software Reliability Growth Models with Time-Varying Learning Effects. *American Journal of Software Engineering and Applications*. Vol. 2, No. 3, 2013, pp. 92-104. doi: 10.11648/j.ajsea.20130203.12

Abstract: Over the last few decades, software reliability growth models (SRGM) has been developed to predict software reliability in the testing/debugging phase. Most of the models are based on the Non-Homogeneous Poisson Process (NHPP), and an S or exponential-shaped type of testing behavior is usually assumed. Chiu et al. (2008) provided an SRGM that considers learning effects, which is able to reasonably describe the S and exponential-shaped behaviors simultaneously. This paper considers both linear and exponential-learning effects in an SRGM to enhance the model in Chiu et al. (2008), assumes the learning effects depend on the testing-time, and discusses when and what learning effects would occur in the software development process. This research also verifies the effectiveness of the proposed models with *R* square (*Rsq*), and compares the results with these of other models by using four real datasets. The proposed models consider constant, linear, and exponential-learning effects simultaneously. The results reveal the proposed models fit the data better than other models, and that the learning effects occur in the software testing process. The results are helpful for the software testing/debugging managers to master the schedule of the projects, the performance of the programmers, and the reliability of the software system.

Keywords: Software Reliability, Non-Homogeneous Poisson Process (NHPP), Learning Effects, Time-Varying Learning Effects

1. Introduction

Recently, various software reliability growth models (SRGM) have been proposed, and there has been a gradual but marked shift in the balance between acceptable software reliability and affordable software testing cost. Chiu et al. (2008) provided an SRGM that learning effects, which is able to reasonably describe the S and exponential-shaped types of behaviors simultaneously, and has good performance in fitting various data. However, the learning effects in this model are assumed be a constant, even though, in some cases they will vary. This research thus added time-varying learning effects to Chiu's model (2008), using both linear and exponential functions to explain these in the software testing process. This paper also presents numerical examples to verify the effectiveness of the proposed models, and compares other models by using four actual datasets.

2. Literatures Review

For the last few decades a number of SRGM have been developed based on the Non-homogeneous Poisson Process (NHPP). These models generally assume that each time an error occurs the fault that caused it can be immediately removed, leading to no new problems, which is usually called perfect debugging, although imperfect debugging has also been discussed (Ohba, 1984b). Gokhale and Trivedi (1999) stated that the assumption of statistical independence for the number of events occurring in disjointed time intervals is constantly violated when SRGM based on NHPP are used, and proposed an enhanced NHPP model to allow the time-varying failures in the debugging process.

In some cases, due to different testing/debugging strategies or resource allocation processes, software reliability may non-monotonically increase or decrease, due to the change-point in SRGM. Zhao (1993) identified the change-point problem, and stated that effect on software reliability should be estimated. Shyur (2003) developed a

generalized reliability growth model by incorporating imperfect debugging with change-points, while Huang (2005) incorporated both a generalized logistic testing-effort function and the change-point parameter into an SRGM.

Another important issue in the field of SRGM is discovering the confidence intervals of software reliability. Yamada and Osaki (1985) stated that the maximum likelihood estimates (MLE) concerning the confidence interval of the mean value function can be estimated using as $m(T) \pm Z_{CR/2}\sqrt{m(T)}$, where CR denotes the critical region, and $Z_{CR/2}$ denotes the value providing an area $CR/2$ of the standard normal distribution. Yin and Trivedi (1999) presented the confidence bounds for the model parameters via the Bayesian approach by using the estimation method in Yamada and Osaki (1985). Huang (2005) also employed the method in Yamada and Osaki (1985) to draw a graph to illustrate the confidence intervals of the mean value function.

Some researchers in the field of software reliability consider that variations in the mean value function stem from the error detection process, and deduce the mean value function by using stochastic differential equations (SDE) (Lee *et al.*, 2004; Tamura and Yamada, 2006). Although SDE is recognized as a more effective method in evaluating the mean value function and the software testing costs, the existing models still have some problems. For instance, Yamada *et al.* (1994) proposed a simple SRGM by applying an Itô type of SDE where the error detection rate is constant, although in practice the error detection rate would vary over time. Lee *et al.* (2004) developed an SRGM by using an Itô type SDE based on the delayed S-shaped and the inflection S-shaped models proposed by Yamada *et al.* (1983) and Ohba (1984), respectively, without considering the variation in mean value function, which is a crucial factor for measuring testing costs and the variation in software reliability during the testing phase. Tamura and Yamada (2006) derived a flexible SDE model to describe the fault-detection process during the system testing phase of a distributed development environment by using an inflection S-shaped SRGM and an Itô type SDE (Karatsas and Shreve, 1997), but this model improperly presents the prediction and variance of the mean value function, and there is no complete process for obtaining the parameters in the model.

Furthermore, the notion of learning as an explicit feedback process has found its way into many areas of the social and management sciences. Learning effects usually occur when staff are involved in a production or service activity, and it is important to investigate how these will affect task times and costs. In recent years, several studies have noted that learning effects exist in the process of software testing/debugging, but few have succeeded in identify them (Smiarowski, 2006; Kapur, 2007; Chiu, 2008, 2009, 2011, 2012).

3. Method

3.1. Notations

Chiu et al. (2008) presented an SRGM that considers constant learning effects, while this paper provides an SRGM which assumes learning effects vary with testing time.

Software reliability growth models are mathematical functions that describe the error detection and removal process.The following notations will be used throughout this study:

- a : the expected number of all potential errors in the software system
- α : the autonomous errors-detected factor
- η : the learning factor
- $\eta_1(t)$: the learning effect function with linear learning effect
- $\eta_2(t)$: the learning effect function with exponential learning effect
- ξ : the accelerative factor with time of learning effect
- τ : the negligent factor, which means the testing staff would cause another error when removing potential errors
- $f(t)$: the intensity function that denotes the fraction of the errors detected at time t
- $F(t)$: the cumulative function that denotes the fraction of the errors detected within time (0, t)
- $m(t)$: the mean value function of the software error detection process, which is the expected number of errors detected within time $(0,t)$
- $m_1(t)$: the mean value function with linear learning effects
- $m_2(t)$: the mean value function with exponential learning effects
- $\lambda(t)$: the intensive value function of the software error detection process, which is the expected number of errors detected at time t
- $\lambda_1(t)$: the intensive value function with linear learning effects
- $\lambda_2(t)$: the intensive value function with exponential learning effects
- $d(t)$: the error detection rate per error at time t
- $d_1(t)$: the error detection rate per error at time t with linear learning effects
- $d_2(t)$: the error detection rate per error at time t with exponential learning effects
- $R(x/t)$: the conditional software reliability, which is defined as the probability that no error is detected within the time interval $(t,t+x)$

Generally, the software testing/debugging process is modeled as an error counting process. A counting process $\{N(t),\ t \geq 0\}$ is said to be an NHPP with intensity

function $\lambda(t)$, where $N(t)$ follows a Poisson distribution with mean function $m(t)$, this probability can be formulated as:

$$\Pr\left(N(t)=k\right)=\frac{[m(t)]^k e^{-m(t)}}{k!},\quad k=0,1,2,... \quad (3.1)$$

The mean value function $m(t)$, which is the expected number of errors detected within time $(0,t)$, and can be expressed as:

$$m(t)=\int_0^t \lambda(x)dx \quad (3.2)$$

The conditional software reliability $R(x/t)$ is defined as the probability that no error is detected within the time interval $(t,t+x)$, given that an error occurred at time t $(t\geq 0,\ x>0)$. Therefore $R(x/t)$ can be formulated as:

$$R(x/t)=e^{-[m(t+x)-m(t)]}. \quad (3.3)$$

Note that the value of conditional software reliability is approximated to 1 when $t\to\infty$.

3.2. Model development

In this paper, we describe the learning effect that occurs in a software testing/debugging task and discuss the time-varying learning effects of this. Further, we explain how the time-varying learning effects influence the process of software reliability growth.

In the proposed model, the influential factors considered for finding errors in software system include the autonomous errors-detected factor α, the negligent factor τ, the learning factor η, and the accelerative factor ξ. In Chiu's model (2008), which supposes that $f(t)$ is the intensity function that denotes the percentage of the errors detected at time t, $F(t)$ is the cumulative function that denotes the percentage of the errors detected within time (0, t), and $1-F(t)$ is the percentage of the errors as undetected yet at time t. In Chiu's model (2008), the interrelationships among the factors can be formulized as a differential equation, and which is given by:

$$f(t)=\left(\alpha+\eta F(t)-\tau\right)\left(1-F(t)\right). \quad (3.4)$$

Where, $\alpha>0$ and $\alpha>\tau\geq 0$ to specify that a constructive debugging activity is in process.

Equation (3.4) implies that imperfect debugging and learning effects may exist. The autonomous errors-detected factor α indicates that the testing staff/software developers spontaneously find software errors of which they were unaware. Meanwhile, the learning factor η indicates that the testing staff/software developers deliberately set out to find software errors from patterns which were previously detected. The negligent factor τ indicates that new software errors are generated while correcting the program code. The first two factors can improve the efficiency of software debugging, but the third cannot. $F(t)$ and $f(t)$ can be derived from Equation (3.4) by using differential equation analysis to solve Equation (3.5):

$$f(t)=\frac{dF(t)}{dt}=-\eta F(t)^2-(\alpha-\eta-\tau)F(t)+(\alpha-\tau), \quad (3.5)$$

, and the explicit solution is given by:

$$F(t)=\frac{e^{(\alpha+\eta-\tau)(t+c)}-\frac{\alpha-\tau}{\eta}}{1+e^{(\alpha+\eta-\tau)(t+c)}}. \quad (3.6)$$

Note that the number of system errors found should be zero when a system's debugging task begins, but $F(t)$ is not equal to zero when t=0. In view of this, we utilize the constant c to adjust the function $F(t)$, so as to let $F(0)=0$. Based on the above discussion, the constant c can be inferred as:

$$c=\frac{\ln\left(\frac{\alpha-\tau}{\eta}\right)}{\alpha+\eta-\tau}, \quad (3.7)$$

To simplify the model, we define $\alpha'=\alpha-\tau$, and thus $F(t)$ and $f(t)$ can be rearranged as follows:

$$F(t)=1-\frac{1+\frac{\eta}{\alpha'}}{\frac{\eta}{\alpha'}+e^{(\alpha'+\eta)t}}, \quad (3.8)$$

$$f(t)=\frac{(\alpha'+\eta)^2 e^{(\alpha'+\eta)t}}{\alpha'\left(\frac{\eta}{\alpha'}+e^{(\alpha'+\eta)t}\right)^2}. \quad (3.9)$$

Furthermore, we need to estimate the total potential errors in the software system before the testing/debugging task starts. Here, a represents the expected number of all potential errors before the debugging task starts, which can be estimated by the size of the software system along with experience of previous testing/debugging tasks. Accordingly, the mean value function of the software error detection process can be written as:

$$m(t)=aF(t)=a\left(1-\frac{1+\frac{\eta}{\alpha'}}{\frac{\eta}{\alpha'}+e^{(\alpha'+\eta)t}}\right), \quad (3.10)$$

the intensity function at time t is given by:

$$\lambda(t)=af(t)=a\left(\frac{(\alpha'+\eta)^2 e^{(\alpha'+\eta)t}}{\alpha'(\frac{\eta}{\alpha'}+e^{(\alpha'+\eta)t})^2}\right). \quad (3.11)$$

In order to find the variation of the error detection rate per error at time t, we identify the error detection rate as:

$$d(t)=\frac{\lambda(t)}{a-m(t)}=(\alpha'+\eta)\left(1-\frac{\eta}{\alpha' e^{(\alpha'+\eta)t}+\eta}\right). \quad (3.12)$$

Note that the boundary of the error detection rate is between α' and $\alpha'+\eta$ at any testing time. However, the way to distinguish the exponential and the S-shaped behaviors is whether the value of the inflection point is greater than zero or not. The proposed model behaves as S-shaped if $\eta>\alpha'$ (the value of an inflection point is

positive); otherwise it behaves as exponential-shaped. The inflection point implies that learning effects exist in the process of testing/debugging, and inflection time is given by:

$$t_{\text{inflection}} = \frac{\ln(\eta) - \ln(\alpha')}{\alpha' + \eta} > 0. \quad (3.13)$$

However, in some special cases, the learning factor η may not be constant, but instead vary over time. In this study, we adopt two functions to deal with the time-varying situation, and these functions are given as (3.14), (3.15), to represent the linear and exponential growth of learning effects with time.

$$\eta_1(t) = \eta + \xi t, \text{ and} \quad (3.14)$$

$$\eta_2(t) = \eta e^{\xi t}, \quad (3.15)$$

Where ξ is the coefficient of the accelerative factor. The mean value function in Chiu's model (2008) could be improved with these two different learning styles as given by:

$$m_1(t) = a\left(1 - \frac{1 + \frac{\eta + \xi t}{\alpha'}}{\frac{\eta + \xi t}{\alpha'} + e^{\alpha' t + \eta t + \xi t^2}}\right), \text{ and} \quad (3.16)$$

$$m_2(t) = a\left(1 - \frac{1 + \frac{\eta e^{\xi t}}{\alpha'}}{\frac{\eta e^{\xi t}}{\alpha'} + e^{(\alpha' + \eta e^{\xi t})t}}\right), \quad (3.17)$$

The learning effects will be constant while $\xi = 0$, and the mean value functions (3.16), (3.17) will degenerate to Equation (3.10), flexibly.

The intensity functions with these two different learning styles are given by:

$$\lambda_1(t) = \left\{a\left[\frac{\left(1 + \frac{\eta + \xi t}{\alpha'}\right)\left(\frac{\xi}{\alpha'} + (\alpha' + \eta + 2\xi t)e^{\alpha' t + \eta t + \xi t^2}\right)}{\left(\frac{\eta + \xi t}{\alpha'} + e^{\alpha' t + \eta t + \xi t^2}\right)^2} - \frac{\xi}{\alpha'\left(\frac{\eta + \xi t}{\alpha'} + e^{\alpha' t + \eta t + \xi t^2}\right)}\right]\right\}$$

, and

(3.18)

$$\lambda_2(t) = \left\{a\left[\frac{\left(1 + \frac{\eta + e^{\xi t}}{\alpha'}\right)\left(\frac{\xi e^{\xi t}}{\alpha'} + \left(\alpha' + \eta e^{\xi t} + \eta \xi t e^{\xi t}\right)e^{\alpha' t + \eta e^{\xi t}}\right)}{\left(\frac{\eta + e^{\xi t}}{\alpha'} + e^{\alpha' t + \eta e^{\xi t}}\right)^2} - \frac{\xi}{\alpha'\left(\frac{\eta + e^{\xi t}}{\alpha'} + e^{\alpha' t + \eta e^{\xi t}}\right)}\right]\right\}$$

(3.19)

, and the functions of the error detection rate with the two different learning styles are given by:

$$d_1(t) = \frac{\alpha'\left(e^{(\alpha' + \eta + \xi t)t}\left((\alpha' + \eta + \xi t)(\alpha' + \eta + 2\xi t) - \xi\right) + \xi\right)}{(\alpha' + \eta + \xi t)\left(e^{(\alpha' + \eta + \xi t)t}\alpha' + \eta + \xi t\right)},$$

(3.20), and

$$d_2(t) = \frac{\alpha'\left(e^{(\alpha' + \eta e^{\xi t})t}\alpha'^2 + \xi\eta e^{\xi t} + \eta e^{(\alpha' + \xi + \eta e^{\xi t})t}\left(\alpha'(2 + \xi t) - \xi\right) + e^{(\alpha' + 2\xi + \eta e^{\xi t})t}(1 + \xi t)\eta^2\right)}{\left(\alpha' + \eta e^{\xi t}\right)\left(e^{(\alpha' + \eta e^{\xi t})t}\alpha' + \eta e^{\xi t}\right)}$$

(3.21)

Since the learning effects can change over time, the proposed model can provide more flexibility in the graphic mapping of the mean value function.

3.3. Parameter estimation

Least squares estimation (LSE) is adopted to validate the proposed model as this is effective when the corresponding equations are complex and must be solved numerically. Fitting the proposed models to actual error data involves estimating the model's parameters from the datasets. The parameters a, α', η, *and* ξ can be obtained by using numerical methods.

4. Results

4.1. Model comparisons

Chiu et al. (2008) evaluated the effectiveness of the SRGM from the perspective of constant learning effects by using eight datasets from eight published papers, and compared the proposed model with six others by using the *MSE, MAE, AE,* and *Rsq* comparison criteria. This paper improved Chiu's model (2008) by including time-varying learning effects, as in equations (3.14) and (3.15), and evaluated its effectiveness by using the recent four datasets in Zhang and Pham (1998), Pham (2003), Bai, Hu, Xie, and Ng (2005), and Jeske and Zhang (2005) (see Table 4.1).

Table 4.1. *Sources of the datasets*

Reference	Dataset
Zhang and Pham (1998)	Failure data of Misra system
Pham (2003)	Failure data of a real-time control system
Bai, Hu, Xie, and Ng (2005)	Failure data of space program
Jeske and Zhang (2005)	Failure data of wireless data service system

Furthermore, this study compared the proposed models with three others (see Table 4.2) using the *Rsq* comparison criteria.

Table 4.2. *Summary of m(t) for the various models*

Model	Mean value function
Huang (2005)	$m(t) = a(1 - e^{-rW^*(t)})$; $W(t) = \frac{N}{\sqrt[k]{1 + Ae^{-\alpha\kappa t}}}$, $W^*(t) = W(t) - W(0)$

Pham and Zhang (2003)	$m(t)=\frac{1}{1+\beta e^{-bt}}\left[(a+c)(1-e^{-bt})-\frac{ab}{b-\alpha}(e^{-\alpha t}-e^{-bt})\right]$
Chiu (2008)	$m(t)=aH(t)=a\left(1-\frac{1+\frac{\eta}{\alpha}}{\frac{\eta}{\alpha}+e^{(\alpha+\eta)t}}\right)$
Proposed linear learning model	$m_1(t)=a\left(1-\frac{1+\frac{\eta+\xi t}{\alpha'}}{\frac{\eta+\xi t}{\alpha'}+e^{\alpha' t+\eta t+\xi t^2}}\right)$
Proposed exponential learning model	$m_2(t)=a\left(1-\frac{1+\frac{\eta e^{\xi t}}{\alpha'}}{\frac{\eta e^{\xi t}}{\alpha'}+e^{(\alpha'+\eta e^{\xi t})t}}\right)$

4.2. Data analysis

The results of parameters estimated for the proposed models and other models with these four datasets presented as Table 4.3 and the results showed better fitting than other models (see Table 4.4).

Table 4.3. *Parameters estimated for different datasets and models*

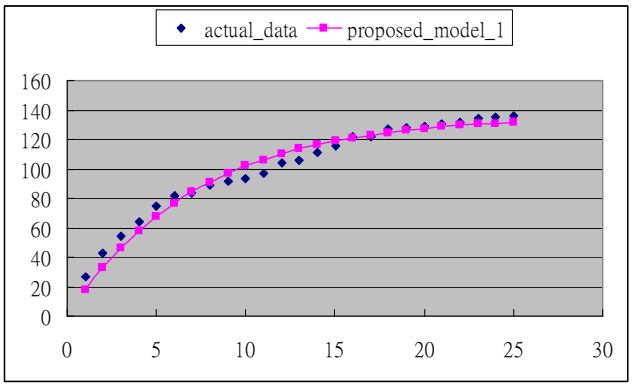
The Sources of Datasets	The comparison models	
	Pham and Zhang (2003)	**Huang (2005)**
Zhang and Pham (1998)	a =120.98 b =0.139304 c =14.883203 α =0.13856 β =0.01	a =145.67 κ =1.02074 r =0.987034 α =0.04297 A =0.163192 N =27.37
Pham (2003)	a =131.14 b =0.04507 c =1E-12 α =1E-12 β =1E-12	a =132.21 κ =1.055028 r =0.256651 α =0.015564 A =0.2127 N =88.23
Bai, Hu, Xie, and Ng (2005)	a =20.17 b =0.30462 c =0.000001 α =0.30462 β =0.000001	a =20.62 κ =1 r =3.44794 α =0.173252 A =0.029234 N =24.35
Jeske and Zhang (2005)	a =21.73 b =0.717 c =0.562066 α =0.178759 β =0.41241	a =23.45 κ =1.14 r =0.657484 α =0.540129 A =3.556583 N =6.37

The Sources of Datasets	The proposed models	
	Linear Learning model	Exponential Learning model
Zhang and Pham (1998)	a =135.97 α =0.138257 η =3.36E-10 ξ =0.00001	a =135.97 α =0.138257 η =3.36E-10 ξ =1
Pham (2003)	a =131.2 α =0.045059 η =1E-13 ξ =0.00001	a =131.2 α =0.045059 η =1E-13 ξ =0.35
Bai, Hu, Xie, and Ng (2005)	a =19.11 α =0.0001 η =0.35186 ξ =0.00001	a =19.11 α =0.0001 η =0.35186 ξ =0.03
Jeske and Zhang (2005)	a =22.25 α =0.4922166 η =0.333534 ξ =0.002	a =22.25 α =0.4922166 η =0.333534 ξ =0.007

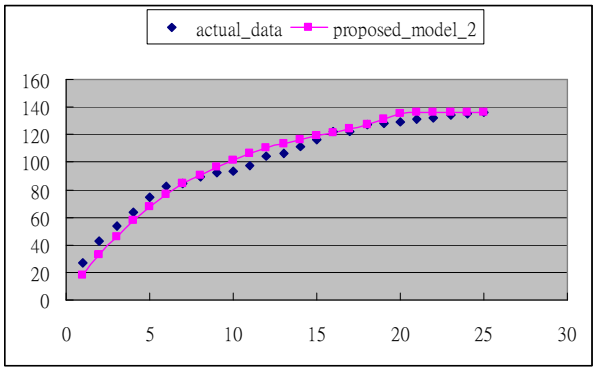
The *Rsq* increase to 0.975, 0.987, 0.976, and 0.989 with linear-learning effects as in Equation (3.14), and to 0.986, 0.989, 0.968, and 0.989 with the exponential-learning effects, as in Equation (3.15) (see Table 4.4). Figure 4.1 shows the predicted and actual data for cumulative errors, Table A.1~A.4 presented the actual data and the predicted values of the models. More detail data of model comparisons show in Appendixes A.

Table 4.4. *The models fitting results with the Rsq*

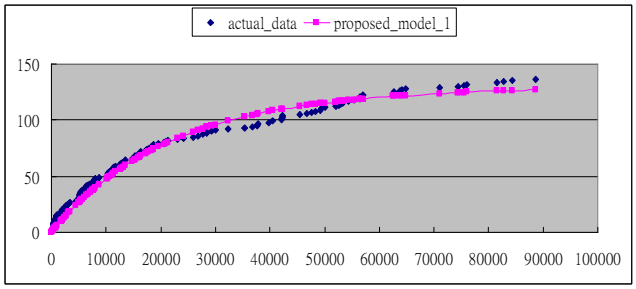
Model	The Sources of Datasets			
	Zhang and Pham (1998)	**Pham (2003)**	**Bai (2005)**	**Jeske and Zhang (2005)**
Pham and Zhang (2003)	0.966	0.975	0.914	0.988
Huang (2005)	0.973	0.982	0.953	0.988
Chiu (2008)	0.966	0.975	0.930	0.989
Proposed linear learning model	0.975	0.987	0.976	0.989
Proposed exponential learning model	0.986	0.989	0.968	0.989



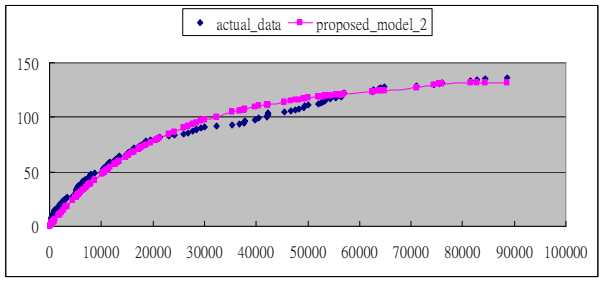
(1) The fitting results of proposed linear learning model for the dataset in Zhang and Pham (1998) (R^2=0.975)



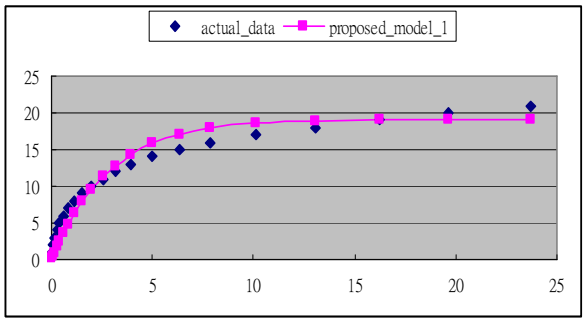
(2) The fitting results of proposed exponential learning model for the dataset in Zhang and Pham (1998) (R^2=0.986)



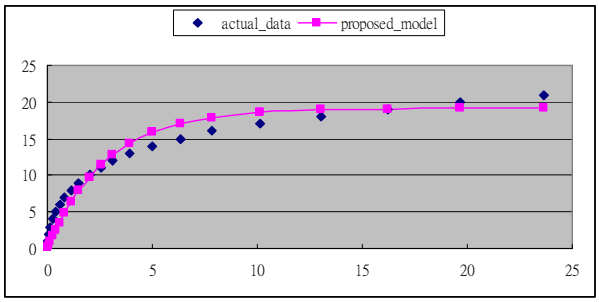
(3) The fitting results of proposed linear learning model for the dataset in Pham (2003) (R^2=0.987)



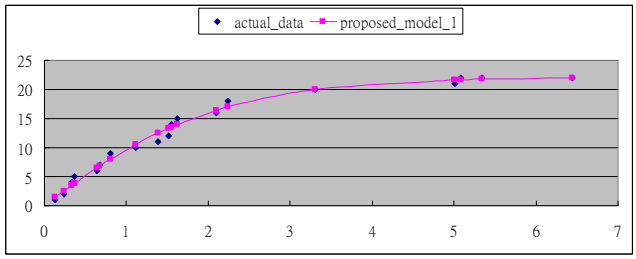
(4) The fitting results of proposed exponential learning model for the dataset in Pham (2003) (R^2=0.989)



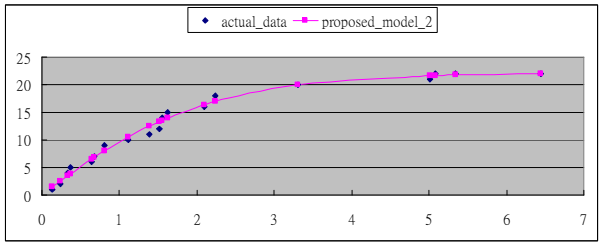
(5) The fitting results of proposed linear learning model for the dataset in Bai (2005) (R^2=0.976)



(6) The fitting results of proposed exponential learning model for the dataset in Bai (2005) (R^2=0.968)



(7) The fitting results of proposed linear learning model for the dataset in Jeske and Zhang (2005) (R^2=0.989)



(8) The fitting results of proposed exponential learning model for the dataset in Jeske and Zhang (2005) (R^2=0.989)

Figure 4.1. *Fitting results for different datasets*

5. Conclusion

This study improved Chiu's model (2008) by including time-varying learning effects, including linear and exponential functions to describe the time variation of learning effects, which described the software testing behavior more reasonably. This paper evaluated the effectiveness of the proposed models using various published datasets, and compared with other models by using the *Rsq* comparison criteria. The results enhanced with both the linear and exponential-shaped learning effects. These proposed models only added one parameter, the accelerative factor ξ, in the original model and enhanced the *Rsq* to fitting better for actual datasets, event the *Rsq* shows good outcome in the original model and shows better fitting than other models, and the proposed models can still present the suitable region of the original model when $\xi = 0$.

This research predicted errors occurred in software

testing process with two mean value functions, individually. We will refine the proposed models by taking account of the change-point problem, which is concerned with a change in the factors that affect software debugging, resulting in variations in the software error intensity function and the subsequent precision of the model prediction to consider two mean value functions, simultaneously. By model comparisons, the results show better fitting than those others, and the results are helpful for the managers managing the schedule of the software testing/debugging projects, the performance of the programmers, and the reliability of the software system.

Appendix

The results of model comparisons

Table A.1. *Model comparisons of partial Misra failure data (the dataset in Zhang and Pham (1998))*

Testing time (per hour)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
1	27	17.515178	18.753639	17.527226	17.527305	17.527226
2	43	32.789511	34.611824	32.795171	32.795691	32.795171
3	54	46.105543	48.073478	46.095057	46.096522	46.095057
4	64	57.711199	59.544218	57.680570	57.683470	57.680571
5	75	67.823776	69.354962	67.772696	67.777430	67.772700
6	82	76.633546	77.776637	76.563933	76.570776	76.563948
7	84	84.306973	85.031825	84.221968	84.231067	84.222016
8	89	90.989586	91.304011	90.892873	90.904255	90.893015
9	92	96.808530	96.744969	96.703889	96.717483	96.704300
10	93	101.874820	101.480660	101.765860	101.781512	101.767016
11	97	106.285370	105.615980	106.175330	106.192839	106.178534
12	104	110.124690	109.238530	110.016420	110.035532	110.025130
13	106	113.466510	112.421750	113.362390	113.382835	113.385768
14	111	116.375100	115.227370	116.277050	116.298563	116.339097
15	116	118.906460	117.707460	118.816010	118.838314	118.978846
16	122	121.109420	119.906050	121.027700	121.050527	121.449567
17	122	123.026490	121.860500	122.954300	122.977408	124.023841
18	127	124.694700	123.602620	124.632560	124.655724	127.209285
19	128	126.146320	125.159520	126.094480	126.117512	131.484050
20	129	127.409420	126.554440	127.367970	127.390682	135.255654
21	131	128.508460	127.807290	128.477290	128.499548	135.971130
22	132	129.464730	128.935240	129.443630	129.465296	135.974000
23	134	130.296760	129.953060	130.285400	130.306378	135.974000
24	135	131.020680	130.873580	131.018670	131.038874	135.974000
25	136	131.650520	131.707870	131.657420	131.676788	135.974000

Table A.2. *Model comparisons of the real-time control system failure data (the dataset in Pham (2003))*

Testing time (per 1000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
0.0030	1	0.017736	20.532290	0.017734	0.017734	0.020000
0.0330	2	0.194966	20.587930	0.194943	0.194942	0.190000
0.1460	3	0.860387	20.798600	0.860285	0.860280	0.860000
0.2270	4	1.335288	20.950680	1.335132	1.335122	1.340000
0.3420	5	2.006560	21.168110	2.006326	2.006305	2.010000
0.3510	6	2.058948	21.185200	2.058708	2.058686	2.060000
0.3530	7	2.070587	21.189000	2.070345	2.070323	2.070000
0.4440	8	2.599045	21.362460	2.598743	2.598706	2.600000
0.5560	9	3.246486	21.577500	3.246111	3.246047	3.250000
0.5710	10	3.332949	21.606420	3.332565	3.332496	3.330000
0.7090	11	4.125672	21.874010	4.125199	4.125079	4.130000
0.7590	12	4.411676	21.971610	4.411172	4.411028	4.410000
0.8360	13	4.850864	22.122570	4.850311	4.850124	4.850000
0.8600	14	4.987442	22.169790	4.986874	4.986672	4.990000
0.9680	15	5.600220	22.383260	5.599585	5.599306	5.600000
1.0560	16	6.097319	22.558380	6.096631	6.096276	6.100000
1.7260	17	9.818094	23.926760	9.817020	9.815605	9.820000
1.8460	18	10.472720	24.178420	10.471580	10.469867	10.470000
1.8720	19	10.614090	24.233210	10.612930	10.611153	10.610000
1.9860	20	11.231990	24.474550	11.230770	11.228666	11.230000
2.3110	21	12.976210	25.172530	12.974830	12.971595	12.970000
2.3660	22	13.268870	25.292100	13.267460	13.264002	13.270000
2.6080	23	14.547970	25.823260	14.546450	14.541899	14.550000
2.6760	24	14.904890	25.973980	14.903330	14.898440	14.900000
3.0980	25	17.095570	26.923790	17.093820	17.086445	17.090000
3.2780	26	18.017380	27.336480	18.015550	18.006919	18.020000
3.2880	27	18.068380	27.359540	18.066540	18.057834	18.070000
4.4340	28	23.762520	30.094030	23.760240	23.740384	23.760000
5.0340	29	26.628490	31.597540	26.626020	26.598015	26.630000
5.0490	30	26.699150	31.635750	26.696670	26.668446	26.700000
5.0850	31	26.868550	31.727580	26.866050	26.837283	26.870000
5.0890	32	26.887350	31.737800	26.884860	26.856025	26.880000
5.0890	33	26.887350	31.737800	26.884860	26.856025	26.880000
5.0970	34	26.924950	31.758230	26.922450	26.893499	26.920000
5.3240	35	27.986180	32.341620	27.983610	27.951070	27.980000
5.3890	36	28.288060	32.509930	28.285480	28.251860	28.290000
5.5650	37	29.101040	32.968480	29.098400	29.061780	29.100000
5.6230	38	29.367540	33.120480	29.364890	29.327243	29.360000
6.0800	39	31.443210	34.333490	31.440440	31.394149	31.440000
6.3800	40	32.782730	35.144360	32.779900	32.727366	32.780000
6.4770	41	33.211990	35.408970	33.209130	33.154485	33.210000
6.7400	42	34.366440	36.132330	34.363520	34.302917	34.360000
7.1920	43	36.318820	37.395260	36.315820	36.244158	36.320000
7.4470	44	37.402860	38.118510	37.399810	37.321459	37.400000

Testing time (per 1000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
7.6440	45	38.231840	38.682430	38.228760	38.145018	38.230000
7.8370	46	39.036880	39.239190	39.033770	38.944562	39.030000
7.8430	47	39.061800	39.256570	39.058680	38.969304	39.060000
7.9220	48	39.389220	39.485720	39.386090	39.294420	39.390000
8.7380	49	42.703860	41.892110	42.700630	42.583457	42.700000
10.0890	50	47.930660	46.019710	47.927330	47.760892	47.930000
10.2370	51	48.484140	46.481510	48.480800	48.308457	48.480000
10.2580	52	48.562380	46.547170	48.559040	48.385845	48.560000
10.4910	53	49.425480	47.278000	49.422130	49.239406	49.420000
10.6250	54	49.917770	47.700140	49.914410	49.726097	49.910000
10.9820	55	51.214880	48.830920	51.211520	51.007928	51.210000
11.1750	56	51.907480	49.445750	51.904120	51.692037	51.900000
11.4110	57	52.746250	50.200660	52.742880	52.520205	52.740000
11.4420	58	52.855760	50.300060	52.852400	52.628311	52.850000
11.8110	59	54.147660	51.487300	54.144300	53.903139	54.140000
12.5590	60	56.701470	53.913600	56.698130	56.420691	56.700000
12.5590	61	56.701470	53.913600	56.698130	56.420691	56.700000
12.7910	62	57.476220	54.670480	57.472900	57.183772	57.470000
13.1210	63	58.564380	55.749780	58.561080	58.254988	58.560000
13.4860	64	59.749250	56.946440	59.745970	59.420671	59.750000
14.7080	65	63.577210	60.961880	63.574040	63.181228	63.570000
15.2510	66	65.211680	62.743390	65.208580	64.784319	65.210000
15.2610	67	65.241410	62.776140	65.238300	64.813460	65.240000
15.2770	68	65.288950	62.828550	65.285840	64.860059	65.290000
15.8060	69	66.841450	64.557480	66.838420	66.381192	66.840000
16.1850	70	67.931200	65.790510	67.928220	67.448034	67.930000
16.2290	71	68.056510	65.933300	68.053530	67.570665	68.050000
16.3580	72	68.422470	66.351420	68.419520	67.928743	68.420000
17.1680	73	70.672390	68.957900	70.669570	70.128328	70.670000
17.4580	74	71.458160	69.881820	71.455380	70.895756	71.460000
17.7580	75	72.260290	70.831660	72.257560	71.678750	72.260000
18.2870	76	73.648550	72.490310	73.645920	73.032909	73.650000
18.5680	77	74.372630	73.362180	74.370060	73.738701	74.370000
18.7280	78	74.780840	73.855570	74.778300	74.136448	74.780000
19.5560	79	76.846930	76.370680	76.844550	76.147877	76.840000
20.5670	80	79.267310	79.345450	79.265150	78.500589	79.260000
21.0120	81	80.298200	80.617880	80.296140	79.501457	80.300000
21.3080	82	80.972550	81.450970	80.970570	80.155783	80.970000
23.0630	83	84.791210	86.157540	84.789660	83.855222	84.790000
24.1270	84	86.963490	88.805080	86.962220	85.955305	86.960000
25.9100	85	90.378070	92.875650	90.377290	89.250153	90.380000
26.7700	86	91.929480	94.673240	91.928950	90.744732	91.930000
27.7530	87	93.630680	96.596960	93.630430	92.381941	93.630000
28.4600	88	94.808460	97.895650	94.808410	93.514447	94.810000
28.4930	89	94.862520	97.954560	94.862480	93.566413	94.860000

Testing time (per 1000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
29.3610	90	96.256040	99.450350	96.256260	94.905360	96.260000
30.0850	91	97.377400	100.620500	97.377830	95.982089	97.380000
32.4080	92	100.738000	103.927200	100.739100	99.205639	100.740000
35.3380	93	104.504500	107.230900	104.506500	102.814533	104.510000
36.7990	94	106.204700	108.568900	106.207200	104.443146	106.210000
37.6420	95	107.136000	109.259300	107.138700	105.335318	107.140000
37.6540	96	107.149000	109.268800	107.151700	105.347776	107.150000
37.9150	97	107.430100	109.471000	107.432900	105.617088	107.430000
39.7150	98	109.281100	110.732600	109.284500	107.391262	109.280000
40.5800	99	110.118600	111.262800	110.122200	108.194536	110.120000
42.0150	100	111.438000	112.046400	111.442000	109.460999	111.440000
42.0450	101	111.464700	112.061600	111.468700	109.486627	111.470000
42.1880	102	111.591300	112.133400	111.595400	109.608322	111.600000
42.2960	103	111.686500	112.187000	111.690600	109.699723	111.690000
42.2960	104	111.686500	112.187000	111.690600	109.699723	111.690000
45.4060	105	114.236600	113.499000	114.241600	112.153850	114.240000
46.6530	106	115.162900	113.917400	115.168200	113.047836	115.170000
47.5960	107	115.829600	114.199500	115.835100	113.692421	115.840000
48.2960	108	116.306500	114.391600	116.312200	114.154161	116.310000
49.1710	109	116.881800	114.612800	116.887700	114.712049	116.890000
49.4160	110	117.038900	114.671100	117.044900	114.864524	117.050000
50.1450	111	117.496100	114.836100	117.502300	115.308833	117.500000
52.0420	112	118.617800	115.210400	118.624500	116.402030	118.630000
52.4890	113	118.868500	115.288200	118.875200	116.646996	118.880000
52.8750	114	119.080900	115.352500	119.087800	116.854817	119.090000
53.3210	115	119.321800	115.423600	119.328700	117.090739	119.330000
53.4430	116	119.386800	115.442400	119.393800	117.154501	119.400000
54.4330	117	119.901700	115.586800	119.908900	117.659911	119.920000
55.3810	118	120.373600	115.711500	120.381100	118.124480	120.390000
56.4630	119	120.888200	115.839200	120.895900	118.632572	120.910000
56.4850	120	120.898400	115.841700	120.906100	118.642666	120.920000
56.5600	121	120.933100	115.850000	120.940800	118.677006	120.960000
57.0420	122	121.153400	115.901800	121.161200	118.895147	121.180000
62.5510	123	123.359000	116.338000	123.368000	121.101578	123.470000
62.6510	124	123.394200	116.343700	123.403200	121.137185	123.510000
62.6610	125	123.397800	116.344300	123.406700	121.140737	123.510000
63.7320	126	123.764700	116.402400	123.773800	121.512953	123.930000
64.1030	127	123.887800	116.421100	123.897000	121.638151	124.070000
64.8930	128	124.143000	116.458400	124.152300	121.898543	124.380000
71.0430	129	125.847800	116.662100	125.858100	123.666055	127.320000
74.3640	130	126.589700	116.727400	126.600500	124.455703	129.710000
75.4090	131	126.801100	116.743600	126.812100	124.683715	130.390000
76.0570	132	126.927300	116.752700	126.938400	124.820530	130.720000
81.5420	133	127.859900	116.809300	127.871600	125.851762	131.200000
82.7020	134	128.029200	116.817600	128.041100	126.043632	131.200000

Testing time (per 1000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
84.4000	135	128.283500	116.828900	128.295500	126.309827	131.200000
88.6820	136	128.774800	116.847000	128.787200	126.910967	131.200000

Table A.3. *Model comparisons of Space program failure data (the dataset in Bai (2005))*

Testing time (per 100 hours)	***Defects found***	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
0.020	1	0.020000	0.020000	0.020000	0.134009	0.134009
0.060	2	0.060000	0.060000	0.060000	0.399217	0.399217
0.150	3	0.150000	0.150000	0.150000	0.982463	0.982463
0.270	4	0.270000	0.270000	0.270000	1.731950	1.731949
0.410	5	0.410000	0.410000	0.410000	2.567288	2.567286
0.580	6	0.580000	0.580000	0.580000	3.527846	3.527841
0.830	7	0.830000	0.830000	0.830000	4.840065	4.840052
1.160	8	1.160000	1.160000	1.160000	6.404571	6.404540
1.520	9	1.520000	1.520000	1.520000	7.916391	7.916333
2.000	10	2.000000	2.000000	2.000000	9.656157	9.656050
2.560	11	2.560000	2.560000	2.560000	11.347247	11.347074
3.140	12	3.140000	3.140000	3.140000	12.780608	12.780362
3.900	13	3.900000	3.900000	3.900000	14.266173	14.265837
4.980	14	4.980000	4.980000	4.980000	15.798014	15.797580
6.330	15	6.330000	6.330000	6.330000	17.050816	17.050322
7.850	16	7.850000	7.850000	7.850000	17.904163	17.903672
10.150	17	10.150000	10.150000	10.150000	18.573490	18.573088
13.040	18	13.040000	13.040000	13.040000	18.916101	18.915842
16.240	19	16.240000	16.240000	16.240000	19.047183	19.047047
19.650	20	19.650000	19.650000	19.650000	19.091105	19.091042
23.680	21	23.680000	23.680000	23.680000	19.105433	19.105410

Table A.4. *Model comparisons of failure data on wireless data service system (the dataset in Jeske and Zhang (2005))*

Testing time (per 10000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
0.1340	1	1.521959	3.772587	1.454990	1.455015	1.455020
0.2350	2	2.625657	4.238855	2.522079	2.522206	2.522228
0.3360	4	3.690907	4.747994	3.562210	3.562564	3.562624
0.3690	5	4.030500	4.923709	3.897220	3.897680	3.897758
0.6380	6	6.640963	6.524091	6.512037	6.514079	6.514426
0.6720	7	6.950800	6.746603	6.822095	6.824428	6.824825
0.8060	9	8.128076	7.662513	8.024262	8.027967	8.028599
1.1090	10	10.536260	9.907183	10.501810	10.509663	10.511013
1.3810	11	12.410300	11.997960	12.439310	12.451716	12.453862
1.5160	12	13.245140	13.010750	13.309990	13.324786	13.327354
1.5500	14	13.445870	13.259970	13.517680	13.533075	13.535749
1.6180	15	13.836150	13.749470	13.921290	13.937882	13.940769

Testing time (per 10000 hours)	*Defects found*	Total defects predicted by the following models				
		Pham and Zhang (2003)	**Huang (2005)**	**Chiu (2008)**	**Proposed linear learning model**	**Proposed exponential learning model**
2.0930	16	16.175690	16.714350	16.331870	16.356094	16.360353
2.2350	18	16.755810	17.418800	16.925040	16.951144	16.955746
3.3050	20	19.782540	20.380550	19.917420	19.949350	19.955094
5.0120	21	21.711430	21.294340	21.662110	21.684046	21.688088
5.0850	22	21.751920	21.303590	21.696310	21.717669	21.721608
5.3410	22	21.877470	21.329520	21.801240	21.820601	21.824182
6.4390	22	22.204640	21.376840	22.068540	22.080424	22.082643

References

[1] Achcar, J.A., Dey, D.K., and Niverthi, M. (1997) "A Bayesian approach using nonhomogeneous Poisson Process for software reliability models", *In Frontiers in Reliability*, Basu et al. (Eds).

[2] Adam Smiarowski , Jr. , Hoda S. Abdel-Aty-Zohdy , Mostafa Hashem Sherif , Hemal Shah (2006) "Wavelet Based RDNN for Software Reliability Estimation", *11th IEEE Symposium on Computers and Communications* (ISCC'06), iscc,pp.312-317.

[3] Bai, C.G. (2005) "Bayesian network based software reliability prediction with an operational profile", *The Journal of Systems and Software*, 77: 103-112.

[4] Bai, C.G., Hu, Q.P., Xie, M., and Ng, S.H. (2005) "Software failure prediction based on a Markov Bayesian network model", *The Journal of Systems and Software,* 74: 275-282.

[5] Bunea, C., Charitosb, T., Cooke, R. M., and Beckerd, G. (2005) "Two-stage Bayesian models—application to ZEDB project", *Reliability Engineering and System Safety*, 90: 123-130.

[6] Chin-Yu Huang, Sy-Yen Kuo, Michael R. Lyu, (2000) "Effort-Index-Based Software Reliability Growth Models and Performance Assessment," *Computer Software and Applications Conference, Annual International*, The Twenty-Fourth Annual International Computer Software and Applications Conference, 0:454.

[7] Chiu, K.-C., Huang, Y.-S., and Lee, T.-Z. (2008) "A Study of Software Reliability Growth from the Perspective of Learning Effects," Reliability *Engineering and Systems Safety*, Vol. 93, No. 10, pp. 1410-1421.

[8] Chiu, K.-C., Ho, J.-W., and Huang, Y.-S. (2009) "Bayesian Updating of Optimal Release Time for Software Systems," *Software Quality Journal*, Vol. 17, No. 1, pp. 99-120.

[9] Cid, J.E.R. and Achcar, J.A. (1999) "Bayesian inference for nonhomogeneous Poisson processes in software reliability models assuming nonmonotonic intensity functions", *Computational Statistics & Data Analysis*, 32: 147-159.

[10] Dietmar Pfahl (2001) "An Integrated Approach to Simulation-Based Learning in Support of Strategic and Project Management in Software Organisations*" , PhD Theses in Experimental Software Engineering,* Fraunhofer-Institut für Experimentelles Software Engineering, 8:27-40.

[11] Goel, A.L. and Okumoto, K. (1979) "Time-varying fault detection rate model for software and other performance measures", *IEEE Transactions on Reliability*, 28: 206-211.

[12] Gokhale, S. S. and Trivedi, K. S. (1999) "A time/structure based software reliability model", *Annals of Software Engineering*, 8: 85-121.

[13] Ho, J.W., Fang, C.C. and Huang, Y.S. (2008) "The Determination of Optimal Software Release Times at Different Confidence Levels with Consideration of Learning Effects," *Software Testing, Verification and Reliability*, 18(4): 221-249. (SCI)

[14] Hossain, S.A. and Dahiya, R.C. (1993) "Estimating the Parameters of a Non-homogeneous Poisson-Process Model for Software Reliability", *IEEE Transactions on Reliability*, 42: 604-612.

[15] Hu, Q.P., Xie, M., Ng, S.H. and Levitin, G. (2007) "Robust recurrent neural network modeling for software fault detection and correction prediction", *Reliability Engineering & System Safety*, 92: 332-340.

[16] Huan-Jyh Shyur (2003),"A stochastic software reliability model with imperfect-debugging and change-point",*The Journal of Systems and Software*, 66,p.135–141

[17] Huang, C.-Y. (2005) "Performance analysis of software reliability growth models with testing-effort and change-point", *Journal of Systems and Software*, 76: 181-194.

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

[19] Jing Zhao, Hong-Wei Liu, Gang Cui and Xiao-Zong Yang (2006),"Software reliability growth model with change-point and environmental function". *Journal of Systems and Software,* Volume 79, Issue 11, November ,P. 1578-1587

[20] Kapur, P.K. and Bhalla, V.K. (1992) "Optimal release policies for a flexible software reliability growth model", *Reliability Engineering & System Safety*, 35: 49-54.

[21] Karatsas, I., Shreve, S. (1997) *Brownian Motion and Stochastic Calculus*, 2nd ed. Springer-Verlag: New York.

[22] Katrina Maxwell, Luk Van Wassenhove, and Soumitra Dutta (1999) "Performance Evaluation of General and Company Specific Models in Software Development Effort Estimation", *Management Science ,* 45: 787-803.

[23] Kimura, M., Toyota, T. and Yamada, S. (1999) "Economic

analysis of software release problems with warranty cost and reliability requirement", *Reliability Engineering & System Safety*, 66: 49-55.

[24] Kuo, L., Lee, J.C., Choi, K., and Yang, T.Y. (1997) "Bayes inference for S-shaped software reliability growth models", *IEEE Transactions on Reliability*, 46: 76-80.

[25] Kuo, L. and Yang, T.Y. (1996) "Bayesian computation for nonhomogeneous Poisson processes in software reliability", *Journal of the American Statistical Association*, 91: 763-773.

[26] Lee, C.H., Kim, Y.T., Park, D.H. (2004) "S-shaped software reliability growth models derived from stochastic differential equations", *IIE Transactions*, 36: 1193-1199.

[27] Littlewood, B. (2006) "Comments on 'Evolutionary neural network modeling for software cumulative failure time prediction", *Reliability Engineering & System Safety*, 91: 485-486.

[28] Melo, A.C.V. and Sanchez, A.J. (2008) "Software maintenance project delays prediction using Bayesian networks", *Expert Systems with Applications*, 34: 908-919.

[29] Moran, P.A.P. (1969) "Statistical inference with bivariate gamma distribution", *Biometrika*, 56: 627-634.

[30] Nalina Suresh, A.N.V. Rao, A.J.G. Babu (1996) "A software reliability growth model", *International Journal of Quality & Reliability Management*, 13:84-94.

[31] Ohba, M. (1984a) "Inflexion S-shaped software reliability growth models", in *Stochastic Models in Reliability Theory*, Osaki, S. and Hatoyama, Y., Eds. Berlin, Germany: Springer-Verlag, 144-162.

[32] Ohba, M. (1984b) "Software reliability analysis models", *IBM Journal of Research and Development*, 28: 428-443.

[33] Özekici, S. and Soyer, R. (2003) "Reliability of software with an operational profile", *European Journal of Operational Research*, 149: 459-474.

[34] P.K. Kapur, D.N. Goswami, and A. Bardhan (2007) "A General Software Reliability Growth Model with Testing Effort Dependent Learning Process", *International Journal of Modeling and Simulation*, 205:4401

[35] Pham, H. and Zhang, X. (1999) "A software cost model with warranty and risk costs", *IEEE Transactions on Computers*, 48: 71-75.

[36] Pham, H. and Zhang, X. (2003) "NHPP software reliability and cost models with testing coverage", *European Journal of Operational Research*, 145: 445-454.

[37] Pham, H. (2003) "Software reliability and cost models - Perspectives, comparison, and practice", *European Journal of Operational Research*, 149: 475-489.

[38] Shyur, H.-J. (2003) "A stochastic software reliability model with imperfect debugging and change-point", *The Journal of Systems and Software*, 66: 135-141.

[39] T. P. Wright. (1936) "Factors Affecting the Cost of Airplanes", *Journal of the Aeronautical Sciences*, February:3

[40] Tamura, Y., Yamada, S. (2006) "A flexible stochastic differential equation model in distributed development environment", *European Journal of Operational Research*, 168: 143-152.

[41] Tian, L. and Noore, A. (2005) "Evolutionary neural network modeling for software cumulative failure time prediction", *Reliability Engineering & System Safety*, 87: 45-51.

[42] William J. Stevenson. (1999) "Production Operations Management", Irwin/McGraw-Hill, 349-358.

[43] Yamada, M. Kimura, H. Tanaka and S. Osaki. (1994) "Software reliability measurement and assessment with stochastic differential equations", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E77-A: 109-116.

[44] Yamada, S., Ohba, M., and Osaki, S.(1983). "S-shaped software reliability modeling for software error detection". *IEEE Transactions on Reliability*, 32: 475-484.

[45] Yamada, S., Osaki. S. (1985) "Software reliability growth modeling: Models and applications", *IEEE Transactions on Software Engineering*, 11: 1431-1437.

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

[47] Yin, L., Trivedi, K.S. (1999) "Confidence Interval Estimation of NHPP-Based Software Reliability Models", *Proceedings of the 10th International Symposium on Software Reliability Engineering*, November: 6-11.

[48] Zhang, X. and Pham, H. (1998) "A software cost model with warranty cost, error removal times and risk costs", *IIE Transactions*, 30: 1135-1142.

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

[50] Zhao, M. (1993) "Change-point problems in software and hardware reliability", *Communications in Statistics Theory and Methods*, 22: 757-768.