Journal of Data Science 857-878 , DOI: 10.6339/JDS.201810_16(4).00010

ANALYSIS OF DELAYED S SHAPED SOFTWARE RELIABILITY GROWTH MODEL WITH TIME DEPENDENT FAULT CONTENT RATE FUNCTION

David D. Hanagal[*] , Nileema N. Bhalerao[2]

Department of Statistics, Savitribai Phule Pune University

Pune – 411007, India

ABSTRACT

Many software reliability growth models based upon a non-homogeneous Poisson process (NHPP) have been proposed to measure and asses the reliability of a software system quantitatively. Generally, the error detection rate and the fault content function during software testing is considered to be dependent on the elapsed time testing. In this paper we have proposed three software reliability growth models (SRGM's) incorporating the notion of error generation over the time as an extension of the delayed S-shaped software reliability growth model based on a non-homogeneous Poisson process (NHPP). The model parameters are estimated using the maximum likelihood method for interval domain data and three data sets are provided to illustrate the estimation technique. The proposed model is compared with the existing delayed S-shaped model based on error sum of squares, mean sum of squares, predictive ratio risk and Akaike's information criteria using three different data sets. We show that the proposed models perform satisfactory better than the existing models.

Keywords: Error detection rate; Fault Content Rate Function; Non-homogeneous Poisson process; Predictive ratio risk.

* Corresponding author: David D. Hanagal
Email:david−hanagal@yahoo.co.in

1. Introduction

Over the past four decades several software models (SRGM) have been proposed by various researchers [See Pham (1993, Teng and Pham (2006), Pham (1996), Obha and Yamada (1984), Teng and Pham (2004), Zhang et al. (2003)]. The pioneering attempt in the non-homogeneous Poisson process (NHPP) software reliability growth model was by Goel and Okumoto (1979) (GO). The model describes the failure observation phenomenon by an exponential curve. There are other SRGM's describing either S-shaped curves or a mixture of exponential and S-shaped curves (flexible). Some of the important models falling under this category are due to Yamada et al. (1983), Obha (1984), Bittanti et al. (1988), Kapur and Garg (1992), Kapur et al. (1999), Pham (2006), Chen (2010) and Lai and Garg (2012).

These models were developed based on the assumption that faults detected in the testing phase are removed immediately with no debugging time delay and no new faults are introduced into the software. In other words, it is assumed that whenever an attempt is made to remove a fault, it is removed with certainty and this is referred as perfect because of a number of factors like tester's skill and expertise. The testing is observed and the original fault may remain, leading to a phenomenon known as imperfect debugging. Another possibility is that while correcting a software error additional errors may be generated and these errors may get into software. Such models may be referred as error generation models. In the case of error generation, the total fault content increases as testing progresses because new faults are introduced into the system while removing the original faults.

Goel and Okumoto (1979) have introduced the concept of imperfect debugging for the first time in literature. Obha and Chon (1989) extended this work by introducing the error generation into the GO model and named it as imperfect debugging model. Kapur and Grag (1990) introduced the imperfect debugging in GO model and they considered fault introduction rate per remaining faults being reduced due to the imperfect debugging. Thus the number of failure observed and detected by infinite time is more than the initial fault content. Yamada et al. (1992) extended GO model by assuming that fault introduction rate is linear and exponential time dependent. Further several authors have used this approach to model error generation activity in software reliability model. Pham and Zhang (1997) have developed a software reliability model considering error introduction rate as exponential and error introduction rate as non-decreasing type. Pham et al. (1999) have discussed linear fault introduction rate function and fault introduction is of non-decreasing time dependent type. Tokuno and Yamada (2000) proposed imperfect debugging SRGM with two types of hazard rates for software reliability measurement and assessment. Zhang and Pham (2000) developed NHPP model with imperfect debugging and time dependent fault-detection rate. Zhang et al. (2003) proposed SRGM model to integrate fault removal efficiency, failure rate, and fault introduction rate into software reliability assessment. Chatterjee and Shukla (2016) proposed a change point-based SRGM under imperfect debugging with revised concept of fault dependency. The above discussed models does not always reflect real testing environments. Recently Hanagal and Bhalerao (2016a, 2016b, 2017) obtained software reliability growth models based on inverse Weibull, generalized inverse Weibull, extended inverse Weibull distributions. It is of interest to construct a SRM by considering introduction of error generation during testing and central processing unit (CPU) execution time.

The remainder of the paper is organized as follows. In Section 2, we describe NHPP delayed S-shaped models. In section 3, we discuss the proposed models with three different fault content functions and software reliability assessment measures and the estimation of unknown model parameters using maximum likelihood method. Parameter estimation of is of primary importance in software reliability prediction. Once the analytical solution for $m(t)$ is known for a given model, the parameters in the solution need to be determined. Parameter estimation is achieved by applying a technique of maximum likelihood estimate (MLE), for the interval domain area data. In many cases, the maximum likelihood estimators are consistent and asymptotically normally distributed as the sample size increases; see Zhao and Xie (1996). In Section 4, we represent the analysis of three data sets DS I, DS II and DS III. Section 5 contains the major conclusions and remarks of the study.

2. NHPP Delayed S-Shaped Model

In the NHPP S-shaped mode, the software reliability growth curve is an S-shaped curve. The detection rate of faults, where the error detection rate changes with time, because the greatest at a certain time after testing begins, after which it decreases exponentially. In other words, some faults are covered by other faults at the beginning of the testing phase, before these faults are actually removed, the covered faults remain undetected. Yamada et al. (1984) also determined that the software testing process usually involves a learning process where testers becomes familiar with the software products, environments, and software specifications. Several S-shaped models (Yamada et al. 1984; Pham 1997a) such as delayed S-shaped, inflection S-shaped, etc., exists in literature. Kareer et al. (1990) proposed an S-shaped SRGM with two types of errors. The errors have been classified depending upon their survey. Gupta et al. (2011) obtained software reliability estimation using delayed S-shaped model under imperfect debugging and time lag. Ahmed et al. (2011) developed an inflation SRGM considering log-logistic testing-effort and imperfect debugging. Kaur and Sharma (2015) predict the time between failure and accuracy by using CASRE tool for SRG models. Bokhari et al. (2017) proposed delayed S-shaped SRGM with imperfect debugging and new modified Weibull testing effort function.

We now discuss a stochastic model for a software error detection process based on NHPP in which the growth curve of the number of detected software errors for the observed failure data is S-shaped, called delayed S-shaped NHPP model (Yamada et al 1984). The software error detection procedure described by an S-shaped curve can be characterized as a learning process in which the test team members become familiar with the test environment, testing tools or project requirements, i.e., their test skills gradually improve. The delayed S-shaped model is based on the following assumptions:

1. All the faults in the program are mutually independent from the failure detection point of view.

2. The probability of failure detection at any time is proportional to the current number of faults in a software.
3. The proportionality of failure detection in constant.
4. The initial error content of the software is a random variable.
5. A software system is subject to failures at random times caused by errors present in the system.
6. The time between $(i-1)^{th}$ and i^{th} failures depends on the time to the $(i-1)^{th}$ failure.
7. Each time a failure occurs, the error which caused it is simultaneously removed and no other errors are introduced.

On the basis of these assumptions we have the following differential equation:

$$\frac{\partial m(t)}{\partial t} = b(t)[a - m(t)]. \quad a > 0, \qquad b(t) > 0, \; t \geq 0 \tag{2.1}$$

a=expected total number of faults that exists in the software before testing.
$b(t)$ = failure detection rate per fault, which also represents the average failure rate of a fault.

In delayed S-shaped NHPP model, $b(t) = \frac{b^2 t}{1+bt}$,where b is the error detection rate per error in the steady state. Solving the differential equation given in (2.1) with respect to $m(t)$ under the initial condition $m(0) = 0$. Yamada et al. (1984) obtained the mean value function given by:

$$m(t) = a[1 - (1 + bt)e^{-bt}] \tag{2.2}$$

which shows an S-shaped curve. This model is called delayed S-shaped NHPP model. Further, the counting process $N(t),\ t \geq 0$ representing the cumulative number of software faults detected up to testing time t is a stochastic process. Basic assumption about this process lead to the commonly accepted conclusion that, for any $t \geq 0,\ N(t)$ is a Poisson distributed with time dependent Poisson parameter $m(t)$, the mean value function (MVF).

$$P\{N(t) = n\} = \frac{[m(t)]^n}{n!} e^{-m(t)}, \quad n = 0,1,2,\dots,\infty \tag{2.3}$$

where $m(t)$ is given by :

$$m(t) = \int_0^t \lambda(x)dx \tag{2.4}$$

The MVF represents the expected number of software errors that have accumulated up to time t, or estimated cumulative faults up to time t.

The intensity function, $\lambda(t)$ of the NHPP given by:

$$\lambda(t) = ab^2 te^{-bt} \tag{2.5}$$

This represents the number of faults represented per unit testing time.

In general the second assumption (2) i.e. the detected faults are removed with certainty is not always true [See Yamada et al. (1992), Pham and Zhang (1997) and Pham (1999)]. If additional introduced faults affect the fraction of the fault content, we have to modify this assumption by introducing the concept of error generation while debugging (or testing) process is an action. That is, total fault content rate function, $a(t)$ representing the sum of expected number of initial software faults and introduced faults by time t. Replacing a by $a(t)$ in equation (2.1), we have differential equation

$$\frac{\partial m(t)}{\partial t} = b(t)[a(t) - m(t)]. \qquad a(t) > 0, \qquad b(t) > 0, \quad t \geq 0 \tag{2.6}$$

The above differential equation was given by Pham (2007).

The generalized mean value function (MVF) solution of the above differential equation (2.6) given by (Pham and Zhang 1997) is as follows:

$$m(t) = e^{-B(t)}[m_0 + \int_0^t a(\tau)b(\tau)e^{B(\tau)}\ d\tau] \tag{2.7}$$

where $B(t) = \int_{t_0}^{t} b(\tau)d\tau$ with initial condition $m(t_0) = m_0$ and t_0 is the time to begin the debugging process. Pham (2007) obtained NHPP SRGM with $a(t) = \alpha(1+\gamma t)^2$ and $b(t) = \frac{\gamma^2 t}{(1+\gamma t)}$ in this case both $a(t)$ and $b(t)$ have common parameter γ. In our proposed models we consider the parameters are independent instead of dependent case.

Many existing software reliability growth models [Pham (2006)], Pham and Zhang (2003), Yamada and Osaki (1985) can be considered as a special case of the above general model. An increasing function $a(t)$ implies an increasing total number of faults and reflects imperfect debugging. An increasing $b(t)$ implies an increasing fault detection rate, which could be either attributed to imperfect debugging, or to software process fluctuations, or a combination of both. Different $a(t)$ and $b(t)$ functions also reflect different assumptions of the software testing processes.

In the simplest model, the function $a(t)$ and $b(t)$ are both constants. A constant $a(t) = a$ stands for the assumption that no new errors are introduced during the debugging process. A constant $b(t) = b$ implies that the proportional factor relating the error detection rate $\lambda(t)$ to the total number of remaining errors is constant. In a general model, the functions $a(t)$ and $b(t)$ are both functions of time.

In the following section we propose general models with three different types of fault content function $a(t)$ and discuss the methods of estimating parameters of the models.

3. Proposed Models with Three Different Fault Content Rate Functions in Delayed S-shaped NHPP

3.1.1Linear Fault Content Rate Function

Here we assume that fault introduction rate is a linear function and dependent on time,

$$a(t) = a_1(t) = a(1 + \alpha t) \quad a > 0, \alpha > 0, \quad t \geq 0$$

where a is the number of initial fault content in the system and α is an increasing rate of the number of introduced faults to the initial fault content function. Let $m_1(t)$ be the MVF when fault content function $a_1(t)$ is substituted for $a(t)$ in (2.6) thus we have:

$$\frac{\partial m_1(t)}{\partial t} = b(t)[a(1 + \alpha t) - m_1(t)]. \quad a > 0, \alpha > 0, b(t) > 0 \quad t \geq 0 \tag{3.1}$$

with $b(t) = \frac{b^2 t}{1+bt}$. Solving the differential equation given in (3.1) with respect to $m_1(t)$ under the initial condtition $m_1(0) = 0$ we derived the mean value function given by:

$$m_1(t) = \left(-a - \frac{2a\alpha}{b}\right) e^{-bt}(1 + bt) + \frac{a\alpha}{b}(1 + bt) + \left(a - \frac{2a\alpha}{b}\right) + \frac{3a\alpha}{b(1 + bt)} \tag{3.2}$$

In this paper we call NHPP model with MVF $m\_1(t)$ given in (3.2) as delayed S-shaped linear model. The intensity function of this model is:

$$\lambda_1(t) = \left(a + \frac{2a\alpha}{b}\right) b^2 t e^{-bt} - \frac{3a\alpha}{(1 + bt)^2} + a\alpha \tag{3.3}$$

Software Reliability Assessment Measures

Based on delayed S-shaped linear model, we can derive the following quantitative measures useful for software reliability assessment.

(a) Software Reliability

Let $S_i, (i = 1, 2, 3 \dots)$ be a random variable representing the i^{th} software failure occurrence time. Define $X_i = S_i - S_{i-1}, (i = 1,2,3 \dots)$; $S_0 = 0$, is a random variable representing the time interval between $(i-1)^{th}$ and i^{th} software failure occurrence. The conditional probability that the i^{th} software failure does not occur between $(t, t + x]$, $(x \geq 0)$ on the condition that the $(i-1)^{th}$ software failure has occurred at testing time t, is given by:

$$R(x|t) = P\{X_i > x | S_{i-1} = t\} = \exp[-\{m(x + t) - m(t)\}], \quad t \geq 0, x \geq 0. \tag{3.4}$$

substituting (3.2) in (3.4) we have delayed S-shaped linear software reliability

$$R_1(x|t) = exp[-\left\{\left(a + \frac{2a\alpha}{b}\right) e^{-bt} \left\{(1 + bt) - (1 + b(t + x)e^{-bx}\right\} + \frac{a\alpha}{b}(1 + bx) - \frac{3ax\alpha}{(1 + bt)\big(1 + b(t + x)\big)}\right\}\Bigg], x, t \geq 0 \tag{3.5}$$

(b) Expected Residual Fault Content

Consider the expectation of $\{a(t) - N(t)\}$ for the NHPP model in (3.2). Since $N(t)$ is a random variable, the number of residual fault content at testing time t is:

$$n(t) = a(t) - E[N(t)] = a(t) - m(t) \tag{3.6}$$

Let $n_1(t)$ denote the expected residual fault content for delayed S-shaped linear SRM, that is

$$n_1(t) = a_1(t) - m_1(t) \tag{3.7}$$

$$n_1(t) = a(1+\alpha t) - [\left(-a - \frac{2a\alpha}{b}\right) e^{-bt}(1+bt) + \frac{a\alpha}{b}(1+bt)$$
$$+(a - \frac{2a\alpha}{b} + \frac{3a\alpha}{b(1+bt)})]. \tag{3.8}$$

3.1.1Parameter Estimation

In this section we discuss the method of estimating the parameters of the models mentioned above

Estimation using interval Domain Data

Suppose that n pairs of observations $(t_i, y_\mathrm{i}); (i = 1,2,3, \dots, n; 0 < t_1 < t_2 < \cdots < t_n)$ are made during the testing phase where y_i denote the number of software faults detected up to the testing time $t_i, i = 1,2, \dots, n$. The corresponding probability mass function is $P[N(t_1) = y_1, N(t_2) = y_2, \dots, N(t_n) = y_n]$ and hence the likelihood function for the interval domain is given by

$$L = \prod_{i=1}^{n} \frac{\left(m(t_i) - m(t_{i-1})\right)^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} \exp[-m(t_i) - m(t_{i-1})] \tag{3.9}$$

where $t_0 = 0$ and $y_0 = 0$. Taking the natural logarithm of equation (3.9) yields log-likelihood function, that is

$$\mathrm{lnL} = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln[m(t_i) - m(t_{i-1})] - m(t_n) - \sum_{i=1}^{n} \ln[(y_i - y_{i-1})!] \tag{3.10}$$

Substituting mean value function $m_1(t)$ into the equation (3.10), we get logarithmic of likelihood function excluding the constant terms,

$$\begin{aligned} lnL = \sum_{i=1}^{n} (y_i - y_{i-1})\, ln \Bigg[& \left(-a - \frac{2a\alpha}{b}\right) e^{-bt_i}(1+bt_i) + \frac{a\alpha}{b}(1+bt_i) \\ & + \left(a - \frac{2a\alpha}{b}\right) + \frac{3a\alpha}{b(1+bt_i)} \\ & - \left(-a - \frac{2a\alpha}{b}\right) e^{-bt_{i-1}}(1+bt_{i-1}) + \frac{a\alpha}{b}(1+bt_{i-1}) \\ & + \left(a - \frac{2a\alpha}{b}\right) + \frac{3a\alpha}{b(1+bt_{i-1})} \Bigg] \\ & - \left(-a - \frac{2a\alpha}{b}\right) e^{-bt_n}(1+bt_n) + \frac{a\alpha}{b}(1+bt_n) \\ & + \left(a - \frac{2a\alpha}{b}\right) + \frac{3a\alpha}{b(1+bt_n)} \end{aligned} \quad (3.11)$$

Closed form expressions of MLEs of a, α and b cannot be obtained. However, the MLEs can be obtained by iterative solution procedure. Setting the derivatives of the log-likelihood function for $(a\,, \alpha, \beta, b)$ to zero, the MLEs $(\hat{a}, \hat{\alpha}, \hat{b})$ are obtained by iterative solution procedure i.e., Newton Raphson method. We have used R-software for the iterative solution procedure. Let $\hat{a}, \hat{\alpha}, \hat{b}$ be the MLEs of a, α, b respectively.

3.2 Quadratic Fault Content Rate Function

Here we assume that fault introduction rate is a quadratic function and dependent on time, $a(t) = a_2(t) = \delta + \alpha t - \beta t^2, \quad \delta > 0,\ \alpha > 0,\ \beta > 0\ \ t \geq 0$
where δ is the number of initial fault content in the system prior to the testing, and α and β are increasing and decreasing rate of the number of introduced faults to the initial fault content function.

Let $m_2(t)$ be the MVF when fault content function $a_2(t)$ is substituted for $a(t)$ in (2.6) thus have

$$\frac{\partial m_2(t)}{\partial t} = b(t)[\delta + \alpha t - \beta t^2 - m_2(t)].$$
$$\delta > 0, \alpha > 0, \beta > 0\,, b(t) > 0,\ \ t \geq 0 \quad (3.12)$$

with $b(t) = \frac{b^2 t}{(1+bt)}$. Solving the differential equation with respect to $m_2(t)$ under the initial condition $m_2(0) = 0$ we derived the mean value function given by :

$$\begin{aligned} m_2(t) = & \frac{\alpha}{b}(1+bt) + \left(\delta - \frac{2\alpha}{b} + \frac{3\alpha}{b(1+bt)} + \frac{4\beta}{b(1+bt)}\right) \\ & - \left(\frac{2\alpha}{b} + \frac{4\beta}{b} + \delta\right) e^{-bt}(1+bt) \end{aligned} \quad (3.13)$$

We shall call the present NHPP model with MVF given in (3.13) as delayed S-shaped quadratic model. The intensity function of this model is given by:

$$\lambda_2(t) = \alpha - \frac{3\alpha + 4\beta}{(1+bt)^2} + \left(\frac{2\alpha + 4\beta}{b} + \delta\right) e^{-bt} b^2 t \tag{3.14}$$

Based on delayed S-shaped quadratic model, we can derive the following quantitative measures useful for software reliability assessment.

(a) Software Reliability

$$\begin{aligned} R(x|t) &= P\{X_i > x | S_{i-1} = t\} \\ &= \exp[-\{m(x+t) - m(t)\}], t \geq 0, x \geq 0. \end{aligned} \tag{3.15}$$

Substituting (3.13) in (3.15) we have delayed S-shaped quadratic software reliability

$$\begin{aligned} R_2(x|t) = \exp\Bigg[-\Bigg\{&\frac{\alpha}{b}(1+bx) - \frac{(3\alpha+4\beta)x}{\big(1+b(t+x)\big)(1+bt)} \\ &- \left(\frac{2\alpha+4\beta}{b} + \delta\right) e^{-bt}\left[e^{-bx}\big(1+b(t+x)\big) \right. \\ &\left. - (1+bt)\right]\Bigg\}\Bigg], \qquad x, t \geq 0. \end{aligned} \tag{3.16}$$

(b) Expected Residual Fault Content

Consider the Expectation of $\{a(t) - N(t)\}$ for the NHPP model in (3.13). Since $N(t)$ is a random variable, the number of residual fault content at testing time t is:

$$n(t) = a(t) - E[N(t)] = a(t) - m(t) \tag{3.17}$$

Let $n_2(t)$ denote the expected residual fault content for delayed S-shaped quadratic SRM, that is:

$$n_2(t) = a_2(t) - m_2(t) \tag{3.18}$$

$$\begin{aligned} n_{2(t)} = \delta + \alpha t - \beta t^2 - \frac{\alpha}{b}(1+bt) + \left(\delta - \frac{2\alpha}{b} + \frac{3\alpha}{b(1+bt)} + \frac{4\beta}{b(1+bt)}\right) \\ - \left(\frac{2\alpha}{b} + \frac{4\beta}{b} + \delta\right) e^{-bt}(1+bt) \end{aligned} \tag{3.19}$$

3.2.1Parameter Estimation

In this section we discuss the method of estimating the parameters of the method mentioned above

Estimation using interval Domain Data

The likelihood for the interval domain data is given by

$$L = \prod_{i=1}^{n} \frac{\big(m(t_i) - m(t_{i-1})\big)^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} \exp[-m(t_i) - m(t_{i-1})] \tag{3.20}$$

where $t_0 = 0$ and $y_0 = 0$. Taking the natural logarithm of equation (3.20) yields log-likelihood function, that is

$$lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln[m(t_i) - m(t_{i-1})] - m(t_n) - \sum_{i=1}^{n} \ln[(y_i - y_{i-1})!] \tag{3.21}$$

substituting mean value function $m_1(t)$ into the equation (3.21), we get logarithmic of likelihood function excluding the constant terms.

$$\begin{aligned} lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln\Big[&\frac{\alpha}{b}(1+bt_i) \\ &+ \left(\delta - \frac{2\alpha}{b} + \frac{3\alpha}{b(1+bt_i)} + \frac{4\beta}{b(1+bt_i)}\right) \\ &- \left(\frac{2\alpha}{b} + \frac{4\beta}{b} + \delta\right) e^{-bt_i}(1+bt_i) - \frac{\alpha}{b}(1+bt_{i-1}) \\ &+ \left(\delta - \frac{2\alpha}{b} + \frac{3\alpha}{b(1+bt_{i-1})} + \frac{4\beta}{b(1+bt_{i-1})}\right) \\ &- \left(\frac{2\alpha}{b} + \frac{4\beta}{b} + \delta\right) e^{-bt_{i-1}}(1+bt_{i-1})\Big] - \frac{\alpha}{b}(1+bt_n) \\ &+ \left(\delta - \frac{2\alpha}{b} + \frac{3\alpha}{b(1+bt_n)} + \frac{4\beta}{b(1+bt_n)}\right) \\ &- \left(\frac{2\alpha}{b} + \frac{4\beta}{b} + \delta\right) e^{-bt_n}(1+bt_n) \end{aligned} \tag{3.22}$$

Closed form expressions for MLEs of a, α and b cannot be obtained. However, the MLEs can be obtained by iterate solution procedure. Setting the derivatives of the log-likelihood function for (a, α, β, b) to zero, the MLEs $(\hat{a}, \hat{\alpha}, \hat{b})$ are obtained by iterative solution procedure i.e., Newton Raphson method. We have used R-software for the iterative solution procedure. Let $\hat{a}, \hat{\alpha}$ and $\hat{b}$ be the MLEs of $a, \alpha, \hat{b}$ respectively.

3.3 Exponential Fault Content Rate Function

Here we assume that the fault introduction rate is exponential function dependent on time. We chose the following fault content function: $a(t) = a_3(t) = \alpha e^{-\beta t}$, $\alpha > 0$, $\beta > 0$, $t \geq 0$ where α is a number of initial fault content in the system prior to the testing and b is an increasing rate of the number of introduced faults to the initial fault content function. Let $m_3(t)$ be the MVF when fault content function $a_3(t)$ is substituted for $a(t)$ in (2.6) thus we have

$$\frac{\partial m_3(t)}{\partial t} = b(t)\left[\alpha e^{-\beta t} - m_3(t)\right], \quad a > 0, \; b(t) > 0, \quad t \geq 0 \tag{3.23}$$

with $b(t) = \frac{b^2 t}{1+bt}$. Solving the differential equation given in (3.23) with respect to $m_3(t)$ under the initial condition $m_3(0) = 0$ we derived the mean value function given by:

$$m_3(t) = \frac{\alpha\beta b^2 e^{-\beta t}}{\beta(\beta-b)^2(1+bt)}(1-\beta t+bt) - \frac{\alpha\beta b(1+bt)}{(\beta-b)^2}e^{-bt} \qquad (3.24)$$

We shall call the present model with MVF given in (3.24) as delayed S-shaped exponential model. The intensity function of this model is given by:

$$\lambda_3(t) = \frac{\alpha b\beta}{(\beta-b)}\left[e^{-\beta t} - \beta\frac{e^{-\beta t}}{(\beta-b)(1+bt)} - b\frac{e^{-\beta t}}{(\beta-b)(1+bt)^2} - \frac{b^2 t e^{-bt}}{(\beta-b)}\right] \qquad (3.25)$$

Based on delayed S-shaped exponential model, we can derive the following quantitative measures useful for software reliability assessment.

(a) Software Reliability

$$\begin{aligned} R(x|t) &= P\{X_i > x | S_{i-1} = t\} \\ &= \exp[-\{m(x+t) - m(t)\}], \qquad t \geq 0, x \geq 0. \end{aligned} \qquad (3.26)$$

substituting (3.24) in (3.26) we have delayed S-shaped exponential software reliability

$$\begin{aligned} R_3(x|t) = \exp\Bigg[-\Bigg\{&\frac{\alpha b^2}{(\beta-b)^2}\left[\frac{\big(1-\beta(t+x)+b(t+x)\big)e^{-\beta(t+x)}}{(1+b(t+x))} - \frac{(1-\beta t+bt)e^{-\beta t}}{(1+bt)}\right] \\ &- \frac{\alpha\beta b}{(\beta-b)^2}\left[\big(1+\ b(t+x)\big)e^{-b(t+x)} - (1+bt)e^{-bt}\right]\Bigg\}\Bigg], \\ & x, t \geq 0 \end{aligned} \qquad (3.27)$$

(b) Expected Residual Fault Content

Consider the Expectation of $\{a(t) - N(t)\}$ for the NHPP model in (3.24). Since $N(t)$ is a random variable, the number of residual fault content at testing time t is:

$$n(t) = a(t) - E[N(t)] = a(t) - m(t) \qquad (3.28)$$

Let $n_3(t)$ denote the expected residual fault content for delayed S-shaped exponential SRM, that is:

$$n_3(t) = a_3(t) - m_3(t) \qquad (3.29)$$

$$n_3(t) = \alpha e^{-bt} - \frac{\alpha b\beta}{\beta-b}\left[-\frac{e^{-\beta t}}{\beta} + \frac{e^{-\beta t}}{(\beta-b)(1+bt)} - \frac{e^{-bt}(1+bt)}{\beta-b}\right] \qquad (3.30)$$

3.3.1Estimation of Parameters Using Interval Domain Data

The likelihood function mentioned above for the interval domain data is given by

$$L = \prod_{i=1}^{n}\frac{\big(m(t_i) - m(t_{i-1})\big)^{y_i - y_{i-1}}}{(y_i - y_{i-1})!}\exp[-m(t_i) - m(t_{i-1})] \qquad (3.31)$$

where $t_0 = 0$ and $y_0 = 0$. Taking the natural logarithm of equation (3.31) yields log-likelihood function , that is

$$lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln[m(t_i) - m(t_{i-1})] - m(t_n) - \sum_{i=1}^{n} \ln[(y_i - y_{i-1})!] \qquad (3.32)$$

substituting mean value function $m_1(t)$ into the equation (3.32), we get logarithmic of likelihood function excluding the constant terms.

$$lnL = \sum_{i=1}^{n} (y_i - y_{i-1}) \ln \left[\frac{\alpha b \beta}{\beta - b} \left[-\frac{e^{-\beta t_i}}{\beta} + \frac{e^{-\beta t_i}}{(\beta - b)(1 + bt_i)} - \frac{e^{-bt_i}(1 + bt_i)}{(\beta - b)} \right] \right.$$
$$- \frac{\alpha b \beta}{(\beta - b)} \left[-\frac{e^{-\beta t_{i-1}}}{\beta} + \frac{e^{-\beta t_{i-1}}}{(\beta - b)(1 + bt_{i-1})} \right.$$
$$\left. \left. - \frac{e^{bt_{i-1}}(1 + bt_{i-1})}{(\beta - b)} \right] \right]$$
$$- \frac{\alpha b \beta}{(\beta - b)} \left[-\frac{e^{-\beta t_n}}{\beta} + \frac{e^{-\beta t_n}}{(\beta - b)(1 + bt_n)} - \frac{e^{-bt_n}(1 + bt_n)}{(\beta - b)} \right] \qquad (3.33)$$

Closed form expressions for MLEs of a, α and b cannot be obtained. However, the MLEs can be obtained by iterate solution procedure. Setting the derivatives of the log-likelihood function for (a , α, β, b) to zero, the MLEs $(\hat{a}, \hat{\alpha}, \hat{b})$ are obtained by iterative solution procedure i.e., Newton Raphson method. We have used R-software for the iterative solution procedure. Let $\hat{a}, \hat{\alpha}, \hat{b}$ be the MLEs of a, α, b respectively.

4. Analysis of Three Data Sets

The Data Set I is about US Tactical Data Systems (NTDS) given by [Goel and Okumoto (1979)a]: the software data set was extracted from information about failures in the development of software for the real time multi-computer complex of the US Naval Fleet Computer Programming Center of the US Naval Tactical Data System (NTDS). The software consists of 38 different project modules. The time horizon is divided into four phases: Production phase, test phase, user phase and subsequent test phase. The 26 software failures were found during the production phase, five during the test phase and the last failure was found on 1th Jan 1971. One failure was observed during the user phase, in September 1971, and two failures during the test phase in 1971. The Data Set II is about On Line Data Entry IBM Software Package: The data reported by [Ohba (1984)] are recorded from testing an online data entry software package developed by IBM. The Data Set III is about Real-Time Command and Control System: The data set was reported by [Musa et el. (1987)] based on failure data from a real-time command and control system, which represents the failure observed during system testing for 25 hours of CPU time.

Applying data sets DS I, DS II and DS III on the models discussed above, we obtain the maximum likelihood (ML) estimates and goodness of fit measures, such as sum of squares due to error (SSE), mean squares error (MSE), predictive ratio risk (PRR) [Pham and Deng (2003)] and Akaike's information criterion (AIC) values for existing and for the proposed finite failure models are summarized in Table 1, 2 and 3 respectively. Further we have plotted the actual data and estimated values of cumulative faults (or Mean Value Function (MVF)) of finite failure models against time for data sets DS I, DS II and DS II as shown in Figures 1, 2 and 3 respectively. The S-shaped model, Yamada et al (1984), with fault content function as constant is compared with the proposed delayed S-shaped linear, quadratic and exponential model.

From the Table 1, we observe that the SSE, MSE, PRR, AIC and BIC values of our proposed delayed S-shaped linear, quadratic and exponential models are smaller than the existing delayed S-shaped and Pham model. Hence we may conclude that our proposed models perform better. Further we observe that the SSE, MSE, PRR, AIC and BIC values for delayed S-shaped exponential model are smaller as compared to the other two proposed models and existing delayed S-shaped and Pham model. Hence we conclude that our proposed delayed S-shaped exponential model performs better than all other models for Data Set I.

Table 1 : ML and AIC estimates of Models for DS I

Model Name	Estimates	SSE	MSE	PRR	AIC	BIC
Delayed S-shaped	$\hat{a}$=27.5041 $\hat{b}$=0.0386	1208.813	50.3672	27.2446	169.8306	171.5162
Pham	$\hat{\alpha}$=0.00013 $\hat{\gamma}$=1.84	2258.013	94.08389	297.8886	116.1015	118.6177
Delayed S-shaped Linear	$\hat{a}$=12.35044 $\hat{\alpha}$=0.003966 $\hat{b}$=0.02737	269.5382	11.71905	17.5369	98.17724	101.9515
Delayed S-shaped Quadratic	$\hat{\delta}$=4 $\hat{\alpha}$=0.1 $\hat{\beta}$=0.2 $\hat{b}$=0.1	246.4403	11.1846	5.335685	97.08888	102.1213
Delayed S-shaped Exponential	$\hat{\alpha}$=39.03 $\hat{\beta}$=0.001 $\hat{b}$=0.0048	156.0458774	6.784603	1.984983	90.48007	94.25436

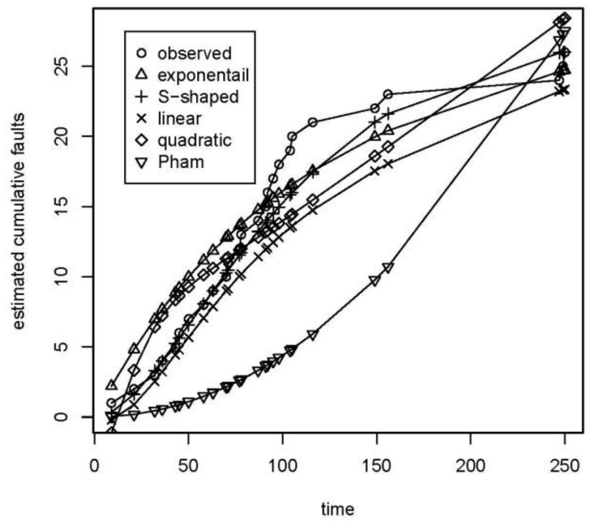

Figure 1 : Plot of Data and Estimated cumulative Faults against time of DS I

From the Figure 1, we observe that for our proposed delayed S-shaped linear, quadratic and exponential models, the estimated MVF is closer to the actual data points as compared to the MVF of existing delayed S-shaped and Pham software reliability growth models. Hence we conclude that our proposed models explains the data better than the given existing models.

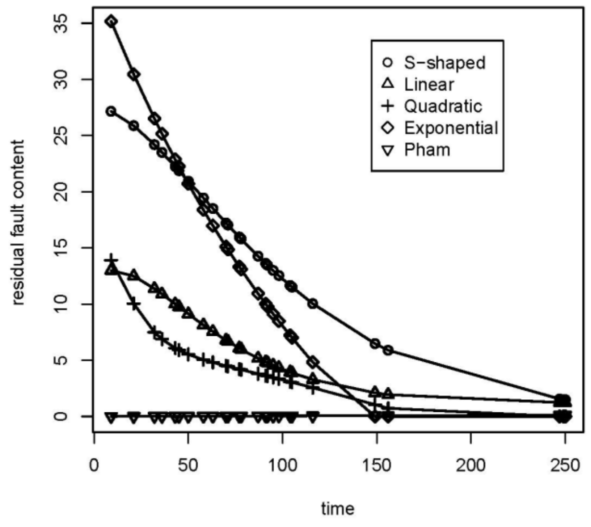

Figure 2 : Plot of Estimated Residual Fault Content Function against time of DS I

The Figure 2 illustrates the behavior of residual fault content functions of software reliability growth models for delayed S-shaped model and our proposed delayed S-shaped linear, quadratic and exponential models. We observe that in comparison with the existing closed S-shaped model our proposed models residual fault content functions are getting closer to zero as time increases. Hence our proposed models perform better for data set I

From the Table 2, we observe that the SSE, MSE, PRR, AIC and BIC values of our proposed delayed S-shaped linear, quadratic and exponential models are smaller than the existing delayed S-shaped model and Pham model. Hence we may conclude that our proposed models perform better. Further we observe that the SSE, MSE, PRR, AIC and BIC values for delayed S-shaped exponential model are smaller as compared to the other two proposed models and existing delayed S-shaped model and Pham model. Hence we conclude that our proposed delayed S-shaped exponential perform better than all other models for Data Set II.

Table 2 : ML and AIC estimates of Models for DS II

Model Name	Estimates	SSE	MSE	PRR	AIC	BIC
Delayed S-shaped	$\hat{a}$=22.6199 $\hat{b}$=0.0080	467.2066	23.3603	12460.87	202.2326	204.4147
Pham	$\hat{\alpha}$=0.000015 $\hat{\gamma}$=1.8	1383.409	69.17047	885.3575	127.2027	129.1821
Delayed S-shaped Linear	$\hat{a}$=8 $\hat{\alpha}$=0.0037 $\hat{b}$=0.02636	107.255	5.645	49.84465	97.00758	100.2807
Delayed S-shaped Quadratic	$\hat{\delta}$=4.9 $\hat{\alpha}$=0.030 $\hat{\beta}$=0.001 $\hat{b}$=0.1	88.9676	4.942644	7.197857	94.8933	99.25747
Delayed S-shaped Exponential	$\hat{\alpha}$=23 $\hat{\beta}$=0.0010095 $\hat{b}$=0.0040436	48.212217	2.537483	5.288426	87.35715	90.63028

From the Figure 3, we observe that for our proposed delayed S-shaped linear, quadratic and exponential models, the estimated MVF is closer to the actual data points as compared to the MVF of existing delayed S-shaped and Pham software reliability growth models. Hence we conclude that our proposed models explain the data than the given existing model

The Figure 4 illustrates the behavior of residual fault content functions of software reliability growth models for delayed S-shaped model and our proposed delayed S-shaped linear, quadratic and exponential models. We observe that in comparison with the existing closed S-shaped model our proposed models residual fault content functions are getting closer to zero as time increases. Hence our proposed models perform better for data set II.

From the Table 3, we observe that the SSE, MSE, PRR, AIC and BIC values of our proposed delayed S-shaped linear, quadratic and exponential models are smaller than the existing delayed S-shaped model and Pham model. Hence we may conclude that our proposed models perform better. Further we observe that the SSE, MSE, PRR, AIC and BIC values for delayed S-shaped exponential model are smaller as compared to the other two proposed models and existing delayed S-shaped model and Pham model. Hence we conclude that our proposed delayed S-shaped exponential perform better than all other models for Data Set III.

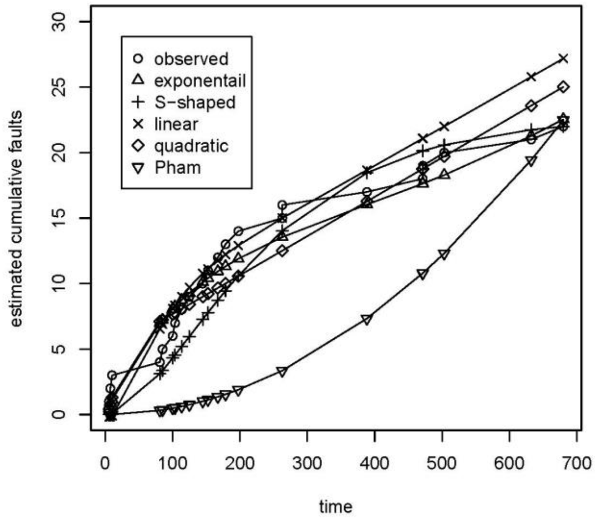

Figure 3 : Plot of Data and Estimated Cumulative Faults against Time of DS II

From the Figure 5, we observe that for our proposed delayed S-shaped linear, quadratic and exponential models, the estimated MVF is closer to the actual data points as compared to the MVF of existing delayed S-shaped and Pham software reliability growth models. Hence we conclude that our proposed models explain the data than the given existing model

The Figure 6 illustrates the behavior of residual fault content functions of software reliability growth models for delayed S-shaped model and our proposed delayed S-shaped linear, quadratic and exponential models. We observe that in comparison with the existing closed S-shaped model our proposed models residual fault content functions are getting closer to zero as time increases. Hence our proposed models perform better for data set III.

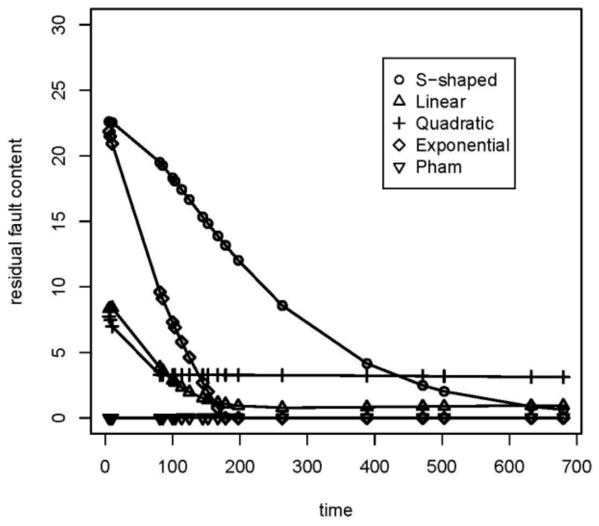

Figure 4 : Plot Estimated Residual Fault Content Function against time of DS II

Table 3 : ML and AIC estimates of Models for DS III

Model Name	Estimates	SSE	MSE	PRR	AIC	BIC
Delayed S-shaped	$\hat{a}$=16.4356 $\hat{b}$=0.0137	24.3006	1.8693	54.5298	124.1217	125.8378
Pham	$\hat{\alpha}$=0.0001 $\hat{\gamma}$=1.3	313	24.0928	281.0441	49.48292	50.89902
Delayed S-shaped Linear	$\hat{a}$=3.7688809 $\hat{\alpha}$=0.0098142 $\hat{b}$=0.0423244	22.00142	1.833452	6.315455	34.61784	36.77199
Delayed S-shaped Quadratic	$\hat{\delta}$=3 $\hat{\alpha}$=0.045 $\hat{\beta}$=0.3 $\hat{b}$=0.11	19.40438	1.730949	3.179109	32.81919	35.65139
Delayed S-shaped Exponential	$\hat{\alpha}$=20.072 $\hat{\beta}$=0.0020095 $\hat{b}$=0.0054436	3.22893	0.2690775	0.03946524	30.53145	32.6556

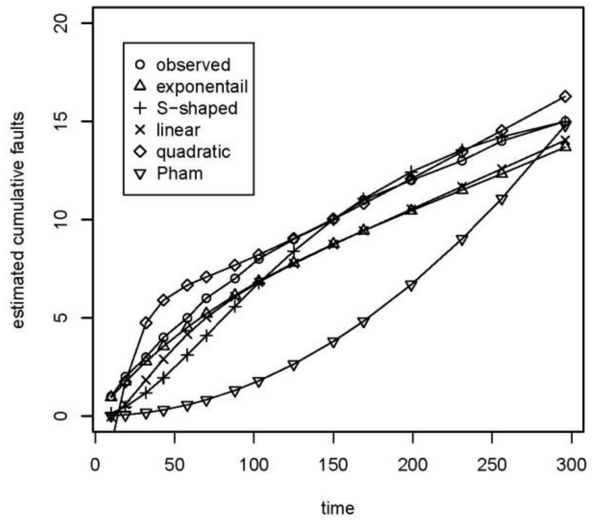

Figure 5 : Plot of Data and Estimated Cumulative Faults against of DS III.

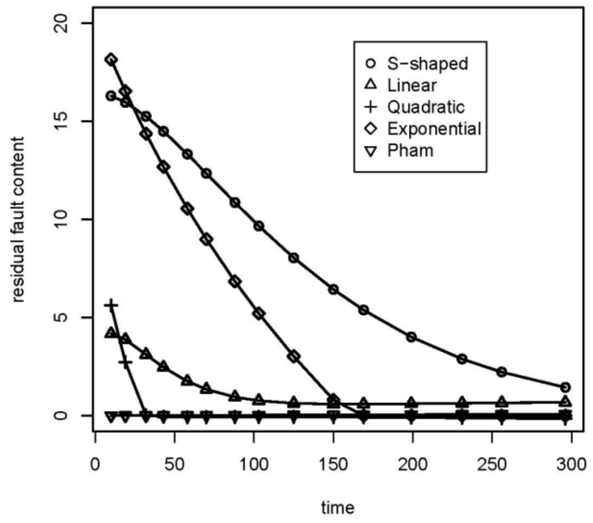

Figure 6 : Plot of Estimated Residual Fault Content Function against time of DS III

5. Conclusions and Remarks

In this paper we proposed three different software reliability growth models based on three types of fault content rate function $a(t)$. The model parameters are estimated using maximum likelihood method by using the interval domain data type. Three data sets are illustrated for the above estimate technique and model comparisons. The proposed models are compared with the existing delayed S-shaped models, using sum of squares due to error, mean squares sum of error, predictive ratio risk, Akaikes information criterion and Bayesian Information criterion. It may be observed that the proposed models perform better than the existing models for all the three data sets. Further we also plotted estimated MVF values versus time.

Acknowledgements

We thank the referee for the valuable suggestions and comments which resulted into the improved version of the earlier manuscript.

References

[1] Ahmad, N., Khan, M.G.M. and Rafi L.S. (2011). Analysis of an inflection S-shaped software reliability model considering log-logistic testing-effort and imperfect debugging. International Journal of Computer Science and Network Security, 11(11), 161-171.

[2] Bittanti, S., Bolzern, P., Pedrotti, E., Pozzi, N., and Scattolini, R. (1988). A flexible modeling approach for software growth. In software Reliability Modeling and Identification, G. Goos and J. Haramanis, Eds. Berlin: Springer Verlag, pp. 101-140.

[3] Bokhari, M.U., Siddiqui, A., and Ahmad, N. (2017). Testing effort dependent delayed S-shaped software reliability growth model with imperfect debugging. International Journal of Computer Science and Engineering, 9(5), 138-148

[4] Chatterjee, S. and Shukla, A. (2016). Change point-based software model under imperfect debugging with revised concept of fault dependency. Journal of Risk and Reliability, 230(6), 579-597.

[5] Chen, Z. (2010). Empirical Bayes analysis on the power law process with natural conjugate priors. Journal of Data Science, 8(1), 151-172.

[6] Gupta, A., Choudhary, D. and Saxena, S. (2011). Software reliability estimation using Yamada delayed S-shaped model under imperfect debugging and time lag. International Journal of Computer Applications, 23(7), 49-52.

[7] Goel, A. L. and Okumotu, K. (1979). Time dependent fault-detection rate model for software and other performance measures. IEEE Transaction on Reliability, vol.28 pp. 206-211.

[8] Hanagal, D. D. and Bhalerao, N. N. (2016a). Modeling and statistical inference on generalized inverse Weibull software reliability growth model. Journal of the Indian Society for Probability and Statistics, 17(2), 145-160.

[9] Hanagal, D. D. and Bhalerao, N. N. (2016b). Analysis of NHPP software reliability growth models. International Journal of Statistics and Reliability, Engineering, 3(2), 53-67.

[10] Hanagal, D. D. and Bhalerao, N. N. (2017). Modeling on extended inverse Weibull software reliability growth model. Journal of Indian Association for Product, Quality & Reliability, 42(2), 75-95.

[11] Kapur, P. K. and Garg, R.B. (1992). A software reliability growth model for an error removal phenomenon, Software Engineering Journal, 7, 291-294.

[12] Kapur, P. K. and Garg, R.B. and Kumar, S. (1999). Contributions to hardware and software reliability, World Scientific Publishing Co. Ltd., Singapore.

[13] Kareer, N., Kapur, P. K. and Grover, P.S. (1990). An S-shaped software reliability growth model with two types of errors. Microelectronics Reliability, 30(6), 1085-1090.

[14] Kaur, D. and Sharma, M. (2015). Software Reliability models: Time between failures and accuracy estimation. International Journal of Computer Science and Information Technologies, 6(4), 3370-3373.

[15] Lai, R. and Garg, M. (2012). A detailed study of software reliability models, Journal of software, 7(6), 1296-1306.

[16] Musa, J. D., Lannino, A. and Okumoto, K. (1987). Software Reliability: Measurement, Prediction and Application, McGraw-Hill, New York.

[17] Obha, M. and Chou, X. (1989). Does imperfect debugging affect software reliability growth models, Proceeding 4th International Conference on Reliability and Maintainability, Tregastel, France, pp 430-436.

[18] Obha, M. and Yamada, S. (1984). S-shaped software reliability growth models. In Proceedings of the 4th International Conference on Reliability and Maintainability, Tregastel, France, pp 430-436.

[19] Obha, H., Osaki, S. and Y. Hatoyama (1984). Inflection S-shaped software reliability growth models. Stochastic Models in Reliability Theory, Springer-Verlag, Berlin, 144-162.

[20] Pham, H. (1996). A software cost model with imperfect debugging, random life cycle and penalty cost. International Journal of Systems Science, vol. 27(5), 455-46.

[21] Pham, H. and Zhang, X. (1997). An NHPP software reliability model and its comparison. International Journal of Reliability, Quality and safety Engineering, 4, 269-282.

[22] Pham, H. (1993). Software Reliability Assessment: Imperfect debugging and multiple failure types in software development. EGrandG-RAAM-10737, Idaho National Engineering Laboratory.

[23] Pham, H. (1999). Reliability analysis for dynamic configurations of systems with three failures models, Reliability Engineering and System Safety, 63, 13-23.

[24] Pham, H. (2006). System Software Reliability, Springer Series in Reliability Engineering Springer, London, pp. 149-149.

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

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

[27] Pham, H. and Zhang, X. (2003). NHPP software reliability and cost models with testing converges. European Journal of Operational Research, 145(2), 443-454.

[28] Teng, X. and Pham, H. (2004). Software cost model for quantifying the gain with considerations of random field environments. IEEE transactions Computers, 53(3), 380-384.

[29] Teng, X. and Pham, H. (2006). A new methodology for predicting software reliability in the random field environments, IEEE Transactions on Reliability, 53(3), 458-468.

[30] Tokuno, K. and Yamada, S. (2000). An imperfect debugging model with two types of hazard rates for software reliability measurement and assessment. Mathematical and Computer Modelling, 3, 343-352.

[31] Yamada, S. and Osaki, S. (1985). Reliability growth modeling: Models and applications. IEEE Transactions on Software Engineering, 11(12), 1431-1437.

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

[33] Yamada, S., Obha, M. and Osaki, S. (1984). S-shaped software reliability growth models and their applications. IEEE Transactions on Reliability, 33, 289-292.

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

[35] Zhang, X. and Pham, H. (2000). Comparisons of non-homogeneous Poisson process software reliability models and its applications. International Journal of Systems Science, 31(9), 1115-1123.

[36] Zhang, X., Teng, H. and Pham, H. (2003). Considering fault removal efficiency in software reliability assessment. IEEE Transactions on System, Management, and Cybernetics-Part A, 33(1), 114-120.

[37] Zaho, M. and Xie, M. (1996). On maximum likelihood estimation on a general non-homogeneous Poisson process. Scandinavian J. Statistics, 23(4), 597-607.