Stochastic Differential Equation Based Software Reliability Growth Modeling With Change Point and Two Types of Imperfect Debugging

P.K. Kapur[1], **Ompal Singh**[2] and **Jagvinder Singh**[3]
Department of Operational Research, University of Delhi, Delhi-110007
[1]pkkapur1@gmail.com and [3]jagvinder.singh@gmail.com

ABSTRACT

Software Reliability is defined as the probability of failure free operation for a specified period of time in a specified environment. If the size of the software system is large, and the number of the faults which are detected and removed through debugging activities becomes sufficiently small compared with the initial faults content at the beginning of the testing phase, in such a situation, we can model the software faults detection process as a stochastic process with continuous state space. Due to the complexity of large software system and incomplete understanding of the software, the testing team may not be able to remove/correct the fault perfectly on observation/detection of a failure and the original fault may remain resulting in a phenomenon known as imperfect debugging, or get replaced by another fault causing error generation. During software testing fault detection/correction rate may not be same throughout the whole testing process, but it may change at any time moment known as change-point. In this paper, we have proposed $it\hat{o}$ *type of stochastic differential equation (SDE) based Software Reliability Growth Models (SRGM) with change-point and two types of imperfect debugging. The proposed model is validated on number of data sets and compared with the result of other established models.*

KEYWORDS

Software Reliability growth model (SRGM), Stochastic Differential Equation (SDE), Imperfect debugging, Change-Point.

1. INTRODUCTION

With the ever–increasing dependence upon the computer system, their reliability has become more and more important. In recent years, computers are used in safety-critical applications such as medicine, transportation and nuclear energy. For a lot of software embedded system, software reliability has been the dominant driver of today's system reliability. It leads to a great demand for high quality software products. However, the poor performance due to unreliable software is exhibited by many systems. To improve the software quality, software reliability engineering plays an important role in many aspects throughout the software life cycle. For example, before software is released for installation or operate to the customers, accurate reliability estimates are required to verify the quality of the software. To assess the reliability of software is also critical in determining the optimal release time of a software system. As we know, there are many reasons for software to fail but usually software fails because of a design problem. Other failures occur when the code is written, or when changes are introduced to a working system (Fenton and Pfleeger, 1997).

Software error detection is one of the most challenging problems in software engineering. Error detection techniques have an important role throughout the lifetime of systems. No matter how thoroughly a system has been assessed before use, information from its actual failure behavior in use is precious. Reported errors and failures can lead to faults being corrected. Hence a lot of emphasis is put on avoiding introduction of faults during Software development and to remove dormant faults before the product is released for use. The testing phase is given a lot of importance and nearly half of the developmental resources are spent during this phase. This phase aims at detecting and removing the faults, which may have been introduced during software development process, thereby increasing the reliability of the software. The SRGM (Software Reliability Growth Model) is a tool of SRE (Software Reliability Engineering), which can be used to evaluate the software quantitatively, develop test status, schedule status and monitor the changes in reliability Performance [5]. In the last three decades several Software Reliability models have been developed in the literature to estimate the fault content, failure rate and fault removal rate.

The non-homogenous Poisson process (NHPP) based SRGMs have proved quite successful in practical software reliability engineering (Musa et al., 1987). Goel and Okumoto [1] proposed an SRGM, which describes the fault detection (FD) rate, as a Non Homogeneous Poisson Process (NHPP) assuming the hazard rate is proportional to the remaining number of faults. Later researchers proposed many SRGMs, which describe FD&/or FC (Fault correction) process by NHPP following the basic assumptions of GO model. Yamada et al. [16] proposed a modified exponential SRGM assuming the software contains two types of faults. The model is based on the observation that in the early stages of the software phase, the testing team removes a large number of simple faults (faults that are easy to remove) while the hard faults are removed in the later stages of the testing phase. There is several Software Reliability models have been developed in the literature showing that the relationship between the testing time and the corresponding number of faults removed. They are either Exponential or S-shaped or a mix of the two [7-13]. The software includes different types of faults, and each fault requires different strategies and different amounts of testing effort to remove it. Ohba [6] refined the Goel-Okumoto model by assuming that the fault detection/removal rate increases with time and that there are two types of faults in the software. SRGM proposed by Bittanti et al. [15] and Kapur and Garg [10] has similar forms as that of Ohba [6] but is developed under different set of assumptions. Many of these SRGMs assume that each time a failure occurs, the fault that caused it can be immediately removed and no new faults are introduced, which is usually called *perfect debugging*. But Imperfect debugging models have proposed a relaxation of the above assumption, (Obha and Chou 1989) is an fault generation model applied on G-O model and has been also named as imperfect debugging model. (Kapur and Garg 1990) introduced the imperfect debugging in G-O model. They assume that the fault detection rate per remaining faults is reduced due to imperfect debugging thus the faults detected by time infinity is more than the initial fault content. Although these two models describe the imperfect debugging phenomenon yet the software reliability growth curve of these models is always exponential. Moreover they assume that the probability of imperfect debugging is independent of the testing time. Thus they ignore the role of the learning process during the testing phase by not accounting for the experience gained with the progress of software testing. (Pham 2006) developed an SRGM for multiple failure types incorporating fault generation. (Zhang et al.2003) proposed a testing efficiency model which includes both imperfect debugging and fault generation. (Kapur at al. 2006) proposed a flexible SRGM with imperfect debugging and fault generation using a logistic function for fault detection rate which reflects the efficiency of the testing team. Several NHPP based SRGMs have been developed in the literature, treating fault detection process during testing phase as a discrete counting process. Recently (Yamada et. al 2003) asserted that if the size of the software system is large then the number of fault detected during the testing phase becomes large and the change of the number of faults which are detected and removed through each debugging activities becomes sufficiently small compared with the initial fault content at the beginning of the testing phase. Therefore, in order to describe the stochastic behavior of the fault detection process, a stochastic model with continuous state space can be used. (Yamda, Nishigaki and Kimura 2003, Yamda and Tamura 2006; Lee, Kim and Park 2) have studied the stochastic behavior of fault detection process described by stochastic process model with continuous state space. Recently (Kapur et.al 2009) proposed a SDE based flexile SRGM. The other assumption of above discussed NHPP based SRGMs is that each failure occurs independently and randomly in time according to the same distribution during the fault detection process (Musa et al., 1987). However, in more realistic situations, the failure distribution can be affected by many factors, such as the running environment, testing strategy and resource allocation. Once these factors are changed during the software-testing phase, this could result in a software failure intensity function that increases or decreases non-monotonically. It is identified as a change-point problem (Zhao, 1993). (Shyur 2003) proposed a SRGM with the concept of imperfect debugging (error generation) with change point. Recently (Kapur et. al) proposed SDE based SRGM with change point. In software reliability estimation the change-point effect should be considered simultaneously if there is a change-point exists. Otherwise the estimation model cannot express the factual software reliability behavior. In this paper, we proposed three SDE based SRGM with Change-point and both type of imperfect debugging. Both of these features reflect more closely to a general SRGM. An overview of previous NHPP models and a brief description of the proposed model have been discussed in this paper. The parameters are estimated using non linear least square method. The estimation is done using the software SPSS (Statistical Package for social Sciences) and software Change Point Analyzer and the model is compared to the existing model using goodness of fit criteria.

2. NOTATIONS

$N(t)$: Number of faults detected during the testing time t and is a random variable.

$E[N(t)]$: Expected number of faults detected in the time interval (0, t] during testing phase.

$M(t)$: The mean value function or the expected number of faults detected or removed by time t.

$a(t)$: Total fault content of the software dependent on the time.

a : Constant, representing the initial number of faults lying dormant in the software when the testing starts.

p : The probability of fault removal on a failure (i.e., the probability of perfect debugging) before and after the change point.

α_1, α_2 : The rate at which the errors may be introduced during the debugging process per detected fault before and after the change point respectively.

σ : Positive constant that represents the magnitude of the irregular fluctuation for faults before and after change point.

$b(t)$: Time dependent rate of fault removal per remaining faults.

b_1, b_2 : Proportionality constant fault removal rate per remaining fault before and after the change point respectively.

β_1, β_2: Constant parameter describing learning in the fault removal rate before and after the change point.

3. BASIC ASSUMPTIONS

The proposed model is based on SDE of $\hat{ito}$ type with following assumption:

1. The software fault detection Process is modeled as a stochastic process with a continuous state space.
2. Software is subject to failure during execution caused by faults remaining in the software.
3. It is assumed that the fault detection rate $b(t)$ may change at any time moment (called change point) and follows exponential growth curve.
4. Let $N(t)$ be a random variable which presents the number of software faults detected in the software system up to testing time t. The faults detected in $t+\Delta t$ are proportional to the mean number of faults remaining in the system.
5. When a software failure occurs, an instantaneous repair efforts start and the following may occur:
   (a) Fault content is reduced by one with probability p.
   (b) Fault content remains unchanged with probability $1-p$.
6. During the fault removal process, whether the fault is removed successfully or not, new faults are generated with a constant probability α.

4. SDE MODELING FOR TWO TYPES OF IMPERFECT DEBUGGING AND CHANGE POINT

4.1FRAMEWORK FOR MODELING OF PROPOSED SRGM

Several SRGM are based on the assumption of NHPP, treating the fault detection process during the testing phase as a discrete counting process. recently Yamada et al. 2003, asserted that if the size of the software system is large then the number of faults detected during the testing phase also is large and change in the number of faults, which are corrected and removed through each debugging, becomes small compared with the initial fault content at the beginning of the testing phase. So, in order to describe the stochastic behavior of the fault detection process, we can use a stochastic model with continuous state space. Since the latent fault in the software system are detected and eliminated during the testing phase, the number of faults remaining in the software system gradually decreases as the testing progresses .therefore; it is reasonable to assume the following differential equation:

$$\frac{d}{dt}N(t) = \begin{cases} b(t)p[a(t)-N(t)] & for\ 0 \le t \le \tau \\ b(t)p[a(t)-N(t)] & for\ t > \tau \end{cases} \tag{1}$$

Where $b(t) = \begin{cases} b_1(t) & \text{for } 0 \le t \ge \tau \\ b_2(t) & \text{for } t > \tau \end{cases}$

And,

$$a(t) = \begin{cases} a + \alpha_1 N(t) & for\ 0 \le t \le \tau \\ a + \alpha_1 N(\tau) + \alpha_2 (N(t) - N(\tau)) & for\ t > \tau \end{cases}$$

So the above equation can be written as:

$$\frac{d}{dt}N(t) = \begin{cases} p(1-\alpha_1)b_1(t)[a_1 - N(t)] & for\ 0 \le t \le \tau \\ p(1-\alpha_2)b_2(t)[a_2 - N(t)] & for\ t > \tau \end{cases} \tag{2}$$

Where $a_1 = \frac{a}{(1-\alpha_1)}$ and $a_2 = \frac{a + (\alpha_1 - \alpha_2)N(\tau)}{(1-\alpha_2)}$

Here $b_1(t)$ and $b_2(t)$ are the fault detection rates before and after the change point respectively and p is the probability of perfect debugging before and after the change point

So in the above case fault detection rate reduces to $pb_1(t)$ and $pb_2(t)$ respectively and in case of error generation the number of faults will increase therefore fault detection rate decreases to $p(1-\alpha_1)b_1(t)$ and $p(1-\alpha_2)b_2(t)$ before and after the change point and denoted by $h_1(t)$ and $h_2(t)$ respectively as a reduced fault detection rate.

So equation (2) can be written as.

$$\frac{d}{dt}N(t) = \begin{cases} h_1(t)[a_1 - N(t)] & for\ 0 \le t \le \tau \\ h_2(t)[a_2 - N(t)] & for\ t > \tau \end{cases} \tag{3}$$

It might happen that $h_1(t)$ and $h_2(t)$ are not completely known, but subject to some random environment effects such as the testing effort expenditure, the skill level of the testers, the testing tool and so on and thus might have irregular fluctuation in the reduced fault detection rate. Thus, we have:

$$h(t) = \begin{cases} h_1(t) + "noise" & for\ 0 \le t \le \tau \\ h_2(t) + "noise" & for\ t > t \end{cases}.$$

In the above equation we do not know the exact behavior of noise term, only its probability distribution is known. The function $h(t)$ is assumed to be non-random. Let $\gamma(t)$ be a standard Gaussian white noise and 'σ' be a positive constant representing a magnitude of the irregular fluctuations before and after the change point. So equation (3) can be written as:

$$\frac{d}{dt}N(t)=\begin{cases}[h_1(t)+\sigma\gamma(t)][a_1-N(t)] & for\ 0\le t\le\tau\\ [h_2(t)+\sigma\gamma(t)][a_2-N(t)] & for\ t>\tau\end{cases} \quad (4)$$

Equation (4) can be extended to the following stochastic differential equation of an it$\hat{}$o type

$$dN(t)=\begin{cases}[h_1(t)-\frac{1}{2}\sigma^2][a_1-N(t)]dt+\sigma[a_1-N(t)]dw(t) & for\ 0\le t\le\tau\\ [h_2(t)-\frac{1}{2}\sigma^2][a_2-N(t)]dt+\sigma[a_2-N(t)]dw(t) & for\ t>\tau\end{cases} \quad (5)$$

Where $w(t)$ is a one-dimensional Wiener process before and after the change- point, which is formally defined as an integration of the white noise $\gamma(t)$ with respect to time t.

Using $\hat{\text{Ito}}$ formula, solution to equation (5) using initial condition we get N(t) as follows:

At $t=0$, N(0)=0 and at $t=\tau$ $N(t)=N(\tau)$;

$$N(t)=\begin{cases}\frac{a}{(1-\alpha_1)}(1-e^{-\int_0^t h_1(x)dx-\sigma w(t)}) & for\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}(1-e^{-\int_0^\tau h_1(x)dx-\int_\tau^t h_2(x)dx-\sigma w(t)})+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}\text{m}(\tau) & for\ t>\tau\end{cases}$$

Or (6)

$$N(t)=\begin{cases}\frac{a}{(1-\alpha_1)}(1-e^{-p(1-\alpha_1)\int_0^t b_1(x)dx-\sigma w(t)}) & for\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}(1-e^{-p(1-\alpha_1)\int_0^\tau b_1(x)dx-p(1-\alpha_2)\int_\tau^t b_2(x)dt-\sigma w(t)})+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & for\ t>\tau\end{cases} \quad (7)$$

Using the fact that the wiener process $w(t)$, is a Gaussian process and has the following properties:

$$\Pr[w(0)=0]=1,$$
$$E[w(t)]=0;$$
$$E[w(t)w(t')]=\min[t,t']$$

4.1.1 Proposed SRGM-I

In this proposed model, it is assumed that the fault detection rate $b(t)$ may change at any time moment and follows exponential growth curve. Fault detection rate is defined as

$$b(t)=\begin{cases}b_1 & \text{for } 0\le \text{t}\ge\tau\\ b_2 & \text{for t}>\tau\end{cases}$$

Now considering the stochastic differential equation (7), the transition probability distribution is:

$$N(t)=\begin{cases}\frac{a}{(1-\alpha_1)}(1-e^{-p(1-\alpha_1)\int_0^t b_1(x)dx-\sigma w(t)}) & for\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}(1-e^{-p(1-\alpha_1)\int_0^\tau b_1(x)dx-p(1-\alpha_2)\int_\tau^t b_2(x)dt-\sigma w(t)})+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & for\ t>\tau\end{cases}$$

Now putting the value of $b(t)$ in the above equation, we get

$$N(t)=\begin{cases}\frac{a}{(1-\alpha_1)}(1-e^{-p(1-\alpha_1)b_1t-\sigma w(t)}) & for\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}(1-e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)-\sigma w(t)})+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & for\ t>\tau\end{cases} \quad (8)$$

As we know that the Brownian motion or wiener Process follows normal distribution. The density function of w (t) is given by:

$$f(w(t))=\frac{1}{\sqrt{2\pi t}}\exp\left\{-\frac{(w(t))^2}{2t}\right\}.$$

Thus the mean number of detected fault is given as

$$m(t)=E[N(t)]=\begin{cases}\frac{a}{(1-\alpha_1)}(1-e^{-p(1-\alpha_1)b_1t+\frac{1}{2}\sigma^2t}) & for\ 0\le t\le\tau\\ \frac{a}{(1-\alpha_2)}(1-e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2t})+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & for\ t>\tau\end{cases} \quad (9)$$

If fluctuation is assumed to be zero then the above model reduces to imperfect debugging and change point model given by Shyur (2003).

4.1.2 Proposed SRGM-II

In this proposed model it is assumed that the fault detection rate $b(t)$ may be changed at any time moment and follows S-shaped growth curve. Therefore the fault detection rate can be defined as:

$$b(t)=\begin{cases}\dfrac{b_1^2 t}{1+b_1 t} & \text{for } 0\le t\le\tau\\[2mm] \dfrac{b_2^2 t}{1+b_2 t} & \text{for } t>\tau\end{cases}$$

Now again considering the stochastic differential equation (7), the transition probability distribution is:

$$N(t)=\begin{cases}\dfrac{a}{(1-\alpha_1)}\left(1-e^{-p(1-\alpha_1)\int_0^t b_1(x)dx-\sigma w(t)}\right) & \text{for } 0\le t\le\tau\\[2mm] \dfrac{a}{(1-\alpha_2)}\left(1-e^{-p(1-\alpha_1)\int_0^\tau b_1(x)dx-p(1-\alpha_2)\int_\tau^t b_2(x)dt-\sigma w(t)}\right)+\dfrac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & \text{for } t>\tau\end{cases}$$

Now putting the value of $b(t)$ in the above equation, we get

$$N(t)=\begin{cases}\dfrac{a}{(1-\alpha_1)}\left(1-(1+b_1t)^{p(1-\alpha_1)}e^{-p(1-\alpha_1)b_1t-\sigma w(t)}\right) & \text{for } 0\le t\le\tau\\[2mm] \dfrac{a}{(1-\alpha_2)}\left(1-(1+b_1\tau)^{p(1-\alpha_1)}\left(\frac{1+b_2t}{1+b_2\tau}\right)^{p(1-\alpha_2)}e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)-\sigma w(t)}\right)\\ \qquad +\dfrac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & \text{for } t>\tau\end{cases} \quad (10)$$

We consider the mean number of faults detected up to time t. as we know that the Brownian motion or Weiner process follows normal distribution.

Thus the mean number of detected fault is given as

$$E[N(t)]=\begin{cases}\dfrac{a}{(1-\alpha_1)}\left(1-(1+b_1t)e^{-p(1-\alpha_1)b_1t+\frac{1}{2}\sigma^2t}\right) & \text{for } 0\le t\le\tau\\[2mm] \dfrac{a}{(1-\alpha_2)}\left(1-\left(\frac{1+b_2t}{1+b_2\tau}\right)^{p(1-\alpha_2)}(1+b_1\tau)^{p(1-\alpha_1)}e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2t}\right)+\dfrac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\\ \qquad \text{for } t>\tau\end{cases} \quad (11)$$

4.1.3 Proposed SRGM-III

In this proposed model the fault detection rate $b(t)$ may change at any time moment and follows exponential, S-shaped or mix of the two growth curves. And the fault detection rate can be defined as

$$b(t)=\begin{cases}\dfrac{b_1}{1+\beta_1 e^{-b_1 t}} & \text{for } 0\le t\le\tau\\[2mm] \dfrac{b_2}{1+\beta_2 e^{-b_2 t}} & \text{for } t>\tau\end{cases}$$

Here, $\beta_1 and \beta_2$ are the learning parameter before and after the change point respectively. Now again considering the stochastic differential equation (7), the transition probability distribution is:

$$N(t)=\begin{cases}\dfrac{a}{(1-\alpha_1)}\left(1-e^{-p(1-\alpha_1)\int_0^t b_1(x)dx-\sigma w(t)}\right) & \text{for } 0\le t\le\tau\\[2mm] \dfrac{a}{(1-\alpha_2)}\left(1-e^{-p(1-\alpha_1)\int_0^\tau b_1(x)dx-p(1-\alpha_2)\int_\tau^t b_2(x)dt-\sigma w(t)}\right)+\dfrac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\\ \qquad \text{for } t>\tau\end{cases}$$

Now putting the value of $b(t)$ in the above equation, we get

$$N(t)=\begin{cases}\dfrac{a}{(1-\alpha_1)}\left(1-\left(\frac{(1+\beta_1)}{(1+\beta_1e^{-b_1t})}\right)^{p(1-\alpha_1)}e^{-p(1-\alpha_1)b_1t-\sigma w(t)}\right) & \text{for } 0\le t\le\tau\\[2mm] \dfrac{a}{(1-\alpha_2)}\left(1-\left(\frac{(1+\beta_1)}{(1+\beta_1e^{-b_1\tau})}\right)^{p(1-\alpha_1)}\left(\frac{(1+\beta_2e^{-b_2\tau})}{(1+\beta_2e^{-b_2t})}\right)^{p(1-\alpha_2)}e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)-\sigma w(t)}\right)\\ \qquad +\dfrac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & \text{for } t>\tau\end{cases} \quad (12)$$

We consider the mean number of faults detected up to time t. as we know that the Brownian motion or Weiner process follows normal distribution.

So the mean number of detected fault is given as

 ISSN 0973-7529 ISBN 978-93-80544-00-7

$$E[N(t)] = \begin{cases} \frac{a}{(1-\alpha_1)}\left(1-\frac{(1+\beta_1)}{(1+\beta_1 e^{-b_1 t})} e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right) & for\ 0 \le t \le \tau \\ \frac{a}{(1-\alpha_2)}\left(1-\left(\frac{(1+\beta_1)}{(1+\beta_1 e^{-b_1\tau})}\right)^{p(1-\alpha_1)}\left(\frac{(1+\beta_2 e^{-b_2\tau})}{(1+\beta_2 e^{-b_2 t})}\right)^{p(1-\alpha_2)} e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}\right) \\ \qquad +\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau) & for\ t>\tau \end{cases} \tag{13}$$

For the further simplifying the estimation procedure it is assumed that $\alpha_1 = \alpha_2 = \alpha$.

4.2 INSTANTANEOUS MTBF FOR PROPOSED MODELS

Instantaneous MTBF (denoted by $MTBF_I$) is the average time between in an interval dt . The instantaneous mean time between software failures is useful to measure the frequency of software failure occurrence. The instantaneous MTBF for the proposed SRGMs has been given below:

4.2.1 For SRGM-I

$$MTBF_I(t) = \begin{cases} \dfrac{1}{\frac{a}{(1-\alpha_1)}\left[b_1-\frac{1}{2}\sigma^2 e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right]} & for\ 0 \le t \le \tau \\ \dfrac{1}{\frac{a}{(1-\alpha_2)}\left[b_2-\frac{1}{2}\sigma^2 e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\right]} & for\ t>\tau \end{cases}$$

4.2.2 For SRGM-II

$$MTBF_I(t) = \begin{cases} \dfrac{1}{\frac{a}{(1-\alpha_1)}\left[(1+b_1 t)^{p(1-\alpha_1)}\left(\frac{b_1^2 t}{1+bt}-\frac{1}{2}\sigma^2\right)e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right]} & for\ 0 \le t \le \tau \\ \dfrac{1}{\frac{a}{(1-\alpha_2)}\left[\left(\frac{(1+b_2 t)}{(1+b_2\tau)}\right)^{p(1-\alpha_2)}(1+b_1\tau)^{p(1-\alpha_1)}\left(\frac{b_2^2 t}{1+b_2 t}-\frac{1}{2}\sigma^2\right)e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\right]} & for\ t>\tau \end{cases}$$

4.2.3 For SRGM-III

$$MTBF_I(t) = \begin{cases} \dfrac{1}{\frac{a}{(1-\alpha_1)}\left[\left(\frac{(1+\beta_1)}{(1+\beta_1 e^{-b_1 t})}\right)^{p(1-\alpha_1)}\left(\frac{b_1}{1+\beta_1 e^{-b_1 t}}-\frac{1}{2}\sigma^2\right)e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right]} & for\ 0 \le t \le \tau \\ \dfrac{1}{\frac{a}{(1-\alpha_2)}\left[\left(\frac{(1+\beta_1)}{(1+\beta_1 e^{-b_1\tau})}\right)^{p(1-\alpha_1)}\left(\frac{(1+\beta_2 e^{-b_2\tau})}{(1+\beta_2 e^{-b_2 t})}\right)^{p(1-\alpha_2)}\left(\frac{b_2}{(1+\beta_2 e^{-b_2 t})}-\frac{1}{2}\sigma^2\right)e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\right]} & for\ t>\tau \end{cases}$$

4.3 CUMULATIVE MTBF FOR PROPOSED SRGM

The cumulative MTBF is the average time between failures from the beginning of the test (denoted by $MTBF_c$).

The cumulative MTBF for the proposed models has been given below:

4.3.1 For SRGM-I

$$(MTBF^c)_c = \begin{cases} \dfrac{t}{\frac{a}{(1-\alpha_1)}\left(1-e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right)} & for\ 0 \le t \le \tau \\ \dfrac{t}{\frac{a}{(1-\alpha_2)}\left(1-e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}\right)+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)} & for\ t>\tau \end{cases}$$

4.3.2 For SRGM-II

$$(MTBF^c)_c = \begin{cases} \dfrac{t}{\frac{a}{(1-\alpha_1)}\left(1-(1+b_1 t)^{p(1-\alpha_1)}e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right)} & for\ 0 \le t \le \tau \\ \dfrac{t}{\frac{a}{(1-\alpha_2)}\left(\left(1-(1+b_1\tau)^{p(1-\alpha_1)}\left(\frac{1+b_2 t}{1+b_2\tau}\right)^{p(1-\alpha_2)}e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}\right)+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\right)} & for\ t>\tau \end{cases}$$

4.3.3 For SRGM-III

$$(MTBF^c)_c = \begin{cases} \dfrac{t}{\frac{a}{(1-\alpha_1)}\left(1-\left(\frac{(1+\beta_1)}{(1+\beta e^{-b_1 t})}\right)^{p(1-\alpha_1)}e^{-p(1-\alpha_1)b_1 t+\frac{1}{2}\sigma^2 t}\right)} & for\ 0 \le t \le \tau \\ \dfrac{t}{\frac{a}{(1-\alpha_2)}\left(\left(1-\left(\frac{(1+\beta_1)}{(1+\beta_1 e^{-b_1\tau})}\right)^{p(1-\alpha_1)}\left(\frac{(1+\beta_2 e^{-b_2\tau})}{(1+\beta_2 e^{-b_2 t})}\right)^{p(1-\alpha_2)}e^{-p(1-\alpha_1)b_1\tau-p(1-\alpha_2)b_2(t-\tau)+\frac{1}{2}\sigma^2 t}\right)+\frac{(\alpha_1-\alpha_2)}{(1-\alpha_2)}m(\tau)\right)} & for\ t>\tau \end{cases}$$

5.NUMERICAL EXAMPLES AND MODEL EVALUATION.

To check the validity of the proposed model and to find out its software reliability growth, it has been tested using Software SPSS and Software Change Point Analyzer on two Data Sets DS-I and DS-II. Results are compared with existing model.DS-1 is the data cited from Brooks and Motley Brooks [3]. The fault data set is for a radar system of size 124 KLOC (kilo lines of code) tested for 35 months in which 1301 faults were identified. The change-point for this data set is 17th week

The second data set (DS-II) had been collected during 19 weeks of testing a real time command and control in which 328 faults were detected during testing. This data is cited from (Ohba 1984). The change-point of the data is 6th week. The proposed models have been compared with existing model given by (Huan-Jyh Shyur 2003) and the goodness of fit curve is given for the proposed models.

6. CRITERIA FOR COMPARISON

To give quantitative comparisons, some criteria were used to judge the performance of the proposed model. Here we let n represent the sample size of selected data set, m_i represent the actual number of faults by time t_i and $m(t_i)$represent the estimated number of faults by time t_i.in all mentioned criteria the lower value indicate less fitting error .

1. COEFFICIENT OF DETERMINATION R^2

$$R^2 = 1 - \frac{corrected\ ss}{residual\ ss}$$

It measures the per unit value of the total variation about the mean accounted for by the fitted curve. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well. The larger R^2, the better the model explains the variation in the data.

The average of the prediction errors is called the prediction *Bias*, and its standard deviation is often used as a measure of the variation in the predictions.

2. The Mean Square Error (MSE) is defined as:

The difference between the expected values, $\hat{m}(t_i)$ and the observed data y_i is measured by MSE as follows

$$MSE = \sum_{i=1}^{k} \frac{(\hat{m}(t_i) - y_i)^2}{k}$$

Where k is the number of observations. The lower MSE indicates less fitting error, thus better goodness of fit.

Models	a	b_1	b_2	α	p	β_1	β_2	σ
Shyur model	1965	.032	.056	.589	-	-	-	-
Proposed model-1 (eqn-09)	1683	.060	.071	.514	.377	-	-	.018
Proposed model-2 (eqn-11)	1648	.098	.101	.001	.910	-	-	.349
Proposed model-3 (eqn-13)	1333	.123	.243	.001	.724	2.03	26.9	.014

Model Parameter Estimate Result (DS-1)

Models	R^2	MSE
Shyur Model	.956	9852.89
Proposed Model-1 (eqn-09)	.974	5608.09
Proposed Model-2 (eqn-11)	.988	2544.43
Proposed Model-3 (eqn-13)	.999	207.4532

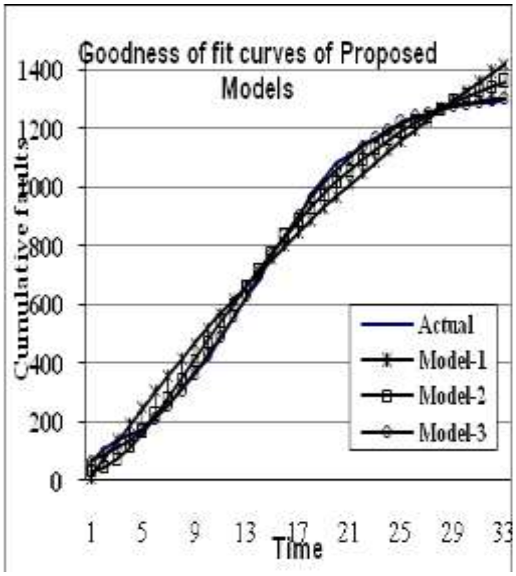
Model Comparison Result (DS-1)

Models	a	b_1	b_2	α	p	β_1	β_2	σ
Shyur model	483	.195	.201	.456	-	-	-	-
Proposed model-1	382	.201	.210	.365	.329	-	-	.022330
Proposed model-2	466	.602	.562	.001	.156	-	-	.000105
Proposed model-3	361	.352	.280	.063	.567	5	3	.000014

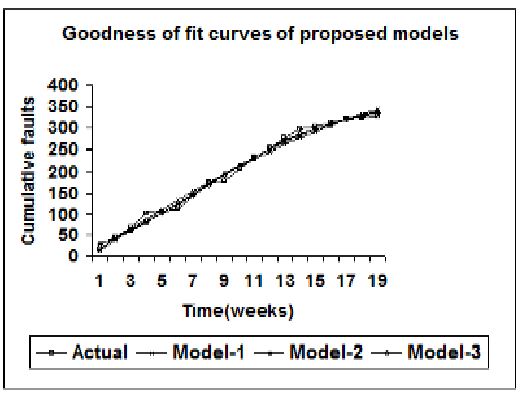
Model Parameter Estimate Results (DS-2)

Models	R^2	MSE
Shyur Model	.979	135.968
Proposed Model-1	.987	120.985
Proposed Model-2	.990	104.906
Proposed Model-3	.992	83.111

Model Comparison Result (DS-II)



For DS-1



For DS-2

CONLUSION

Software reliability models with two types of imperfect debugging and change-point using $It\hat{o}$ type stochastic differential equations have been proposed. The proposed models capture the irregular fluctuation in the fault detection rate. The goodness of fit analysis has been done on real software failure data sets. The results obtained show better fit and wider applicability of the model to different types of failure data sets. Development of software Reliability Growth Models using Stochastic Differential Equations is a new and vibrant emerging area in the field of software reliability engineering. In future, an attempt can be made to develop testing effort functions incorporating irregular fluctuations of testing effort consumptions during detection/removal of faults. Moreover, effort can also be made to develop testing effort dependent software reliability growth models using stochastic differential equations.

REFERENCES

[1]. Huan-Jyh Shyur, “A stochastic software reliability model with imperfect debugging and change -point". the journal of system and software,135-141,66,2003

[2]. Goel, A.L., *Software Reliability Models: Assumptions, Limitations and Applicability,* IEEE Transactions on Software Engineering; SE-11, pp. 1411-1423, 1985.

[3]. P.K.Kapur,V.B Singh and Sameer Anand, “effect of change-point on software reliability growth models using stochastic differential equations”INCRSE-2007,Udaipure,pp220-233,2007.

[4]. Gokhale S.S., Lyu M.R., Trivedi K.S., “Analysis of software fault removal policies using a non-homogenous continuous time Markov chain”, *Software Quality Journal*, vol. 12, pp.211-230, 2004.

[5]. Gokhale S.S., Lyu M.R., Trivedi K.S., “Incorporating fault debugging activities into software reliability models: A simulation approach”, *IEEE Transactions on Reliability*; vol. 55(2), pp.281–292, 2006.

[6]. Jelinski, Z. and P. B. Moranda, *Software Reliability Research,* Freiberger W., *(Ed.)* Statistical Computer Performance Evaluation, New York, Academic Press, pp. 465-497, 1972.

[7]. Kapur P.K., Garg R.B. and Kumar S., “Contributions to hardware and software reliability”, *World Scientific, Singapore*,1999.

[8]. Kapur P.K., Gupta D, Gupta A, Jha P.C., “Effect of Introduction of faults and Imperfect Debugging on Release Time”, To be published , *Ratio Mathematica*.

[9]. Lyu M.R., “Handbook of Software Reliability Engineering”, *McGraw-Hill: New York*, 1996.

[10]. Lo.Jung-Hua, Huang Chin-Yu, “An integration of fault detection and correction processes in software reliability analysis”, *The Journal of Systems and Software*, vol. 79, pp.1312-1323, 2006.

[11]. Mishra PN “ Software reliability analysis”, IBM Systems Journal, 22:262- 270, 1983.

[12]. Musa J.D., Iannino A., Okumoto K., “Software Reliability: Measurement, Prediction, Application”, McGraw-Hill, New York, 1987.

[13]. Ohba M. and Chou, X.M., “*Does Imperfect Debugging Effect Software Reliability Growth*”, proceedings of 11th International Conference of Software Engineering, pp 237-244, 1989.

[14]. Pham H, “System software reliability”, *Springer series in Reliability Engineering*, 2006.

[15]. Pham, H., Nordmann, L. and Zang, X., “*A General Imperfect Software Debugging Model with S-Shaped Fault Detection Rate*”, *IEEE Trans. Rel*., Vol. 48, pp 169-175, 1999.

[16]. Pillai K. and Nair, VSS., “A model for software development effort and cost estimations”, *IEEE Transactions on software engineering*, vol. 23(8), pp. 485-497, 1997.

[17]. Schneidewind, N.F., “Analysis of error processes in computer software”, *Sigplan Notices*, vol. 10, pp.337–346, 1975.

[18]. Schneidewind N.F., “Modelling the fault correction process”, *Proceedings of the 12th International Symposium on Software Reliability Engineering*, IEEE Computer Society Press: Los Alamitos, CA, pp. 185–190, 2001

[19]. Schneidewind, N.F., “Fault correction profiles”, *In Proceedings of the 14th International Symposium on Software Reliability Engineering*, pp. 60–67, 2003.

[20]. Sehgal VK, Kapur Reecha and Kumar Deepak “On The Development of Imperfect Debugging Models " PK Kapur and AK Verma (Ed.) “Quality Reliability and Infocom Technology Trends and Future Directions”. Mc Millan India ., New Delhi, pp. 596-604, 2007.

[21]. Xie M., “ A Study of the Effect of Imperfect debugging on Software Development Cost”, IEEE Transactions on Software Engineering, 29 (5), 2003.

[22]. Xie, M., Zhao, M., “The Schneidewind software reliability model revisited”, *In Proceedings of the 3rd International Symposium on Software Reliability Engineering*, pp. 184–192, May 1992.

[23]. Xie, M., Hu, Q.P., Wu, Y.P. and Ng, S.H, “A study of the Modeling and Analysis of Software Fault-detection and Fault-correction Processes”, *Quality and Reliability Engineering International*, vol. 23, pp.459-470, 2006.

[24]. Xia,G., Zeephongsekul, P., and Kumar, S., “Optimal software release policies for models incorporating learning in testing”, *Asia Pacific Journal of Operational Research*, vol. 9, pp.221-234, 1992.

[25]. Yamada, S., Ohba, M., Osaki, S., “S-shaped reliability growth modeling for software error detection”, *IEEE Transaction on Reliability*, vol. 32, pp.475–478, 1983.

[26]. Yamada S., Tokuno K. and Osaki S., “Imperfect debugging models with fault introduction rate for software reliability assessment”, *Int. J. Syst. Sci*., 23 (12), 1992.

[27]. Oksendal B., “Stochastic Differential Equations-An Introduction with Applications”, Springer, 2003.

[28]. Yamada S., Kimura M., Tanaka H., Osaki S., “Software reliability measurement and assessment with stochastic differential equations”, IEICE Transactions on Fundamentals of Electronics and Computer Sciences, Vol. E77-A, No. 1, pp. 109-116, 1994.

[29]. Yamada S., Nishigaki A., Kimura M., “A Stochastic Differential Equation Model for Software Reliability Assessment and its Goodness of Fit”, International Journal of Reliability and Applications, Vol. 4, No. 1, pp. 1-11, 2003.

[30]. Yamada S., Tamura Y., “A Flexible Stochastic Differential Equation Model in Distributed Development Environment”, European Journal of Operational Research, Vol. 168, pp. 143-152, 2006.

[31]. A.L. Goel and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206–211, 1979.

[32]. A.L. Goel and K. Okumoto, “Time-dependent error-detection rate model for software reliability and other performance measures,” *IEEE Transactions on Reliability*, vol. 28, no. 3, pp. 206–211, 1979.

[33]. Inoue S, A study on Stochastic Modelling for Accurate Software Reliability Assessment, *Ph.D. Thesis*, Doctoral Program of Graduate School of Engineering, Tottori University, Japan, 2006.

[34]. P. K. Kapur and R. B. Garg, “Software reliability growth model for an error-removal phenomenon,” *Software Engineering Journal*, vol. 7, no. 4, pp. 291–294, 1992.

[35]. Yamada S, Ohba M, and Osaki S. (1984) S-shaped software reliability growth models and their applications, IEEE Transactions on Reliability, 1984; R-33: 169-175.

[36]. Sameer Anand, Generalized Framework in software Reliability Modeling, Ph.d Thesis, university of Delhi, delhi, 2010.