International Journal of Automation and Computing 04(4), October 2007, 396-405
DOI: 10.1007/s11633-007-0396-6

Discrete Software Reliability Growth Modeling for Errors of Different Severity Incorporating Change-point Concept

D. N. Goswami[1] Sunil K. Khatri[2,*] Reecha Kapur[3]

[1] School of Studies in Computer Science and Applications, Jiwaji University, Gwalior 474 011, India
[2]Mother Teresa Institute of Management, Guru Gobind Singh Indraprastha University, Delhi, 110 092, India
[3]Department of Mathematics and Computer Applications, Bundelkhand University, Jhansi 284 128, India

Abstract: Several software reliability growth models (SRGM) have been developed to monitor the reliability growth during the testing phase of software development. In most of the existing research available in the literatures, it is considered that a similar testing effort is required on each debugging effort. However, in practice, different types of faults may require different amounts of testing efforts for their detection and removal. Consequently, faults are classified into three categories on the basis of severity: simple, hard and complex. This categorization may be extended to r type of faults on the basis of severity. Although some existing research in the literatures has incorporated this concept that fault removal rate (FRR) is different for different types of faults, they assume that the FRR remains constant during the overall testing period. On the contrary, it has been observed that as testing progresses, FRR changes due to changing testing strategy, skill, environment and personnel resources. In this paper, a general discrete SRGM is proposed for errors of different severity in software systems using the change-point concept. Then, the models are formulated for two particular environments. The models were validated on two real-life data sets. The results show better fit and wider applicability of the proposed models as to different types of failure datasets.

Keywords: Discrete software reliability growth model, non-homogeneous Poisson process, fault severity, change point, probability generating function.

1 Introduction

With the growth in demand for zero defects, predicting reliability of software products is gaining importance. Software reliability models are used to estimate the reliability of a software product. A number of software reliability growth models (SRGM) have been developed in the literature, under different sets of assumptions and testing environments.

SRGMs are generally classified into two groups. The first group contains models, which use the execution time (i.e., CPU time) or calendar time. Such models are called continuous time models[1−4]. The second group contains models, which use the test cases as a unit of fault removal period. Such models are called discrete time models, since the unit of software fault removal period is countable[5−7]. A test case can be a single computer test run executed in an hour, day, week or even month. Therefore, it includes the computer test run and length of time spent to visually inspect the software source code. A large number of models have been developed in the first group while there are fewer in the second group due to the difficulties in terms of mathematical complexity involved.

The utility of discrete SRGM cannot be underestimated. As the software failure data sets are discrete, these models often provide better fit than their continuous time counterparts. Therefore, in spite of difficulties in terms of mathematical complexity involved, discrete models are proposed regularly.

In the last three decades several SRGMs have been developed to estimate the fault content, failure rate and fault removal rate (FRR) per fault in software and to predict the reliability of the software at the release time. Most of these are characterized by the mean value function of a non-homogeneous Poisson process (NHPP) and utilize historical failure data collected during the testing phase to evaluate the quality of the software. Most of these models were proposed under the assumption that similar testing efforts and testing strategy is required for removing each of the faults. However this assumption may not be true in practice. Different faults may require a different amount of testing efforts and testing strategy for their removal from the system. To incorporate this phenomenon, faults are categorized into different types, and are analyzed respectively. Yamada *et al.*[8] proposed a modified exponential SRGM assuming that there are two types of faults in the software and exponential failure curve. Pham[9] proposed an SRGM with multiple failure types. Later, Kapur *et al.*[1] introduced a flexible model called the generalized Erlang SRGM by classifying the faults in the software system as simple, hard and complex faults. It is assumed that the time delay between the failure observation and its subsequent removal represent the severity of faults. The model is extended to r type faults. Therefore, it is desired to study the testing and debugging process of each type of faults separately[2,10]. The mean value function of the SRGM is described by the joint effect of the type of faults present in the system.

It has been assumed in the above models that the FRR remains constant over the entire testing period. However, it is observed that the FRR may not be constant and can change as the testing progresses. Due to the complexity of the software system and the incomplete understanding of the software requirements, specifications and structure, the

Manuscript received January 5, 2007; revised August 8, 2007
*Corresponding author. E-mail address: sunilkkhatri@gmail.com

testing team may not be able to detect the faults at the same rate. As the testing progresses, the FRR changes. The time at which FRR changes is called change-point. There can be multiple change-points in the testing process[4,11−13]. The idea behind the change point concept is that it divides the testing period into intervals and assumes that during a particular interval the testing strategy and testing environment are more or less similar and are slightly different from the other subintervals. The FRR is either assumed to be constant or a function of number of test cases executed during each subinterval but varies from the other subintervals. The concept of change point was started by Zhao[14] who introduced the change-point analysis in hardware and software reliability. Shyur[13] and Kapur *et al.*[3,15] also made their contributions in this area. Shyur[13] has developed an SRGM for multiple types of faults incorporating the concept of change point keeping the FRR constant and different for different types of faults.

In this paper, we propose a general discrete SRGM considering r types of faults on the basis of severity of faults in the software system incorporating the effect of change point. The general framework of the model can be reformulated for specific applications and testing environment with ease. Further, we have formulated two models for the software system developed for critical applications under a specific testing environment. The models are validated on real-life data sets. This paper is an extension to the earlier work done by Kapur *et al.*[4].

This paper is organized as follows. Section 2 presents the model formulation for the proposed models. Section 3 gives the method used for parameter estimation and criteria used for validation and evaluation of the proposed models. The goodness of fit of the proposed models is compared with the discrete version of Yamada′s exponential model[8] with two types of faults and Shyur′s Model[13] for three types of faults with one change point. For comparison purpose, Yamada′s exponential model[8] is modified assuming three types of faults to be in the system. We conclude this paper in Section 4.

Notation

n : Number of test cases executed.

r : Types of faults on the basis of severity.

$m(n)$: Expected number of faults after r test cases during testing phase.

$m_i(n)$: Mean value function of type i faults, $i = 1, 2, \cdots, r$.

τ_i : Change points from where a change in FRR is observed, $i = 1, 2, 3, \cdots, q$.

a : Total fault content.

a_i : Initial fault content of type i faults, $i = 1, 2, 3, \cdots, r$.

p_i : Proportion of type i faults in the software, $i = 1, 2, 3, \cdots, r$.

$b_{ij}(n)$: FRR for a fault type i in j-th interval (each interval corresponding to each change point), $i = 1, 2, 3, \cdots, r$.

2 Software reliability growth modelling

The general assumptions of the model are as follows.

Assumption 1. Failure observation / fault removal phenomenon is modeled by NHPP.

Assumption 2. Each time a failure is observed, an immediate effort takes place to decide the cause of the failure in order to remove it.

Assumption 3. The fault removal process is perfect.

Assumption 4. The delay between the failure observation and its subsequent removal is assumed to represent the severity of faults. The more severe the fault, the more the delay is.

Assumption 5. Software is subject to failures during execution caused by faults remaining in the software.

Assumption 6. During the fault isolation / removal, no new fault is introduced into the system.

Assumption 7. The FRR per remaining fault of each type of fault is different and the rate changes with the change point.

2.1 General framework of model

Faults may be classified as simple, hard and complex depending upon the time between their observation and removal. The more severe the fault, the more the effort and time to remove the fault cost. Here we have modeled the removal phenomenon of the testing and debugging process in terms of Assumption 7. Depending on severity, the faults can be of r types and q number of change-points.

Based on Assumptions 1–7, the differential equation describing the model can be given as

$$\frac{m_i(n+1) - m_i(n)}{\delta} = b_{ij}\left[ap_i - m_i(n)\right] \qquad (1)$$
$$i = 1, 2, \cdots, r, \quad j = 1, 2, \cdots, q$$

where

$$b_{ij} = \begin{cases} b_{i1}(n) & , \ 0 \le n \le \tau_1 \\ b_{i2}(n) & , \ \tau_1 < n \le \tau_2 \\ \cdots & , \quad \cdots \\ b_{iq}(n) & , \ n > \tau_q \end{cases}. \qquad (2)$$

The solution of (1) can be obtained by substituting the forms of the FRR in (2) and defining the number q of change points based on the past data or by experience. The mean value function of the expected total number of faults removed from the system is given as

$$m(n) = \sum_{i=1}^{r} m_i(n). \qquad (3)$$

2.2 Proposed SRGM 1

In some applications, we may expect that the FRR for each type of fault may increase with the number of test cases as the testing team gains experience with the code and learning occurs and reaches a certain constant level towards the end of the testing phase. The FRR of hard and complex faults is less than that of simple fault type and least for complex faults. We may also observe a decreasing FRR towards the end of the testing phase as most of the faults in the software were removed and failure intensity has become much less (see Table 1).

Table 1 FRR with two change points

Interval	FRR		
	Simple faults	Hard faults	Complex faults
$0 \le n \le \tau_1$	b_{11}	$\frac{\left(b_{21}^2 n\right)}{(1+b_{21}n)}$	$\frac{\left(b_{31}^3 n(n+1)/2\right)}{\left(1+b_{31}n+b_{31}^3 n(n+1)/2\right)}$
$\tau_1 < n \le \tau_2$	b_{12}	b_{22}	$\frac{\left(b_{32}^2 n\right)}{(1+b_{32}n)}$
$n > \tau_3$	b_{13}	b_{23}	b_{33}

Here, τ_1 is the number of test cases after which fault detection rate increases due to the expertise or efficiency gained by the present testing team, fault density, introduction of skilled testing personnel, *etc.* τ_2 is the number of test cases in which the testing efficiency gained by the testing team results in the removal of different types of faults with a constant FRR.

2.2.1 Simple faults

For $0 \le n \le \tau_1$,

$$\frac{m_1(n+1)-m_1(n)}{\delta} = b_{11}\left[a_1 - m_1(n)\right]. \quad (4)$$

Solving (4) using probability generating function (PGF) under the initial conditions at $n=0$, $m_1(n)=0$ we get

$$m_1(n) = a_1\left(1-(1-b_{11}\delta)^n\right). \quad (5)$$

For $\tau_1 < n \le \tau_2$,

$$\frac{m_1(n+1)-m_1(n)}{\delta} = b_{12}\left[a_1 - m_1(n)\right]. \quad (6)$$

Solving (6) using PGF under the initial conditions at $n=\tau_1$, $m_1(n)=m_1(\tau_1)$ we get

$$m_1(n) = a_1\left(1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(n-\tau_1)}\right). \quad (7)$$

For $n > \tau_2$,

$$\frac{m_1(n+1)-m_1(n)}{\delta} = b_{13}\left[a_1 - m_1(n)\right]. \quad (8)$$

Solving (8) using PGF under the initial conditions at $n=\tau_2$, $m_1(n)=m_1(\tau_2)$ we get

$$m_1(n) = a_1\left(1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(\tau_2-\tau_1)}\cdot (1-b_{13}\delta)^{(n-\tau_2)}\right). \quad (9)$$

2.2.2 Hard faults

For $0 \le n \le \tau_1$,

$$\frac{m_2(n+1)-m_2(n)}{\delta} = \frac{\left(b_{21}^2 n\right)}{(1+b_{21}n)}\left[a_2 - m_2(n)\right]. \quad (10)$$

Solving (10) using PGF under the initial conditions at $n=0$, $m_2(n)=0$ we get

$$m_2(n) = a_2\left(1-(1+b_{21}n\delta)(1-b_{21}\delta)^n\right). \quad (11)$$

For $\tau_1 < n \le \tau_2$,

$$\frac{m_2(n+1)-m_2(n)}{\delta} = b_{22}\left[a_2 - m_2(n)\right]. \quad (12)$$

Solving (12) using PGF under the initial conditions at $n=\tau_1$, $m_2(n)=m_2(\tau_1)$ we get

$$m_2(n) = a_2\left[1-(1+b_{21}\delta\tau_1)\cdot (1-b_{21}\delta)^{\tau_1}(1-b_{22}\delta)^{(n-\tau_1)}\right]. \quad (13)$$

For $n > \tau_2$,

$$\frac{m_2(n+1)-m_2(n)}{\delta} = b_{23}\left[a_2 - m_2(n)\right]. \quad (14)$$

Solving (14) using PGF under the initial conditions at $n=\tau_2$, $m_2(n)=m_2(\tau_2)$ we get

$$m_2(n) = a_2\left[1-(1+b_{21}\delta\tau_1)(1-b_{21}\delta)^{\tau_1}\cdot (1-b_{22}\delta)^{(\tau_2-\tau_1)}(1-b_{23}\delta)^{(n-\tau_2)}\right]. \quad (15)$$

2.2.3 Complex faults

For $0 \le n \le \tau_1$,

$$\frac{m_3(n+1)-m_3(n)}{\delta} = \frac{\left(\frac{b_{31}^3 n(n+1)}{2}\right)}{\left(1+b_{31}n+\frac{b_{31}^2 n(n+1)}{2}\right)}\cdot \left[a_3 - m_3(n)\right]. \quad (16)$$

Solving (16) using PGF under the initial conditions at $n=0$, $m_3(n)=0$ we get

$$m_3(n) = a_3\left(1-\left(1+b_{31}n\delta+\frac{b_{31}^2 n(n+1)\delta^2}{2}\right)\cdot (1-b_{31}\delta)^n\right). \quad (17)$$

For $\tau_1 < n \le \tau_2$,

$$\frac{m_3(n+1)-m_3(n)}{\delta} = \frac{\left(b_{32}^2 n\right)}{(1+b_{32}n)}\left[a_3 - m_3(n)\right]. \quad (18)$$

Solving (18) using PGF under the initial conditions at $n=\tau_1$, $m_3(n)=m_3(\tau_1)$ we get

$$m_3(n) = a_3\left[1-\left(\frac{1+b_{32}n\delta}{1+b_{32}\tau_1\delta}\right)\cdot \left(1+b_{31}\tau_1\delta+\frac{b_{31}^2\tau_1(\tau_1+1)\delta^2}{2}\right)\cdot (1-b_{31}\delta)^{\tau_1}(1-b_{32}\delta)^{(n-\tau_1)}\right]. \quad (19)$$

For $n > \tau_2$,

$$\frac{m_3(n+1)-m_3(n)}{\delta} = b_{33}\left[a_3 - m_3(n)\right]. \quad (20)$$

Solving (20) using PGF under the initial conditions at $n=\tau_2$, $m_3(n)=m_3(\tau_2)$ we get

$$\begin{aligned} m_3(n) = & a_3\left[1-\left(\frac{1+b_{32}\tau_2\delta}{1+b_{32}\tau_1\delta}\right)\cdot\right. \\ & \left(1+b_{31}\tau_2\delta+\frac{b_{31}^2\tau_1(\tau_1+1)\delta^2}{2}\right)\cdot \\ & (1-b_{31}\delta)^{\tau_1}(1-b_{32}\delta)^{(\tau_2-\tau_1)}\cdot \\ & \left.(1-b_{33}\delta)^{(n-\tau_2)}\right]. \end{aligned} \quad (21)$$

The total fault removal phenomenon of the proposed SRGM is given by the sum of the mean value function of the simple, hard and complex faults.

Thus, the mean value function of superimposed NHPP is

$$m(n)=m_1(n)+m_2(n)+m_3(n) \quad (22)$$

which implies

$$\begin{aligned} m(n) = & a_1\left(1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(\tau_2-\tau_1)}\cdot\right. \\ & \left.(1-b_{13}\delta)^{(n-\tau_2)}\right)+ \\ & a_2\left[1-(1+b_{21}\tau_1\delta)\ (1-b_{21}\delta)^{\tau_1}\cdot\right. \\ & \left.(1-b_{22}\delta)^{(\tau_2-\tau_1)}(1-b_{23}\delta)^{(n-\tau_2)}\right]+ \\ & a_3\left[1-\left(\frac{1+b_{32}\tau_2\delta}{1+b_{32}\tau_1\delta}\right)\left(1+b_{31}\tau_2\delta+\frac{b_{31}^2\tau_1(\tau_1+1)\delta^2}{2}\right)\cdot\right. \\ & \left.(1-b_{31}\delta)^{\tau_1}(1-b_{32}\delta)^{(\tau_2-\tau_1)}(1-b_{33}\delta)^{(n-\tau_2)}\right]. \end{aligned} \quad (23)$$

2.3 Proposed SRGM 2

In critical applications, the testing team starts with a constant FRR for each type of fault. When some critical faults that can severely effect the application are detected, the new testing personnel are added and modifications in testing strategy are made to improve the overall efficiency of the testing. As a result, the FRR decreases. When the testing progresses and the learning occurs, the FRR starts increasing and ultimately reaches a certain constant level towards the end of the testing phase. For simple faults, it is reasonable to assume a constant FRR in each change point interval since not many learning testing strategies are applied for their removals.

Table 2 shows the FRR with three change points. Here we observe that at τ_1, fault detection rate shows a decrease due to the addition of new testing personnel and modifications in testing strategy to improve the efficiency of the testing team. Then at τ_2, fault detection rate starts increasing as the testing progresses due to the learning of the testing team, and ultimately reaching a stable value after τ_3.

Table 2 FRR with three change points

Interval	FRR		
	Simple faults	Hard faults	Complex faults
$0\le n\le\tau_1$	b_{11}	b_{21}	b_{31}
$\tau_1<n\le\tau_2$	b_{12}	$\frac{\left(b_{22}^2 n\right)}{(1+b_{22}n)}$	$\frac{\left(b_{32}^3 n(n+1)/2\right)}{\left(1+b_{32}n+b_{32}^3 n(n+1)/2\right)}$
$\tau_2<n\le\tau_3$	b_{13}	b_{23}	$\frac{\left(b_{33}^2 n\right)}{(1+b_{33}n)}$
$n>\tau_3$	b_{14}	b_{24}	b_{34}

2.3.1 Simple faults

For $0\le n\le\tau_1$,

$$\frac{m_1(n+1)-m_1(n)}{\delta}=b_{11}\left[a_1-m_1(n)\right]. \quad (24)$$

Solving (24) using PGF under the initial conditions at $n=0$, $m_1(n)=0$ we get

$$m_1(n)=a_1\left(1-(1-b_{11}\delta)^n\right). \quad (25)$$

For $\tau_1<n\le\tau_2$,

$$\frac{m_1(n+1)-m_1(n)}{\delta}=b_{12}\left[a_1-m_1(n)\right]. \quad (26)$$

Solving (26) using PGF under the initial conditions at $n=\tau_1$, $m_1(n)=m_1(\tau_1)$ we get

$$m_1(n)=a_1\left(1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(n-\tau_1)}\right). \quad (27)$$

For $\tau_2<n\le\tau_3$,

$$\frac{m_1(n+1)-m_1(n)}{\delta}=b_{13}\left[a_1-m_1(n)\right]. \quad (28)$$

Solving (28) using PGF under the initial conditions at $n=\tau_2$, $m_1(n)=m_1(\tau_2)$ we get

$$\begin{aligned} m_1(n) = & a_1\left(1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(\tau_2-\tau_1)}\cdot\right. \\ & \left.(1-b_{13}\delta)^{(n-\tau_2)}\right). \end{aligned} \quad (29)$$

For $n>\tau_3$,

$$\frac{m_1(n+1)-m_1(n)}{\delta}=b_{14}\left[a_1-m_1(n)\right]. \quad (30)$$

Solving (30) using PGF under the initial conditions at $n=\tau_3$, $m_1(n)=m_1(\tau_3)$ we get

$$\begin{aligned} m_1(n) = & a_1\left[1-(1-b_{11}\delta)^{\tau_1}(1-b_{12}\delta)^{(\tau_2-\tau_1)}\cdot\right. \\ & \left.(1-b_{13}\delta)^{(\tau_3-\tau_2)}(1-b_{14}\delta)^{(n-\tau_3)}\right]. \end{aligned} \quad (31)$$

2.3.2 Hard faults

For $0\le n\le\tau_1$,

$$\frac{m_2(n+1)-m_2(n)}{\delta}=b_{21}\left[a_2-m_2(n)\right]. \quad (32)$$

Solving (32) using PGF under the initial conditions at $n=0$, $m_2(n)=0$ we get

$$m_2(n)=a_2\left(1-(1-b_{21}\delta)^n\right). \tag{33}$$

For $\tau_1<n\le\tau_2$,

$$\frac{m_2(n+1)-m_2(n)}{\delta}=\frac{\left(b_{22}^2 n\right)}{(1+b_{22}n)}\left[a_2-m_2(n)\right]. \tag{34}$$

Solving (34) using PGF under the initial conditions at $n=\tau_1$, $m_2(n)=m_2(\tau_1)$ we get

$$m_2(n)=a_2\left[1-\frac{(1+b_{22}n\delta)}{(1+b_{22}\tau_1\delta)}(1-b_{21}\delta)^{\tau_1}\cdot (1-b_{22}\delta)^{(n-\tau_1)}\right]. \tag{35}$$

For $\tau_2<n\le\tau_3$,

$$\frac{m_2(n+1)-m_2(n)}{\delta}=b_{23}\left[a_2-m_2(n)\right]. \tag{36}$$

Solving (36) using PGF under the initial conditions at $n=\tau_2$, $m_2(n)=m_2(\tau_2)$ we get

$$m_2(n)=a_2\left[1-\frac{(1+b_{22}\tau_2\delta)}{(1+b_{22}\tau_1\delta)}(1-b_{21}\delta)^{\tau_1}\cdot (1-b_{22}\delta)^{(\tau_2-\tau_1)}(1-b_{23}\delta)^{(n-\tau_2)}\right]. \tag{37}$$

For $n>\tau_3$,

$$\frac{m_2(n+1)-m_2(n)}{\delta}=b_{24}\left[a_2-m_2(n)\right]. \tag{38}$$

Solving (38) using PGF under the initial conditions at $n=\tau_3$, $m_2(n)=m_2(\tau_3)$ we get

$$m_2(n)=a_2\left[1-\frac{(1+b_{22}\tau_2\delta)}{(1+b_{22}\tau_1\delta)}(1-b_{21}\delta)^{\tau_1}\cdot (1-b_{22}\delta)^{(\tau_2-\tau_1)}\cdot (1-b_{23}\delta)^{(\tau_3-\tau_2)}(1-b_{24}\delta)^{(n-\tau_3)}\right]. \tag{39}$$

2.3.3 Complex faults

For $0\le n\le\tau_1$,

$$\frac{m_3(n+1)-m_3(n)}{\delta}=b_{31}\left[a_3-m_3(n)\right]. \tag{40}$$

Solving (40) using PGF under the initial conditions at $n=0$, $m_3(n)=0$ we get

$$m_3(n)=a_3\left(1-(1-b_{31}\delta)^n\right). \tag{41}$$

For $\tau_1<n\le\tau_2$,

$$\frac{m_3(n+1)-m_3(n)}{\delta}=\frac{\left(\frac{b_{32}^3 n(n+1)}{2}\right)}{\left(1+b_{32}n+\frac{b_{32}^2 n(n+1)}{2}\right)}\cdot \left[a_3-m_3(n)\right]. \tag{42}$$

Solving (42) using PGF under the initial conditions at $t=\tau_1$, $m_3(t)=m_3(\tau_1)$ we get

$$m_3(n)=a_3\left[1-\frac{\left(1+b_{32}n\delta+\frac{b_{32}^2 n(n+1)\delta^2}{2}\right)}{\left(1+b_{32}\tau_1\delta+\frac{b_{32}^2\tau_1(\tau_1+1)\delta^2}{2}\right)}\cdot (1-b_{31}\delta)^{\tau_1}(1-b_{32}\delta)^{(n-\tau_1)}\right]. \tag{43}$$

For $\tau_2<n\le\tau_3$,

$$\frac{m_3(n+1)-m_3(n)}{\delta}=\frac{\left(b_{33}^2 n\right)}{(1+b_{33}n)}\left[a_3-m_3(n)\right]. \tag{44}$$

Solving (44) using PGF under the initial conditions at $n=\tau_2$, $m_3(n)=m_3(\tau_2)$ we get

$$m_3(n)=a_3\left[1-\left(\frac{1+b_{32}\tau_2\delta+\frac{b_{32}^2\tau_2(\tau_2+1)\delta^2}{2}}{1+b_{32}\tau_1\delta+\frac{b_{32}^2\tau_1(\tau_1+1)\delta^2}{2}}\right)\cdot \left(\frac{1+b_{33}n\delta}{1+b_{33}\tau_2\delta}\right)\cdot (1-b_{31}\delta)^{\tau_1}(1-b_{32}\delta)^{(\tau_2-\tau_1)}\cdot (1-b_{33}\delta)^{(n-\tau_2)}\right]. \tag{45}$$

For $n>\tau_3$,

$$\frac{m_3(n+1)-m_3(n)}{\delta}=b_{34}\left[a_2-m_2(n)\right]. \tag{46}$$

Solving (46) using PGF under the initial conditions at $n=\tau_3$, $m_3(n)=m_3(\tau_3)$ we get

$$m_3(n)=a_3\left[1-\left(\frac{1+b_{32}\tau_2\delta+\frac{b_{32}^2\tau_2(\tau_2+1)\delta^2}{2}}{1+b_{32}\tau_1\delta+\frac{b_{32}^2\tau_1(\tau_1+1)\delta^2}{2}}\right)\cdot \left(\frac{1+b_{33}\tau_3\delta}{1+b_{33}\tau_2\delta}\right)(1-b_{31}\delta)^{\tau_1}\cdot (1-b_{32}\delta)^{(\tau_2-\tau_1)}\cdot (1-b_{33}\delta)^{(\tau_3-\tau_2)}(1-b_{34}\delta)^{(n-\tau_3)}\right]. \tag{47}$$

The total fault removal phenomenon of the proposed SRGM 2 is given by the sum of the mean value function of the simple, hard and complex faults. Thus, the mean value function of superimposed NHPP is

$$m(n)=m_1(n)+m_2(n)+m_3(n)$$

which implies

$$\begin{aligned} m(n) =& a_1 \Big[1 - (1-b_{11}\delta)^{\tau_1} (1-b_{12}\delta)^{(\tau_2-\tau_1)} \cdot \\ & (1-b_{13}\delta)^{(\tau_3-\tau_2)} (1-b_{14}\delta)^{(n-\tau_3)}\Big] + \\ & a_2 \Big[1 - \frac{(1+b_{22}\tau_2\delta)}{(1+b_{22}\tau_1\delta)} (1-b_{21}\delta)^{\tau_1} (1-b_{22}\delta)^{(\tau_2-\tau_1)} \cdot \\ & (1-b_{23}\delta)^{(\tau_3-\tau_2)} (1-b_{24}\delta)^{(n-\tau_3)}\Big] + \\ & a_3 \Bigg[1 - \left(\frac{1+b_{32}\tau_2\delta + \frac{b_{32}^2\tau_2(\tau_2+1)\delta^2}{2}}{1+b_{32}\tau_1\delta + \frac{b_{32}^2\tau_1(\tau_1+1)\delta^2}{2}}\right) \cdot \\ & \left(\frac{1+b_{33}\tau_3\delta}{1+b_{33}\tau_2\delta}\right) (1-b_{31}\delta)^{\tau_1} (1-b_{32}\delta)^{(\tau_2-\tau_1)} \cdot \\ & (1-b_{33}\delta)^{(\tau_3-\tau_2)} (1-b_{34}\delta)^{(n-\tau_3)}\Bigg]. \end{aligned} \tag{48}$$

In (23) and (48),

$$a_1 = ap_1\,, \quad a_2 = ap_2\,, \quad a_3 = ap_3. \tag{49}$$

Continuous time models for (23) and (48) are derived in Appendix.

3 Parameter estimation

Parameter estimation and model validation is an important aspect of modeling. The mathematical equations of the proposed SRGM are non-linear. Technically, it is more difficult to find the solution for non-linear models using least square method and requires numerical algorithms to solve. Statistical software packages such as statistical package for social sciences (SPSS) help to overcome this problem. For the estimation of the parameters of the proposed model, the non-linear regression method of SPSS has been used. Non-linear regression is a method of finding a nonlinear model of the relationship between the dependent variable and a set of independent variables. Unlike traditional linear regression, which is restricted to estimating linear models, non-linear regression can estimate models with arbitrary relationships between independent and dependent variables.

3.1 Comparison criteria for SRGM

The performance of SRGM is judged by their ability to fit the past software fault data (goodness of fit).

3.1.1 Goodness of fit criteria

Goodness of fit denotes how good a mathematical model (for example a linear regression model) fits the data.

1) The mean square fitting error (MSE)

The models under comparison are used to simulate the fault data, the difference between the expected values, $\hat{m}(n_i)$ and the observed data y_i is measured by MSE[6] as follows:

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

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

2) Coefficient of multiple determination R^2

We define this coefficient as the ratio of the sum of squares (SS) resulting from the trend model to that from the constant model subtracted from 1[9], i.e.,

$$R^2 = 1 - \frac{\text{Residual SS}}{\text{Corrected SS}}.$$

R^2 measures the percentage of the total variation about the mean accounted for 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 is, the better the model explains the variation in the data.

3) Prediction error (PE)

The difference between the observation and prediction of number of failures at any instant of time i is known as PE_i The lower the value of PE is, the better the goodness of fit is[12].

4) Bias

The average of PEs is known as Bias. The lower the value of Bias is, the better the goodness of fit is[12].

5) Variation

The standard deviation of PE is known as variation.

$$\text{Variation} = \sqrt{\frac{1}{N-1}\sum (\text{PE}_i - \text{Bias})^2}$$

The lower the value of Variation is, the better the goodness of fit is[12].

6) Root mean square prediction error (RMSPE)

It is a measure of closeness with which a model predicts the observation.

$$\text{RMSPE} = \sqrt{(\text{Bias}^2 + \text{Variation}^2)}$$

The lower the value of RMSPE is, the better the goodness of fit is[12].

3.2 Model validation and data description

To check the validity of the proposed SRGM, it was tested on two data sets. The Proposed SRGM was compared with a discrete version of Yamada′s modified exponential model[8] and Shyur′s model[13]. For comparison purposes, we have recomputed the mean value function of the removal phenomenon of the discrete version of Yamada′s modified exponential model[8] assuming three types of faults in the system. The recomputed mean value function for the model is given by

$$m(n) = \sum_{i=1}^{3} m_i(n) = \sum_{i=1}^{3} ap_i(1-(1-b_i)^n) \tag{50}$$

where b_i $(i = 1, 2, 3)$ are FRR for the simple, hard and complex faults. The Shyur′s model[13] uses the mean value function (MVF) in (50) with one change point. Here b_i $(i = 1, 3, 5)$ are the FRR before the change point and b_i $(i = 2, 4, 6)$ are the FRR after the change point. Since the number of unknown parameters is sixteen in the proposed models, therefore to yield better estimates we assume $b_{11} = b_{12} = b_{13} = b_{14} = b_1$, $b_{21} = b_{22} = b_{23} = b_{24} = b_2$, and $b_{31} = b_{32} = b_{33} = b_{34} = b_3$. The FRR for each type of faults in each interval are given in Tables 3 and 4.

Table 3 FRR with two change points

Interval	FRR Simple faults	Hard faults	Complex faults
$0 \le n \le \tau_1$	b_1	$\dfrac{\left(b_2^2 n\right)}{(1+b_2 n)}$	$\dfrac{\left(b_3^3 n(n+1)/2\right)}{\left(1+b_3 n+b_3^3 n(n+1)/2\right)}$
$\tau_1 < n \le \tau_2$	b_1	b_2	$\dfrac{\left(b_3^2 n\right)}{(1+b_3 n)}$
$n > \tau_3$	b_1	b_2	b_3

Table 4 FRR with three change points

Interval	FRR Simple faults	Hard faults	Complex faults
$0 \le n \le \tau_1$	b_1	b_2	b_3
$\tau_1 < n \le \tau_2$	b_1	$\dfrac{\left(b_2^2 n\right)}{(1+b_2 n)}$	$\dfrac{\left(b_3^3 n(n+1)/2\right)}{\left(1+b_3 n+b_3^3 n(n+1)/2\right)}$
$\tau_2 < n \le \tau_3$	b_1	b_2	$\dfrac{\left(b_3^2 n\right)}{(1+b_3 n)}$
$n > \tau_3$	b_1	b_2	b_3

In Shyur′s model[13], the FRR before and after the change point are all taken to be constant for each type of faults, whereas in the proposed SRGM, and FRR may change with respect to number of test cases for each type of fault in each change point interval. The proposed SRGM provides better goodness of fit for both the datasets due to its applicability and flexibility. However, the increased accuracy achieved shows the capability of the model to capture different types of failure datasets. Two real time data sets DS 1 and DS 2 are used for estimation.

DS 1. This data is cited from an online communication system (OCS) project at ABC software company[16]. The data was collected over a period of 12 weeks during which 136 faults were removed. The parameter estimation result and the goodness of fit results for the proposed SRGM are given in Tables 5 and 6. The goodness of fit curves for DS 1 are given in Figs. 1 - 4. In this dataset we have taken $\tau_1 = 3$ and $\tau_2 = 5$ for the proposed SRGM 1, and $\tau_1 = 3$, $\tau_2 = 5$, $\tau_3 = 9$ for the proposed SRGM 2. The values of p_1, p_2 and p_3 are computed from the actual data set since data was available separately for each type of faults on the basis of severity.

DS 2. This data is cited from [11]. The software was tested for 38 weeks during which 2 456.4 computer hours were used and 231 faults were removed. The parameter estimation result and the goodness of fit results for the proposed SRGM are given in Tables 7 and 8. The goodness of fit curves for DS 2 are given in Figs. 5 - 8. In this dataset we have taken $\tau_1 = 18$, $\tau_2 = 36$ for the proposed SRGM 1, and $\tau_1 = 10$, $\tau_2 = 18$, $\tau_3 = 36$ for the proposed SRGM 2. The values of p_1, p_2 and p_3 are computed from the actual data set since data was available separately for each type of faults on the basis of severity.

It is evidently seen from the tables that the proposed SRGM fits better than discrete version of both Yamada′s modified exponential SRGM (50) and Shyur′s model[13]. The proposed SRGM 2 gives better results than the proposed SRGM 1 because of the flexibility in curve in capturing the relevant actual data points.

In Figs. 1 - 8, model 1 indicates the discrete version in (50), model 2 indicates discrete version of Shyur′s model[13], model 3 indicates proposed SRGM 1, and model 4 indicates the proposed SRGM 2.

Table 5 Parameter estimation for DS 1

Model	Parameter estimation a	b_1	b_2	b_3	b_4	b_5	b_6	p_1	p_2	p_3
Yamada SRGM (50)	153	0.10	0.10	0.18	-	-	-	0.41	0.40	0.19
Shyur SRGM[13]	148	0.06	0.11	0.09	0.17	0.83	0.28	0.41	0.40	0.19
Proposed SRGM 1	185	0.06	0.13	0.99	-	-	-	0.41	0.40	0.19
Proposed SRGM 2	155	0.16	0.18	0.09	-	-	-	0.41	0.40	0.19

Table 6 Goodness of fit metrics criterion for DS 1

Model	Comparison criterion R^2	MSE	Bias	Variation	RMSPE
Yamada SRGM (50)	0.942	391.1	-5.7	208.9	209.1
Shyur SRGM[13]	0.970	352.5	-14.5	153.9	154.5
Proposed SRGM 1	0.989	158.7	3.4	160.7	160.7
Proposed SRGM 2	0.993	134.6	-3.2	135.4	135.5

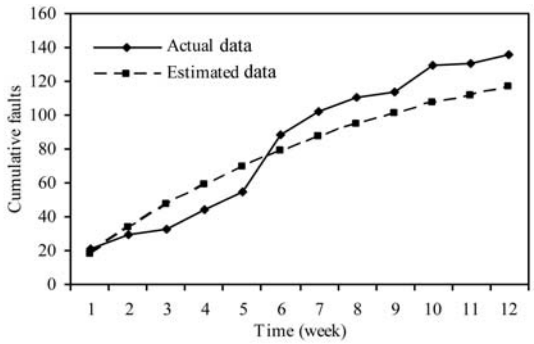

Fig. 1 Goodness of fit of model 1 on DS 1

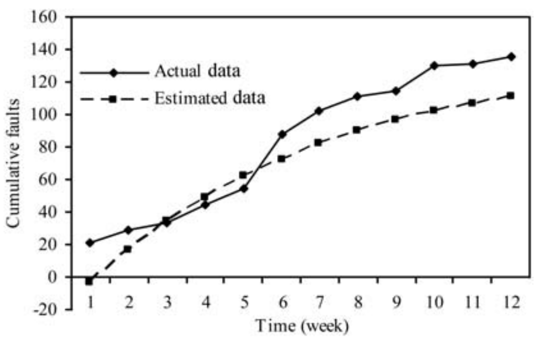

Fig. 2 Goodness of fit of model 2 on DS 1

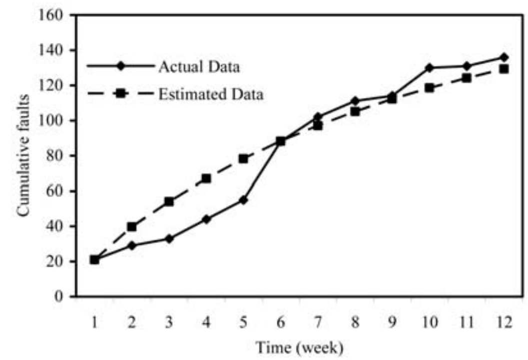

Fig. 3 Goodness of fit of model 3 on DS 1

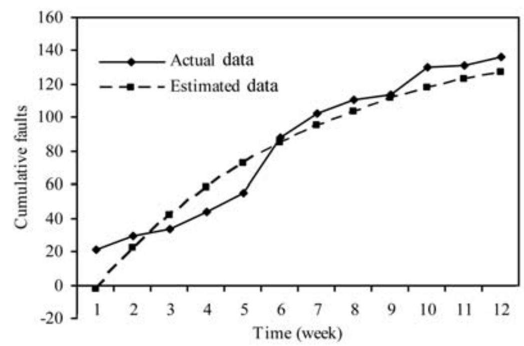

Fig. 4 Goodness of fit of model 4 on DS 1

Table 7 Parameter estimation for DS 2

Model	Parameter estimation									
	a	b_1	b_2	b_3	b_4	b_5	b_6	p_1	p_2	p_3
Yamada SRGM (50)	235	0.04	0.03	0.39	-	-	-	0.64	0.34	0.02
Shyur SRGM[13]	236	0.79	0.49	0.02	0.05	0.01	0.05	0.64	0.34	0.02
Proposed SRGM 1	283	0.06	0.02	0.01	-	-	-	0.64	0.34	0.02
Proposed SRGM 2	274	0.04	0.03	0.01	-	-	-	0.64	0.34	0.02

Table 8 Goodness of fit metrics criterion for DS 2

Model	Comparison criterion				
	R^2	MSE	Bias	Variation	RMSPE
Yamada SRGM (50)	0.948	192.3	-2.4	191.5	191.5
Shyur SRGM[13]	0.962	140.6	-1.3	142.8	142.8
Proposed SRGM 1	0.955	166.5	-2.9	161.9	161.9
Proposed SRGM 2	0.975	94.6	-1.7	94.1	94.2

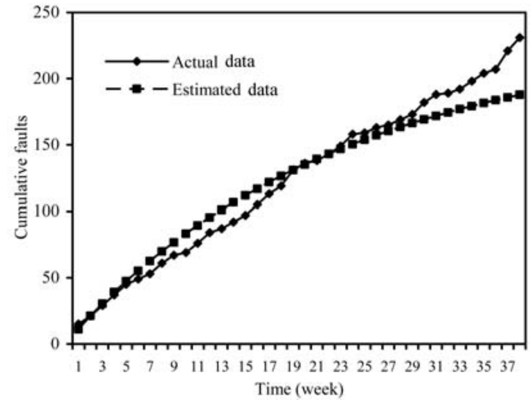

Fig. 5 Goodness of fit of model 1 on DS 2

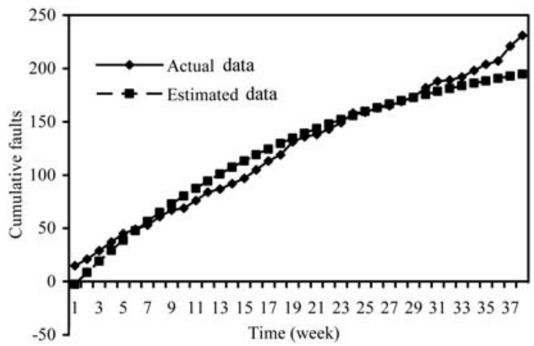

Fig. 6 Goodness of fit of model 2 on DS 2

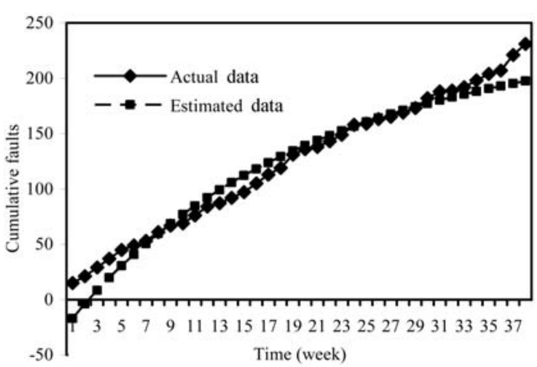

Fig. 7 Goodness of fit of model 3 on DS 2

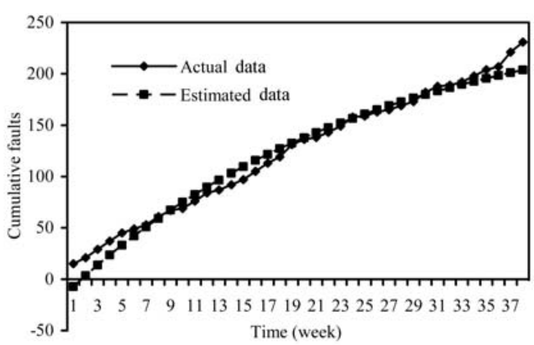

Fig. 8 Goodness of fit of model 4 on DS 2

4 Conclusion

In this paper, we proposes SRGM defining errors of different severity using the change point concept. The goodness of fit of the proposed SRGM are compared with a discrete version of Yamada′s modified exponential model (50) and Shyur′s model[13]. The results obtained show better fit and wider applicability of the proposed models to different types of failure datasets. The model will be more valuable under a higher level of accuracy. From the numerical illustrations, we see that the models provide considerably improved results. Applications of the models are shown for two particular environments. The models can be modified according to the testing environment. Moreover, the proposed SRGM with suitable changes can be applied to the distributed environment and will be introduced in the future.

Appendix

The derivation of continuous SRGM from discrete SRGM is presented as follows.

Let us define $t = n\delta$. If $n \to \infty$, then $\delta \to 0$.

Let

$$(1-\delta b)^n = (1-\delta b)^{t/\delta} \to \mathrm{e}^{-bt}, \quad \delta \to 0 \tag{A1}$$

and

$$(1+\delta nb) = (1+bt)\,. \tag{A2}$$

Using (A1) and (A2), the limit of the right-hand side of (23) as δ tends to 0 is the right-hand side of the following equation

$$\begin{aligned} m\,(t) =& a_1\left(1-\mathrm{e}^{-b_{11}\tau_1-b_{12}(\tau_2-\tau_1)-b_{13}(t-\tau_2)}\right)+\\ & a_2\left[1-(1+b_{21}\tau_1)\ \mathrm{e}^{-b_{21}\tau_1-b_{22}(\tau_2-\tau_1)-b_{23}(t-\tau_2)}\right]+\\ & a_3\left[1-\left(\frac{1+b_{32}\tau_2}{1+b_{32}\tau_1}\right)\left(1+b_{31}\tau_2+\frac{b_{31}^2\tau_1^2}{2}\right)\cdot\right.\\ & \left.\mathrm{e}^{-b_{31}\tau_1-b_{32}(\tau_2-\tau_1)-b_{33}(t-\tau_2)}\right]. \end{aligned} \tag{A3}$$

The limit of the right-hand side of (48) as δ tends to 0 is the right-hand side of the following equation

$$\begin{aligned} m(t) =& a_1\left(1-\mathrm{e}^{-b_{11}\tau_1-b_{12}(\tau_2-\tau_1)-b_{13}(\tau_3-\tau_2)-b_{14}(t-\tau_3)}\right)+\\ & a_2\left[1-\left(\frac{1+b_{22}\tau_2}{1+b_{22}\tau_1}\right)\cdot\right.\\ & \left.\mathrm{e}^{-b_{21}\tau_1-b_{22}(\tau_2-\tau_1)-b_{23}(\tau_3-\tau_2)-b_{24}(t-\tau_3)}\right]+\\ & a_3\left[1-\left(\frac{1+b_{33}\tau_3}{1+b_{33}\tau_2}\right)\left(\frac{1+b_{32}\tau_2+\frac{b_{32}^2\tau_2^2}{2}}{1+b_{32}\tau_1+\frac{b_{32}^2\tau_1^2}{2}}\right)\cdot\right.\\ & \left.\mathrm{e}^{-b_{31}\tau_1-b_{32}(\tau_2-\tau_1)-b_{33}(\tau_3-\tau_2)-b_{34}(t-\tau_3)}\right]. \end{aligned} \tag{A4}$$

References

[1] P. K. Kapur, S. Younes, S. Agarwala. Generalised Erlang Model with N types of faults. *ASOR Bulletin*, vol. 14, no. 1, pp. 5–11, 1995.

[2] P. K. Kapur, A. K. Bardhan, O. Shatnawi. Why Software Reliability Growth Modelling Should Define Errors of Different Severity. *Journal of the Indian Statistical Association*, vol. 40, no. 2, pp. 119–142, 2002.

[3] P. K. Kapur, V. B. Singh, S. Anand, V. S. S. Yadavalli. Software Reliability Growth Model with Change Point and Effort Control Using a Power Function of the Testing Time. *International Journal of Production Research*, [online], Available: http://www.informaworld.com, November 17, 2006.

[4] P. K. Kapur, A. Kumar, K. Yadav, S. Khatri. Software Reliability Growth Modelling for Errors of Different Severity Using Change Point. *International Journal of Reliability, Quality and Safety Engineering*, to be published.

[5] P. K. Kapur, O. Shatnawi, O. Singh. Discrete Imperfect Software Reliability Growth Models under Imperfect Debugging Environment. In *Proceedings of the International Conference on Multimedia and Design*, N. J. Rajaram, A. K. Verma (eds), Arena Multimedia & IIT-Mumbai, Mumbai, vol. 2, pp. 114–129, 2002.

[6] P. K. Kapur, R. B. Garg, S. Kumar. *Contributions to Hardware and Software Reliability*, World Scientific, Singapore, 1999.

[7] S. Yamada, S. Osaki. Discrete Software Reliability Growth Models. *Journal of Applied Stochastic Models and Data Analysis*, vol. 1, no. 1, pp. 65–67, 1985.

[8] S. Yamada, S. Osaki, H. Narithisa. A Software Reliability Growth Model with Two Types of Errors. *RAIRO Operations Research*, vol. 19, no. 1, pp. 87–104, 1985.

[9] H. Pham. Software Reliability Assessment: Imperfect Debugging and Multiple Failure Types in Software Development, Technical Report EG&G-RAMM-10737, Idaho National Engineering Laboratory, 1993.

[10] P. K. Kapur, S. Younes, P. S. Grover. Software Reliability Growth Model with Errors of Different Severity. *Computer Science and Informatics*, Indian, vol. 25, no. 3, pp. 51–65, 1995.

[11] P. N. Misra. Software Reliability Analysis. *IBM System Journal*, vol. 22, no. 3, pp. 262–270, 1983.

[12] K. Pillai, V. S. S. Nair. A Model for Software Development Effort and Cost Estimation. *IEEE Transactions on Software Engineering*, vol. 23, no. 8, pp. 485–497, 1997.

[13] H. J. Shyur. A Stochastic Software Reliability Model with Imperfect-debugging and Change-point. *Journal of Systems and Software*, vol. 66, no. 2, pp. 135–141, 2003.

[14] M. Zhao. Change-point Problems in Software and Hardware Reliability. *Communications in Statistics Theory and Methods*, vol. 22, no. 3, pp. 757–768, 1993.

[15] P. K. Kapur, A. Gupta, O. Shatnawi, V. S. S. Yadavalli. Testing Effort Control Using Flexible Software Reliability Growth Model with Change Point. *International Journal of Performability Engineering*, vol. 2, no. 3, pp. 245–263, 2006.

[16] H. Pham. *System Software Reliability*, *Springer Series in Reliability Engineering*, Springer, London, 2006.

D. N. Goswami received the M.Sc. and Ph.D. degrees in computer applications in 1989 and 2004, respectively. He is currently working as reader and head of School of Studies in Computer Science and Applications, Jiwaji University, Gwalior. He has been into teaching for 15 years, and published a number of papers in national and international journals.

His research interests include software reliability and network security.

Sunil K. Khatri received the Master degree in computer applications from Rani Durgavati University, Jabalpur in 1999, and is a Ph.D. candidate in software reliability of Jiwaji University, Gwalior. His Ph.D. research area is discrete software reliability growth modeling. He is a senior lecturer at Department of Computer Science, Mother Teresa Institute of Management, GGSIP University. He has been into teaching for 8 years.

His research interests include software engineering, data mining, and computer network security.

Reecha Kapur is a research scholar at Department of Mathematics and Computer Application, Bundelkhand University, Jhansi, India. She has done her Post Graduation in mathematics from Bundelkhand University, Jhansi, India. She has published three research papers.

Her research interests include imperfect debugging models in software reliability and its effect on software testing cost.