Chapter-2

Software Reliability Growth modeling incorporating errors of different severity and change point

The development of high quality software satisfying cost, schedule, and resource requirements is an essential prerequisite for improved competitiveness of any organization. One major difficulty to master this challenge is the inevitability of defects in software products during its use in operational phase. It is the testing phase of the software that aims to remove most of the faults lying in software. The testing of software systems is subject to strong conflicting forces. Therefore the software developer attempts to have a tight control over the testing of the software. One of the most effective ways to do this is to apply SRE tools and techniques to testing and development.

In the last two decades several Software Reliability models have been developed in the literature to estimate the fault content, failure rate and fault removal rate 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 NHPP and utilize historical failure data collected during the testing phase to evaluate the quality of software. Goel and Okumoto (1976) have proposed first time dependent NHPP based SRGM assuming that the failure intensity is proportional to the number of faults remaining in the software. The model is very simple and describes exponential failure curves. In general most of the existing SRGM describes either an exponential or s-shaped failure curves. Ohba (1984) refined the Goel-Okumoto model by assuming that the fault detection \ removal

This chapter is based on the following paper:

rate increases with time and that there are two types of faults in the software. SRGM proposed by Bittanti et al. (1988) and Kapur and Garg (1992) have similar forms as that of Ohba (1984) but are developed under different set of assumptions. Bittanti et al. (1988) proposed an SRGM exploiting the fault removal (exposure) rate during the initial and final time epochs of testing. Whereas, Kapur and Garg (1992) describe a fault removal phenomenon, where they assume that during a removal process of a fault some of the additional faults might be removed without these faults causing any failure. These models can describe both exponential and S-shaped growth curves and therefore are termed as flexible models (Bittanti (1988), Kapur and Garg (1992), Obha (1984)).

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 different amount of testing efforts and testing strategy for their removal from the system. In literature to incorporate this phenomenon faults are categorized as of different types and are analyzed separately. Yamada (1985) proposed a modified exponential SRGM assuming that there are two types of faults in the software and exponential failure curve. Pham et al (1993) proposed a SRGM with multiple failure types. Later Kapur et al (1995) 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 n-types of faults. Another model due to Kapur et al (2000) describes the implicit categorization of faults based on the time of detection of fault. However an SRGM should explicitly define the errors of different severity as it is expected that any type of fault can be detected at any point of testing time. Therefore it is desired to study the testing and debugging process of each type of faults separately (Kapur et al 1995, 2002). The mean value function of the SRGM is described by the joint effect of the type of faults present in the system. Such an approach can capture the variability in the reliability growth curve due to the errors of different severity depending on the testing environment.

The models discussed were derived under the assumption that the fault detection and/or removal rate remains constant over the entire testing period. But during the period of
testing, it is observed that the FDR or FRR may not be constant and can change as the testing progresses. The changes in the FDR or FRR can be accounted due to the changes in the testing environment, testing strategy, complexity and size of the functions under testing, skill, motivation and constitution of the testing and debugging team etc. The change in FDR or FRR can be analyzed using the "change point concept". The idea behind the change point concept is that it divides the testing period into subintervals and assumes that during a particular subinterval the testing strategy and testing environment are more or less similar and are slightly different from the other subintervals. The FDR and/or FRR is either assumed to be constant or a function of testing time during each subinterval but varies (constant but different or a different relation to time) from the other subintervals. The concept of change point was started by Zhao (1993) who introduced the change-point analysis in Hardware and Software reliability. Shyur (2003), Kapur et. al (2006a,2006b) also made their contributions in this area. Shyur (2003) has developed an SRGM for multiple types of faults incorporating the concept of change point keeping the FDR constant and different for different types of faults.

In this chapter we propose a general software reliability growth model considering three types of faults (simple, hard and complex) in the software system incorporating the effect of changing fault-debugging rate using the change point concept. The general framework of the model can be reformulated for specific applications and testing environment with ease. Further we have formulated the model for the software system developed for critical application under a specific testing environment.

2.1 Incorporating errors of different severity and change point in software reliability growth modeling

All faults lying in software are not similar. Some of them may be critical from the users point of view i.e. failure due to these faults results in total system failure while others may be critical from the developers point of view in the sense that large amount of testing resources are required for their removal as they are difficult to remove due to complexity of the underlying code. In this chapter we have described the severity of faults from the developers view point. Faults are categorized with respect to time they
take for isolation and removal after their observation. Faults are classified as “simple” if the time between their observation and removal is negligible else if more efforts and time is required for the removal the fault is classified as “hard fault” and if on the detection of a fault the amount of effort and time required to remove it is much more as compared to hard faults, the faults are classified as complex fault (Kapur and Garg, 1999). Applying this type of model to the initial failure data (or past data) gives an idea about the nature of remaining faults and their respective proportions. If this information is made available to the test manager, testing effort can be streamlined to achieve better fault detection and appropriate control can be initiated for a particular category (Kapur and Garg 1999, Musa 1990).

Here we model the removal phenomenon of the testing and debugging process assuming that the fault removal rate per remaining fault of the each type of fault is different and the rate changes with the change point. The general assumptions and notations of the model are as follows:

2.1.1 Assumptions

1. Failure observation / fault removal phenomenon is modeled by NHPP.
2. Software is subject to failures during execution caused by faults remaining in the software.
3. Each time a failure is observed, an immediate effort takes place to decide the cause of the failure in order to remove it.
4. The time delay between the failure observation and its subsequent removal is assumed to represent the severity of faults. The more severe the fault, more the time delay.
5. During the fault isolation / removal, no new fault is introduced into the system.
6. The fault removal process is perfect.
7. The fault removal rate per remaining fault of the each type of fault is different and the rate changes with the change point.
2.1.2 Notations

- $m(t)$: Expected number of faults identified in the time interval $(0,t]$ during testing phase
- $\tau_i$: Change Points (time from where a change in FRR is observed, $i = 1,2,3$)
- $a$: Total fault content
- $a_i$: Initial fault content of type $i$ faults (simple, hard and complex) $i = 1,2,3$
- $b_{ij}(t)$: Fault removal rate for a fault type $i$ in $j$th time interval (each time interval corresponding to each change point), $i = 1,2,3; j = 1,2,…,n$
- $m_i(t)$: Mean value function of type $i$ faults (simple, hard and complex) $i = 1,2,3$
- $p_i$: Proportion of type $i$ faults in the software’s, $i = 1,2,3$

The differential equation describing the model can be given as

$$m_i'(t) = b_{ij} \left[ ap_j - m_i(t) \right] \quad i = 1,2,3; j = 1,2,…,n$$

(2.1)

Where

$$b_{ij} = \begin{cases} 
  b_{i1}(t) & 0 \leq t \leq \tau_1 \\
  b_{i2}(t) & \tau_1 < t \leq \tau_2 \\
  \vdots \\
  b_{in}(t) & t > \tau_n 
\end{cases}$$

(2.2)

The exact solution of the above model equations (2.1) can be obtained on substituting the functional forms of the FRR in (2.2) and defining the number of change points based on past data or by the experience. Now the mean value function of the expected total number of faults removed from the system is given as

$$m(t) = \sum_{i=1}^{3} m_i(t)$$

(2.3)

Various diverse testing environment and testing strategies existing for different type of software can be analyzed from the above model by choosing the appropriate forms of the fault removal rates $b_{ij}(t)$ (based on the past failure data and experience of the developer). One of the most simple and general case would be the one if we consider
FRR for each type of fault in each change point interval to be constant but distinct for each $i$ and $j$ if we observe exponential failure curve growth pattern for each type of fault. However in case of a general purpose software we may expect that the FRR for each type of fault may increase with time 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 fault detection rate of hard and/or complex faults is slightly less than that of simple fault type. We may also observe a decreasing FRR towards the end of testing phase since most of the faults lying in the software are removed and failure intensity has become very less (see table 2.1).

**Table 2.1: Severity of faults with two Points**

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Simple</th>
<th>Hard</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Interval</td>
<td>Fault Detection Rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0 \leq t \leq \tau_1$</td>
<td>$b_{11}$</td>
<td>$\left(b_{21}^2 t\right)/\left(1+b_{21} t\right)$</td>
<td>$\left(b_{31}^3 t^2\right)/\left(1+b_{31} t+b_{31}^3 t^2/2\right)$</td>
</tr>
<tr>
<td>$\tau_1 &lt; t \leq \tau_2$</td>
<td>$b_{12}$</td>
<td>$b_{22}$</td>
<td>$\left(b_{32}^2 t\right)/\left(1+b_{32} t\right)$</td>
</tr>
<tr>
<td>$t &gt; \tau_2$</td>
<td>$b_{13}$</td>
<td>$b_{23}$</td>
<td>$b_{33}$</td>
</tr>
</tbody>
</table>

Increasing and/or decreasing trend in FRR can be depicted with the time dependent forms of $b_{0j}(t)$. From the literature study of various fault detection rates used in reliability growth modelling we summarize in the table 2.2 an interesting application and develop its model assuming three change points. This type of testing environment and testing strategy is usually seen while testing of safety critical systems.

**Table 2.1: Severity of faults with three Points**

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Simple</th>
<th>Hard</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Interval</td>
<td>Fault Detection Rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$0 \leq t \leq \tau_1$</td>
<td>$b_{11}$</td>
<td>$b_{21}$</td>
<td>$b_{31}$</td>
</tr>
<tr>
<td>$\tau_1 &lt; t \leq \tau_2$</td>
<td>$b_{12}$</td>
<td>$\left(b_{22}^2 t\right)/\left(1+b_{22} t\right)$</td>
<td>$\left(b_{32}^3 t^2\right)/\left(1+b_{32} t+b_{32}^3 t^2/2\right)$</td>
</tr>
<tr>
<td>$\tau_2 &lt; t \leq \tau_3$</td>
<td>$b_{13}$</td>
<td>$b_{23}$</td>
<td>$\left(b_{33}^2 t\right)/\left(1+b_{33} t\right)$</td>
</tr>
<tr>
<td>$t &gt; \tau_3$</td>
<td>$b_{14}$</td>
<td>$b_{24}$</td>
<td>$b_{34}$</td>
</tr>
</tbody>
</table>
In the beginning of the testing phase the testing team start with some constant FRR for each type of fault with $b_{11} \geq b_{21} \geq b_{31}$ due to motivation of the testing and debugging personnel to remove the critical faults (from users point of view) from the system. After a certain period of time the fault detection rate may increase or decrease due to the various possible changes in the testing process. Here we consider the case when fault detection rate first decreases due to the reason such as addition of new testing personnel, modifications in testing strategy etc. to further improve the overall efficiency of the testing, then start increasing as the testing progresses and the learning occurs ultimately reaching a certain constant level towards the end of the testing phase. For simple faults it is reasonable to assume constant FRR in each change point interval since not much learning testing strategies are applied for their removals. Here it may be noted the above explanation reflects one particular situation, various other possibilities also exist depending upon the testing environment and testing strategies employed.

2.1.3 Framework for modeling proposed SRGMs

In this section we have shown the derivation of the closed form solution of the models discussed above in table 2.1 and table 2.2.

2.1.3.1 Proposed SRGM1

Here $\tau_1$ is the time point at 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. Further, $\tau_2$ is the time moment by which the testing efficiency gained by the testing team results in the removal of different type of faults with a constant FDR.

Case 1: Simple Faults

Case 1.1 $0 \leq t \leq \tau_1$

$$m_1 (t) = b_{11} \left[ \alpha_1 - m_1 (t) \right]$$

Solving the equation (2.4) under the boundary conditions at $t = 0$, $m_1 (0) = 0$ we get

$$m_1 (t) = \alpha_1 \left( 1 - e^{-b_{11} t} \right)$$

(2.5)
Case 1.2 \( \tau_1 < t \leq \tau_2 \)

\[
m_1'(t) = b_{12} \left[ a_1 - m_1(t) \right]
\]

(2.6)

Solving the equation (2.6) under the boundary conditions at \( t = \tau_1, \ m_1(t) = m_1(\tau_1) \) we get

\[
m_1(t) = a_1 \left( 1 - e^{-b_{11} \tau_1 - b_{12} (t - \tau_1)} \right)
\]

(2.7)

Case 1.3: \( t > \tau_2 \)

\[
m_1'(t) = b_{13} \left[ a_1 - m_1(t) \right]
\]

(2.8)

Solving the equation (2.8) under the boundary conditions at \( t = \tau_3, \ m_1(t) = m_1(\tau_2) \) we get

\[
m_1(t) = a_1 \left( 1 - e^{-b_{11} \tau_1 - b_{12} (\tau_2 - \tau_1) - b_{13} (t - \tau_2)} \right)
\]

(2.9)

Case 2: Hard Faults

Case 2.1 \( 0 \leq t \leq \tau_1 \)

\[
m_2'(t) = \left( \frac{b_{21}^2 t}{1 + b_{21} t} \right) \left[ a_2 - m_2(t) \right]
\]

(2.10)

Solving the equation (2.10) under the boundary conditions at \( t = 0, \ m_2(t) = 0 \) we get

\[
m_2(t) = a_2 \left( 1 - \left( 1 + b_{21} t \right) e^{-b_{21} t} \right)
\]

(2.11)

Case 2.2 \( \tau_1 < t \leq \tau_2 \)

\[
m_2'(t) = b_{22} \left[ a_2 - m_2(t) \right]
\]

(2.12)

Solving the equation (2.12) under the boundary conditions at \( t = \tau_1, \ m_2(t) = m_2(\tau_1) \) we get

\[
m_2(t) = a_2 \left[ 1 - (1 + b_{21} \tau_1) e^{-b_{21} \tau_1 - b_{22} (t - \tau_1)} \right]
\]

(2.13)

Case 2.3 \( t > \tau_2 \)

\[
m_2'(t) = b_{23} \left[ a_2 - m_2(t) \right]
\]

(2.14)

Solving the equation (2.14) under the boundary conditions at \( t = \tau_2, \ m_2(t) = m_2(\tau_2) \) we get

\[
m_2(t) = a_2 \left[ 1 - \left( 1 + b_{21} \tau_1 \right) e^{-b_{21} \tau_1 - b_{22} (\tau_2 - \tau_1) - b_{23} (t - \tau_2)} \right]
\]

(2.15)
Case 3: Complex Faults

Case 3.1 $0 \leq t \leq \tau_1$

$$m_3'(t) = \left[ \frac{b_3^2 t^2}{2} \left( 1 + b_3 t + \frac{b_3^2 t^2}{2} \right) \right] \left[ a_3 - m_3(t) \right]$$

(2.16)

Solving the equation (2.16) under the boundary conditions at $t = 0$, $m_3(t) = 0$ we get

$$m_3(t) = a_3 \left[ 1 - \left( 1 + b_3 t + \frac{b_3^2 t^2}{2} \right) e^{-b_3 t} \right]$$

(2.17)

Case 3.2 $\tau_1 < t \leq \tau_2$

$$m_3'(t) = \left( \frac{b^2_{32} t}{(1 + b_{32} t)} \right) \left[ a_3 - m_3(t) \right]$$

(2.18)

Solving the equation (2.18) under the boundary conditions at $t = \tau_1$, $m_3(t) = m_3(\tau_1)$ we get

$$m_3(t) = a_3 \left[ 1 - \left( \frac{1 + b_{32} t}{1 + b_{32} \tau_1} \right) \left( 1 + b_{31} \tau_1 + \frac{b_{31}^2 \tau_1^2}{2} \right) e^{-b_{31} \tau_1 - b_{32} (t - \tau_1)} \right]$$

(2.19)

Case 3.3 $t > \tau_2$

$$m_3'(t) = b_{33} \left[ a_3 - m_3(t) \right]$$

(2.20)

Solving the equation (2.20) under the boundary conditions at $t = \tau_2$, $m_3(t) = m_3(\tau_2)$ we get

$$m_3(t) = a_3 \left[ 1 - \left( \frac{1 + b_{32} \tau_2}{1 + b_{32} \tau_1} \right) \left( 1 + b_{31} \tau_2 + \frac{b_{31}^2 \tau_2^2}{2} \right) e^{-b_{31} \tau_1 - b_{32} (\tau_2 - \tau_1) - b_{33} (t - \tau_2)} \right]$$

(2.21)

2.1.3.2 Proposed SRGM2

The example suits well to the real time safety critical systems however the testing environment might not be exactly similar as discussed above but the model can be modified accordingly (see table 2.1). Here we observe at time $\tau_1$ fault detection rate shows a decrease due to addition of new testing personnel, modifications in testing strategy, fault density etc carried to further improve the efficiency of the testing team. Then at time moment $\tau_2$, fault detection rate starts increasing as the testing progresses due to the learning of the testing team and ultimately reaching a stable value after $\tau_3$. 
Case 4: Simple Faults

**Case 4.1** $0 \leq t \leq \tau_1$

$$m_1'(r) = b_{11} \left[ a_1 - m_1(t) \right]$$

(2.22)

Solving the equation (2.22) under the boundary conditions at $t = 0, m_1(t) = 0$ we get

$$m_1(t) = a_1 \left( 1 - e^{-b_{11}t} \right)$$

(2.23)

**Case 4.2** $\tau_1 < t \leq \tau_2$

$$m_1'(r) = b_{12} \left[ a_1 - m_1(t) \right]$$

(2.24)

Solving the equation (2.24) under the boundary conditions at $t = \tau_1, m_1(t) = m_1(\tau_1)$ we get

$$m_1(t) = a_1 \left( 1 - e^{-b_{11}\tau_1 - b_{12}(t-\tau_1)} \right)$$

(2.25)

**Case 4.3** $\tau_2 < t \leq \tau_3$

$$m_1'(r) = b_{13} \left[ a_1 - m_1(t) \right]$$

(2.26)

Solving the equation (2.26) under the boundary conditions at $t = \tau_2, m_1(t) = m_1(\tau_2)$ we get

$$m_1(t) = a_1 \left( 1 - e^{-b_{11}\tau_1 - b_{12}(\tau_2 - \tau_1) - b_{13}(t-\tau_2)} \right)$$

(2.27)

**Case 4.4:** $t > \tau_3$

$$m_1'(r) = b_{14} \left[ a_1 - m_1(t) \right]$$

(2.28)

Solving the equation (2.28) under the boundary conditions at $t = \tau_3, m_1(t) = m_1(\tau_3)$ we get

$$m_1(t) = a_1 \left( 1 - e^{-b_{11}\tau_1 - b_{12}(\tau_2 - \tau_1) - b_{13}(\tau_3 - \tau_2) - b_{14}(t-\tau_3)} \right)$$

(2.29)

**Case 5: Hard Faults**

**Case 5.1** $0 \leq t \leq \tau_1$

$$m_2'(t) = b_{21} \left[ a_2 - m_2(t) \right]$$

(2.30)

Solving the equation (2.30) under the boundary conditions at $t = 0, m_2(t) = 0$ we get

$$m_2(t) = a_2 \left( 1 - e^{-b_{21}t} \right)$$

(2.31)
Case 5.2 $t_1 < t \leq t_2$

\[ m_2'(t) = \left( \frac{b_{22}^2 t}{1 + b_{22} t} \right) \left[ a_2 - m_2(t) \right] \] 

Solving the equation (2.32) under the boundary conditions at $t = t_1$, $m_2(t) = m_2(t_1)$ we get

\[ m_2(t) = a_2 \left[ 1 - \frac{1}{1 + b_{22} t_1} \right] e^{-b_{22} t_1 - b_{22} (t - t_1)} \] 

Case 5.3 $t_2 < t \leq t_3$

\[ m_2'(t) = b_{23} \left[ a_2 - m_2(t) \right] \] 

Solving the equation (2.34) under the boundary conditions at $t = t_2$, $m_2(t) = m_2(t_2)$ we get

\[ m_2(t) = a_2 \left[ 1 - \frac{1}{1 + b_{22} t_2} \right] e^{-b_{22} t_2 - b_{22} (t - t_2)} \] 

Case 5.4: $t > t_3$

\[ m_2'(t) = b_{24} \left[ a_2 - m_2(t) \right] \] 

Solving the equation (2.36) under the boundary conditions at $t = t_3$, $m_2(t) = m_2(t_3)$ we get

\[ m_2(t) = a_2 \left[ 1 - \frac{1}{1 + b_{22} t_3} \right] e^{-b_{22} t_3 - b_{22} (t - t_3)} \] 

Case 6: Complex Faults

Case 6.1 $0 \leq t \leq t_1$

\[ m_3'(t) = b_{31} \left[ a_3 - m_3(t) \right] \] 

Solving the equation (2.38) under the boundary conditions at $t = 0$, $m_3(t) = 0$ we get

\[ m_3(t) = a_3 \left( 1 - e^{-b_{31} t} \right) \] 

Case 6.2 $t_1 < t \leq t_2$

\[ m_3'(t) = \left( \frac{b_{32}^2 t_2}{2} \right) \left[ 1 + b_{32} t + \frac{b_{32}^2 t^2}{2} \right] \left[ a_3 - m_3(t) \right] \] 

By Solving the equation (2.40) under the boundary conditions at $t = t_1$, $m_3(t) = m_3(t_1)$ we get
\[ m_3(t) = a_3 \left[ 1 - \left( \frac{1 + b_3 t + b_3^2 t^2}{2} \right) \left( \frac{1 + b_3^2 t^2}{2} \right) e^{-b_3 (t - \tau_1)} \right] \] \hspace{1cm} (2.41)

**Case 6.3** \( \tau_2 < t \leq \tau_3 \)

\[ m_3'(t) = \left( \frac{b_3^2 t}{1 + b_3^2 t^2} \right) \left[ a_3 - m_3(t) \right] \] \hspace{1cm} (2.42)

Solving the equation (2.42) under the boundary conditions at \( t = \tau_2, \ m_3(t) = m_3(\tau_2) \) we get

\[ m_3(t) = a_3 \left[ 1 - \left( \frac{1 + b_3^2 \tau_2}{1 + b_3^2 \tau_1^2} \right) \left( \frac{1 + b_3^2 t^2}{2} \right) e^{-b_3 \tau_1 - b_3^2 (\tau_2 - \tau_1) - b_3^2 (t - \tau_2)} \right] \] \hspace{1cm} (2.43)

**Case 6.4** \( t > \tau_3 \)

\[ m_2'(t) = b_{34} \left[ a_2 - m_2(t) \right] \] \hspace{1cm} (2.44)

Solving the equation (2.44) under the boundary conditions at \( t = \tau_3, \ m_3(t) = m_3(\tau_3) \) we get

\[ m_3(t) = a_3 \left[ 1 - \left( \frac{1 + b_3^2 \tau_2}{1 + b_3^2 \tau_1^2} \right) \left( \frac{1 + b_3^2 \tau_3^2}{2} \right) e^{-b_3 \tau_1 - b_3^2 (\tau_2 - \tau_1) - b_3^2 (\tau_3 - \tau_2) - b_3^2 (t - \tau_3)} \right] \] \hspace{1cm} (2.45)

### 2.1.4 Modeling Total Fault Removal Phenomenon

The total fault removal phenomenon of the proposed SRGMs 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

\[ m(t) = m_1(t) + m_2(t) + m_3(t) \] \hspace{1cm} (2.46)

*Which implies*
Chapter 2  Software Reliability Growth Modeling incorporating errors of different Severity and change point

\[ m(t) = a_1 \left( 1 - e^{-b_1_t^1 - b_2(t_2 - t_1)} - b_3(t_3 - t_2) \right) + a_2 \left[ 1 - \left( 1 + \frac{b_2 t_2}{b_1 t_1} \right) e^{-b_2(t_2 - t_1)} - b_3(t_3 - t_2) \right] + a_3 \left[ 1 - \left( 1 + \frac{b_2 t_2}{b_3 t_3} \right) \left( 1 + \frac{b_3 t_3}{b_1 t_1} \right) e^{-b_3(t_3 - t_1)} - b_4(t_4 - t_3) \right] \]

(2.47)

for the Proposed SRGM1

and

\[ m(t) = a_1 \left( 1 - e^{-b_1_t^1 - b_2(t_2 - t_1)} - b_3(t_3 - t_2) \right) + a_2 \left[ 1 - \left( 1 + \frac{b_2 t_2}{b_1 t_1} \right) e^{-b_2(t_2 - t_1)} - b_3(t_3 - t_2) \right] + a_3 \left[ 1 - \left( 1 + \frac{b_2 t_2}{b_3 t_3} \right) \left( 1 + \frac{b_3 t_3}{b_1 t_1} \right) e^{-b_3(t_3 - t_1)} - b_4(t_4 - t_3) \right] \]

(2.48)

for the Proposed SRGM2

where \( a_1 = a p_1, a_2 = a p_2, a_3 = a(1 - p_1 - p_2) \)

(2.49)

2.1.5 Model Validation and Data Description

To check the validity of the proposed SRGMs and to find out their software reliability growth, it has been tested on two data sets. The Proposed SRGMs have been compared with Yamada’s modified exponential model and Shyur’s model (2003). In the Yamada’s two types of faults are assumed in the system, therefore for comparison purpose we have recomputed the mean value function of the removal phenomenon of the model assuming three types of faults in the system. The recomputed mean value function for the model is given by

\[ m(t) = \sum_{i=1}^{3} m_i(t) = \sum_{i=1}^{3} a p_i (1 - \exp^{-b_i}) \]

(2.50)
Where \( b_i; \ i = 1, 2, 3 \) are the removal rates per remaining fault for the simple, hard and complex faults. The Shyur’s model (2003) is based on Yamada’s model with one change point. Here \( b_i; \ i = 1, 3, 5 \) are the FRR before the change point and \( b_i; \ i = 2, 4, 6 \) after the change point. Since only a few data points are available and the number of unknown parameters is sixteen therefore to yield better estimates we assume \( b_1 = b_2 = b_3 = b_4 = b_5 \) (say), \( b_{21} = b_{22} = b_{23} = b_{24} = b_2 \) (say), \( b_{31} = b_{32} = b_{33} = b_{34} = b_3 \) (say) for the proposed models. The fault removal rates for each type of faults in each time interval are now as given in table 2.3-2.4.

### Table 2.3: Severity of faults with two Points

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Simple</th>
<th>Hard</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Interval</td>
<td>Fault Detection Rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 0 \leq t \leq \tau_1 )</td>
<td>( b_1 )</td>
<td>( \left( \frac{b_2^2 t}{1+b_2 t} \right) )</td>
<td>( \left( \frac{b_3^2 t^2/2}{1+b_3 t+\left(b_3^2 t^2/2\right)} \right) )</td>
</tr>
<tr>
<td>( \tau_1 &lt; t \leq \tau_2 )</td>
<td>( b_1 )</td>
<td>( b_2 )</td>
<td>( \left( \frac{b_3^2 t}{1+b_3 t} \right) )</td>
</tr>
<tr>
<td>( t &gt; \tau_2 )</td>
<td>( b_1 )</td>
<td>( b_2 )</td>
<td>( b_3 )</td>
</tr>
</tbody>
</table>

### Table 2.4: Severity of faults with three Points

<table>
<thead>
<tr>
<th>Type of Fault</th>
<th>Simple</th>
<th>Hard</th>
<th>Complex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Interval</td>
<td>Fault Detection Rates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( 0 \leq t \leq \tau_1 )</td>
<td>( b_1 )</td>
<td>( b_2 )</td>
<td>( b_3 )</td>
</tr>
<tr>
<td>( \tau_1 &lt; t \leq \tau_2 )</td>
<td>( b_1 )</td>
<td>( \left( \frac{b_2^2 t}{1+b_2 t} \right) )</td>
<td>( \left( \frac{b_3^2 t^2/2}{1+b_3 t+\left(b_3^2 t^2/2\right)} \right) )</td>
</tr>
<tr>
<td>( \tau_2 &lt; t \leq \tau_3 )</td>
<td>( b_1 )</td>
<td>( b_2 )</td>
<td>( \left( \frac{b_3^2 t}{1+b_3 t} \right) )</td>
</tr>
<tr>
<td>( t &gt; \tau_3 )</td>
<td>( b_1 )</td>
<td>( b_2 )</td>
<td>( b_3 )</td>
</tr>
</tbody>
</table>
The assumptions implies that the fault removal rate of simple faults doesn’t change over time whereas it changes for the hard and complex faults. For simple faults we may observe that not much learning is required for their removals. However it is difficult to remove hard and complex faults and require more resources for their removal therefore the developer puts more efforts to increase the learning of the testing and debugging team to remove these faults. In Shyur’s model (2003), the fault removal rates before and after the change point are all taken to be constant for each type of faults whereas in the proposed SRGMs the fault removal rate may change with respect to time according to the dynamic testing environment due to the time dependent forms of FRR for each type of fault in each change point interval. The locations of change points are identified by plotting the cumulative number of faults versus time. The Proposed SRGMs 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. The real time data sets used for estimation are

DS-1
This data is cited from (Mishra 1983). The software was tested for 38 weeks during which 2456.4 computer hours were used and 231 faults were removed. The Parameter Estimation result and the goodness of fit results for the proposed SRGMs are given in Table 2.5 and 2.6 respectively. The goodness of fit curves for Proposed SRGMs are given in Figure 2.1 and Figure 2.2. In this dataset we have taken $r_1 = 10$ and $r_2 = 22$ for the Proposed SRGM1 and $r_1 = 13, r_2 = 21, \text{ and } r_3 = 29$ for the Proposed SRGM2. 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.

DS-2
This data is cited from (Brooks and Motley 1980). The software was tested for 12 months during which 2657 faults were removed. The Parameter Estimation result and the goodness of fit results for the proposed SRGMs are given in Table 2.7 and 2.8 respectively. The goodness of fit curves for the Proposed SRGMs are given in Figure
2.3 and Figure 2.4. In this dataset we have taken \( \tau_1 = 4 \) and \( \tau_2 = 10 \) for the Proposed SRGM1 and \( \tau_1 = 3, \tau_2 = 5, \) and \( \tau_3 = 11 \) for the Proposed SRGM2.

### Table 2.5: Parameter Estimates for DS-1

<table>
<thead>
<tr>
<th>Models</th>
<th>Parameter Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( a )</td>
</tr>
<tr>
<td>Yamada SRGM (3.1)</td>
<td>263</td>
</tr>
<tr>
<td>Shyur SRGM [21]</td>
<td>264</td>
</tr>
<tr>
<td>Proposed SRGM1</td>
<td>293</td>
</tr>
<tr>
<td>Proposed SRGM2</td>
<td>260</td>
</tr>
</tbody>
</table>
Table 2.6: Goodness of Fit Metrics

<table>
<thead>
<tr>
<th>Models</th>
<th>R²</th>
<th>MSE</th>
<th>AIC</th>
<th>Bias</th>
<th>Variation</th>
<th>RMSPE</th>
<th>K-S Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamada SRGM (3.1)</td>
<td>.96935</td>
<td>113.79</td>
<td>228.72</td>
<td>-2.41</td>
<td>10.53</td>
<td>10.80</td>
<td>.198</td>
</tr>
<tr>
<td>Shyur SRGM [21]</td>
<td>.96937</td>
<td>113.70</td>
<td>223.32</td>
<td>-2.80</td>
<td>10.62</td>
<td>10.80</td>
<td>.234</td>
</tr>
<tr>
<td>Proposed SRGM1</td>
<td>.97224</td>
<td>103.07</td>
<td>197.05</td>
<td>-1.86</td>
<td>10.11</td>
<td>10.28</td>
<td>.109</td>
</tr>
<tr>
<td>Proposed SRGM2</td>
<td>.98051</td>
<td>72.37</td>
<td>193.57</td>
<td>-1.61</td>
<td>8.46</td>
<td>8.61</td>
<td>.096</td>
</tr>
</tbody>
</table>

Table 2.7: Parameter Estimates for DS-2

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>b₁</th>
<th>b₂</th>
<th>b₃</th>
<th>b₄</th>
<th>b₅</th>
<th>b₆</th>
<th>p₁</th>
<th>p₂</th>
<th>p₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamada SRGM (3.1)</td>
<td>2899</td>
<td>.190</td>
<td>.185</td>
<td>.103</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.445</td>
<td>.364</td>
<td>.190</td>
</tr>
<tr>
<td>Shyur SRGM [21]</td>
<td>3386</td>
<td>.143</td>
<td>.127</td>
<td>.133</td>
<td>.122</td>
<td>.128</td>
<td>.125</td>
<td>.442</td>
<td>.433</td>
<td>.125</td>
</tr>
<tr>
<td>Proposed SRGM1</td>
<td>3227</td>
<td>.158</td>
<td>.097</td>
<td>.073</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.596</td>
<td>.260</td>
<td>.143</td>
</tr>
<tr>
<td>Proposed SRGM2</td>
<td>3289</td>
<td>.176</td>
<td>.089</td>
<td>.048</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>.779</td>
<td>.167</td>
<td>.058</td>
</tr>
</tbody>
</table>
### Table 2.8: Goodness of Fit Metrics

<table>
<thead>
<tr>
<th>Models</th>
<th>R²</th>
<th>MSE</th>
<th>AIC</th>
<th>Bias</th>
<th>Variation</th>
<th>RMSPE</th>
<th>K-S Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yamada SRGM (3.1)</td>
<td>.98987</td>
<td>4971.95</td>
<td>202.17</td>
<td>7.42</td>
<td>73.23</td>
<td>73.61</td>
<td>.510</td>
</tr>
<tr>
<td>Shyur SRGM [21]</td>
<td>.99097</td>
<td>4445.64</td>
<td>238.21</td>
<td>-6.92</td>
<td>55.84</td>
<td>68.59</td>
<td>.686</td>
</tr>
<tr>
<td>Proposed SRGM1</td>
<td>.99332</td>
<td>3278.43</td>
<td>195.08</td>
<td>-4.97</td>
<td>42.42</td>
<td>58.67</td>
<td>.504</td>
</tr>
<tr>
<td>Proposed SRGM2</td>
<td>.99525</td>
<td>2329.77</td>
<td>184.91</td>
<td>-3.96</td>
<td>43.76</td>
<td>49.89</td>
<td>.467</td>
</tr>
</tbody>
</table>

**GOODNESS OF FIT CURVES**

![Goodness of Fit Curve for Proposed SRGM1 for DS-1](image)

**Figure 2.1: Goodness of fit curve for DS-1 (SRGM 1)**
Figure 2.2: Goodness of fit curve for DS-1 (SRGM 2)

Figure 2.3: Goodness of fit curve for DS-2 (SRGM 1)
It is evidently seen from the tables that the proposed SRGMs fits better than both Yamada SRGM and Shyur’s Model (2003) in terms of MSE, AIC, Bias, Variation, RMSPE, $R^2$, K-S Test. Here it can also be seen that $b_1 > b_2 > b_3$ as the testing teams have to spend more time to analyze and remove the cause of failure of hard and complex faults and therefore require greater efforts to remove them as the faults in the components comprising a complete software can be of different severity. For the first data set the results of Yamada model and Shyur model are very close. For the proposed SRGMs the results are better for both the data sets. Proposed SRGM2 gives better results than Proposed SRGM1 because of the flexibility in curve in capturing the relevant actual data points. The values of initial fault contents $a_1, a_2, a_3$ can be calculated from the tables 2.5, 2.7 for both the datasets i.e DS-1 and DS-2 using $a_i = a_{p_i}; i = 1, 2, 3$. 

**Figure 2.4 : Goodness of fit curve for DS-2 (SRGM 2)**