Chapter-4

Software Reliability Growth Models with various modes

Various authors have tried to develop a unifying approach so as to capture different growth curves, thus easing the model selection process. The work in this area done so far relates the fault removal process to the testing / execution time and does not consider the consumption pattern of resources such as computer working time, manpower and number of executed test cases etc. More realistic modeling can result if the reliability growth process is studied with respect the amount of expended testing efforts. Due to the complexity of software system and incomplete understanding of software, the testing process may not be perfect or the fault detection /correction rate may change at any time. In this chapter, we propose a generalized framework for deriving several existing as well as new testing effort dependent software reliability growth models incorporating change point and the possibility of imperfect debugging. The proposed framework is based on standard probability distribution functions. The developed models have been validated and verified using real data sets. Estimated Parameters and comparison criteria results have also been presented.

This chapter is based on the following paper:

4.1 Unified Framework for developing Testing effort Dependent Software Reliability Growth Models with change point and imperfect debugging

In most of the models discussed in earlier chapters, it is assumed that whenever an attempt is made to remove a fault, it is removed with certainty i.e. a case of perfect debugging. But the debugging activity is not always perfect because of number of factors like tester’s skill/expertise, complexity of the software etc. The testing team may not be able to remove/correct fault perfectly on observation/detection of a failure and the original fault may remain leading to a phenomenon known as imperfect debugging, or replaced by another fault resulting in error generation. In case of imperfect debugging the fault content of the software is not changed, but because of incomplete understanding of the software, the original detected fault is not removed perfectly. However, in case of error generation, the total fault content increases as the testing progresses because new faults are introduced in the system while removing the old original faults (We have described about this concept in chapter 1 in detail).

In this chapter, we present a unified framework for software reliability growth modeling with respect to testing effort expenditure and incorporate the concept of change point with imperfect debugging and error generation. This unified scheme is based on Probability distribution functions. It is also shown that previously reported Non-Homogeneous Poisson Process (NHPP) based SRGMs with imperfect debugging and error generation are special cases of the proposed framework. From this approach, we can not only obtain existing models but also develop some new NHPP models. The existing and new models derived here have been validated and evaluated on two actual software failure data sets. Non-linear regression based on least square method has been used for Parameter estimation and MSE (Mean Squared Error) and $R^2$ has been used as the comparison criteria. The goodness of fit curves have been drawn to illustrate the fitting of the models to the data graphically.

4.1.1 Notations

$m(W_t)$: The mean value function or the expected number of faults detected or removed by time $t$. 
Chapter 4  
Software Reliability Growth Models with various modes

\(a(W_t)\) : Total fault content of software dependent on time.

\(p\) : The probability of fault removal on a failure (i.e., the probability of perfect debugging).

\(\alpha\) : The rate at which the faults/errors may be introduced during the debugging process.

\(b\) : Fault removal/correction rate.

\(\lambda(W_t)\) : Intensity function for NHPP models or fault detection rate per unit time.

\(F(W_t)\): Distribution functions for fault removal/correction times.

\(f(W_t)\): Density functions for fault removal/correction times.

\(s(W_t)\): Hazard rate function.

\(\beta\) : Learning parameter in logistic function.

\(\tau\) : Change Point.

### 4.1.2 Basic Assumptions

The proposed models are based upon the following basic assumptions:

1. Fault removal phenomenon is modeled by NHPP.
2. Software is subject to failures during execution caused by faults remaining in the software.
3. Failure rate is equally affected by all the faults remaining in the software.
4. When a software failure occurs, an instantaneous repair effort starts and the following may occur:
   a) Fault content is reduced by one with probability \(p\)
   b) Fault content remains unchanged with probability \(1-p\).
5. During the fault removal process, whether the fault is removed successfully or not, new faults are generated with a constant probability \(\alpha\).
6. Fault detection / removal rate may change at any time moment \( \tau \).

Assumption 4 and 5 captures the effect of imperfect debugging and error generation respectively.

### 4.1.3 Model Development

Let the counting processes \( \{X(W_i), t \geq 0\} \) and \( \{N(W_i), t \geq 0\} \) represent the cumulative number of failures observed and faults corrected up to time \( t \) respectively and let the test begun at time \( t=0 \). Then the distribution of \( N(W_i) \) is given by

\[
\Pr(N(W_i) = n) = \sum_{j=0}^{\infty} \Pr(N(W_i) = n \mid X(0) = j) \Pr(X(0) = j) \tag{4.1}
\]

Here it can be noted that the conditional probability \( \Pr(N(W_i) = n \mid X(0) = j) \) is zero for \( j < n \). For \( j \geq n \) it is given by

\[
\Pr(N(W_i) = n \mid X(0) = j) = \binom{j}{n} (F(W_i))^n (1 - F(W_i))^{j-n} \tag{4.2}
\]

Therefore, we have

\[
\Pr(N(W_i) = n) = \frac{[a F(W_i)]^n}{n!} \exp(-a) \sum_{j=0}^{\infty} \frac{[a(1 - F(W_i))]^{j-n}}{(j-n)!} 
\]

Or we can write

\[
\Pr(N(W_i) = n) = \frac{(a F(W_i))^n \exp(-a F(W_i))}{n!} \tag{4.3}
\]

Hence we can conclude that the fault correction process is poisson with mean value function (MVF) as given by:

\[
m(W_i) = E[N(W_i)] = a F(W_i) \tag{4.4}
\]
As specified before, here \( F(W_t) \) is the testing effort dependent probability distribution function for fault correction times. It can be noted that \( F(W_t) \) so defined satisfy all the properties of probability distribution functions.

1. At \( t=0, W_t=0 \) and \( F(W_t)=0 \). In this paper, we have used three types of testing effort function namely Exponential, Rayleigh and Weibull type. All these functions satisfy the property that at \( t=0, W_t=0 \). It can be verified from their expressions, discussed in detail in appendix at the end of the paper.

2. For \( t>0, W_t>0 \) and \( F(W_t)>0 \).

3. In this paper we have assumed \( F(W_t) \) to be either of Exponential, Erlang, Logistic type. As \( t \) increases, \( W_t \) also increases indicating monotonically increasing nature of \( F(W_t) \). Similarly the continuity of \( F(W_t) \) can also be explained.

4. As testing continues for an infinitely large time i.e. \( t \to \infty, W_t \to \bar{W} \), the corresponding value of distribution function \( F(W_t) \) is \( F(\bar{W}) \). Here \( \bar{W} \) is a very large positive number representing the upper bound on the availability of testing resources. Therefore, \( F(\bar{W}) \) can be assumed to be of order 1.

From Equation (4.4), the instantaneous failure intensity function \( \lambda(W_t) \) is given by:

\[ \lambda(W_t) = a F'(W_t) \]

Or we can write

\[ \lambda(W_t) = \frac{dm}{dW_t} \frac{dt}{dt} = [a - m(W_t)] \frac{F'(W_t)}{1 - F(W_t)} \quad (4.5) \]

Let us define \( \sigma(W_t) = \frac{F'(W_t)}{1 - F(W_t)} \)
Here $s(W_t)$ represents hazard rate function or fault detection/correction rate per remaining fault of the software, or the rate at which the individual faults manifest themselves as failures during testing.

Now, Equation (4.5) can be written as:

$$\frac{dm}{dt} = s(W_t) \left[ a - m(W_t) \right]$$

(4.6)

### 4.1.4 Model Development with Change Point

Incorporating change-point concept, $s(W_t)$ i.e. fault detection/ correction rate per remaining fault of the software can be written as

$$s(W_t) = \begin{cases} 
\frac{F'_1(W_t)}{1 - F_1(W_t)} & t \leq \tau \\
\frac{F'_2(W_t)}{1 - F_2(W_t)} & t > \tau 
\end{cases}$$

(4.7)

where $F'_1(W_t) = \frac{d}{dt} F_1(W_t)$ and $F'_2(W_t) = \frac{d}{dt} F_2(W_t)$

Further, incorporating the change-point concept in modeling, Equation (4.6) becomes:

For $0 \leq t \leq \tau$

$$\frac{dm}{dt} = \left[ a - m(W_t) \right] \frac{F'_1(W_t)}{1 - F_1(W_t)}$$

Solving the above equation with initial condition at $t = 0, W'_t = 0$ and $m(W_t) = 0$, we get

$m(W_t) = a \cdot F_1(W_t)$

For $t > \tau$
\[
\frac{dm}{dt} = \left[ a - m(W_t) \right] \frac{F_2'(W_t)}{1 - F_2(W_t)}
\]

Solving above equation with initial condition at \( t = \tau \), \( W_t = W_\tau \) and \( m(W_t) = m(W_\tau) \), we get

\[
m(W_t) = a \left[ 1 - \frac{(1 - F_1(W_t))(1 - F_2(W_t))}{(1 - F_2(W_\tau))} \right]
\]

(4.8)

i.e.

\[
m(t) = \begin{cases} 
  a \frac{F_1(W_t)}{} & \text{for } t \leq \tau \\
  a \left[ 1 - \frac{(1 - F_1(W_t))(1 - F_2(W_t))}{(1 - F_2(W_\tau))} \right] & \text{for } t > \tau 
\end{cases}
\]

(4.9)

### 4.1.5 Model Development with Change Point & Two Type of Imperfect Debugging

In this section, we formulate distribution based software reliability growth models incorporating change-point and two types of imperfect debugging. Here we consider the following differential equation.

\[
\frac{dm}{dt} = b(W_t) \left( a - m(W_t) \right)
\]

(4.10)

Where \( b(W_t) \) is a fault detection rate per remaining faults at testing time \( t \). Here we consider the fault detection rate as hazard rate \( s(W_t) \), initial fault is not the constant but the function of time and incorporating the imperfect debugging. So the above equation can be written as
A Generalized Modeling Framework in Software Reliability and Related Problems

\[ \frac{dm}{dt} = s(W_i) p(W_i) \left( a(W_i) - m(W_i) \right) \] \hspace{1cm} (4.11)

We assume that faults can be introduced during the debugging phase with a constant fault introduction rate \( \alpha \). Therefore, the fault content rate function, \( a(W_i) \), is a linear function of the expected number of faults detected by \( W_i \) and it is defined by:

\[
a(W_i) = \begin{cases} 
  a + \alpha_i m(W_i) & \text{for } t \leq \tau \\
  a + \alpha_i m(W_i) + \alpha_2 (m(W_i) - m(W_{\tau})) & \text{for } t > \tau 
\end{cases} \] \hspace{1cm} (4.12)

and probability of perfect debugging rate will be

\[
p(W_i) = \begin{cases} 
  p_1 & \text{for } t \leq \tau \\
  p_2 & \text{for } t > \tau 
\end{cases} \] \hspace{1cm} (4.13)

Now using equation (4.7), (4.12) and (4.13), equation (4.11) can be rewritten as,

\[
\frac{dm(W_{\tau})}{dt} = \begin{cases} 
  \frac{f_1(W_i)}{1-F_1(W_{\tau})} p_1 \left( a + \alpha_i m(W_i) - m(W_i) \right) & \text{for } t \leq \tau \\
  \frac{f_2(W_i)}{1-F_2(W_{\tau})} p_2 \left( a + \alpha_i m(W_i) + \alpha_2 (m(W_i) - m(W_{\tau})) - m(W_{\tau}) \right) & \text{for } t > \tau 
\end{cases}
\]

After solving the above equations, we get the following solutions

\[
m(W_i) = \begin{cases} 
  \frac{a}{1-\alpha_1} \left[ 1 - (1-F_1(W_i))^{p_1(1-\alpha_1)} \right] & \text{for } t \leq \tau \\
  \frac{a}{1-\alpha_2} \left[ 1 - (1-F_1(W_i))^{p_1(1-\alpha_1)} \left( 1-F_2(W_i) \right)^{p_2(1-\alpha_2)} \right] \\
  + \left( \frac{\alpha_1 - \alpha_2}{1-\alpha_2} \right) m(W_{\tau}) & \text{for } t > \tau 
\end{cases} \] \hspace{1cm} (4.14)
DERIVATION OF NEW AND EXISTING MODELS

SRGM-1

Let

\[ F_1(W_t) = 1 - e^{b_1 W_t} \quad \text{for} \quad t \leq \tau \]

and

\[ F_2(W_t) = 1 - e^{b_2 W_t} \quad \text{for} \quad t > \tau \]

Substituting \( F_1(W_t) \) and \( F_2(W_t) \) into Equation (4.14), we get:

\[ m(W_t) = \begin{cases} 
\frac{a}{1-\alpha_1} \left[ 1 - e^{b_1 p_1 (1-\alpha_1) W_t} \right] & \text{for} \quad t \leq \tau \\
\frac{a}{1-\alpha_2} \left[ 1 - e^{b_2 p_1 (1-\alpha_1) W_t - b_2 p_2 (1-\alpha_2) (W_t - W_\tau)} \right] + \frac{\alpha_1 - \alpha_2}{1-\alpha_2} m(W_t) & \text{for} \quad t > \tau
\end{cases} \]

(4.15)

The above model can be reduced to the model given by Shyur (2003) if we consider the perfect debugging and no fault generation.

SRGM-2

Let \( F(W_t) \) be a two-stage Erlangian distribution function i.e.,

\[ F_1(W_t) = 1 - (1 + b_1 W_t) e^{-b_1 W_t} \quad \text{for} \quad t \leq \tau \]

and

\[ F_2(W_t) = 1 - (1 + b_2 W_t) e^{-b_2 W_t} \quad \text{for} \quad t > \tau \]

Substituting \( F_1(W_t) \) and \( F_2(W_t) \) into Equation (4.14), we get:
A Generalized Modeling Framework in Software Reliability and Related Problems

Department of Operational Research, University of Delhi

\[ m(W_t) = \begin{cases} \frac{a}{(1 - \alpha_1)} \\ 1 - ((1 + b_1 W_t) \exp(-b_1 W_t))^{p_1(1-\alpha_1)} \end{cases} \quad \text{for } t \leq \tau \]

\[ \begin{cases} \frac{a}{(1 - \alpha_2)} \\ 1 - ((1 + b_1 W_t) \exp(-b_1 W_t))^{p_1(1-\alpha_1)} \left(\frac{1 + b_2 W_t}{1 + b_2 W_t}\right)^{p_2(1-\alpha_2)} \exp\left(-b_1 p_1 (1-\alpha_1) W_t - b_2 p_2 (1-\alpha_2)(W_t - W_\tau)\right) + \frac{(\alpha_1 - \alpha_2)}{(1 - \alpha_2)} m(W_\tau) \end{cases} \quad \text{for } t > \tau \]

(4.16)

SRGM-3

Let \( F(W_t) \) be a logistic distribution function

\[ F_1(W_t) = \frac{1 - \exp(-b_1 W_t)}{1 + \beta_1 \exp(-b_1 W_t)} \quad \text{for } t \leq \tau \]

And

\[ F_2(W_t) = \frac{1 - \exp(-b_2 W_t)}{1 + \beta_2 \exp(-b_2 W_t)} \quad \text{for } t > \tau \]

Then the corresponding mean value function is given by:

\[ m(W_t) = \begin{cases} \frac{a}{(1 - \alpha_1)} \\ 1 - \left(\frac{1 + \beta_1}{1 + \beta_1 \exp(-b_1 W_t)}\right)^{p_1(1-\alpha_1)} \exp(-b_1 p_1 (1-\alpha_1) W_t) \end{cases} \quad \text{for } t \leq \tau \]

\[ \begin{cases} \frac{a}{(1 - \alpha_2)} \\ 1 - \left(\frac{1 + \beta_1}{1 + \beta_1 \exp(-b_1 W_t)}\right)^{p_1(1-\alpha_1)} \left(\frac{1 + \beta_2 \exp(-b_2 W_t)}{1 + \beta_2 \exp(-b_2 W_t)}\right)^{p_2(1-\alpha_2)} \exp\left(-b_1 p_1 (1-\alpha_1) W_t - b_2 p_2 (1-\alpha_2)(W_t - W_\tau)\right) + \frac{(\alpha_1 - \alpha_2)}{(1 - \alpha_2)} m(W_\tau) \end{cases} \quad \text{for } t > \tau \]

(4.17)

For further simplifying the estimation procedure we may assume \( \alpha_1 = \alpha_2 = \alpha \) and \( p_1 = p_2 = p \).
4.1.6 Parameter Estimation

To illustrate the estimation procedure and application of the SRGM (existing as well as proposed) we have carried out the data analysis of real software data set. The parameters of the models have been estimated using statistical package SPSS and the change-point of the data sets have been judged by using change-point analyzer.

The SRGM with mean value function $m(W_t)$ are estimated for finding their unknown parameters. For testing effort estimation we have worked out results on all three effort functions namely Exponential, Rayleigh and Weibull (refer Table 4.1 and 4.2). But for model parameter estimation we have used Weibull function as it gives best results as compared to other two effort functions.

Data set 1(DS-1)

The first data set (DS-1) had been collected during 35 months of testing a radar system of size 124 KLOC and 1301 faults were detected during testing. This data is cited from Brooks and Motley (1980). The change-point for this data set is 17th month.

The parameter estimation and comparison criteria results for DS-1 of all the models under consideration can be viewed through Table 4.3 and 4.4. It is clear from the table that the value of $R^2$ for SRGM-1 is higher and value of MSE is lower in comparison with other models and provides better goodness of fit for DS-1.

Data set 2(DS-2)

The second data set (DS-2) had been collected during 19 weeks of testing a real time command and control system and 328 faults were detected during testing. This data is cited from Ohba (1984). The change-point for this data set is 6th week.

The parameter estimation and comparison criteria results for DS-2 of all the models under consideration can be viewed through Table 4.5 and 4.6. It is clear from the table that the value of $R^2$ for SRGM-1 is higher and value of MSE is lower in comparison with other models and provides better goodness of fit for DS-2.
Table 4.1: Estimation of testing Effort Function Parameters for DS-1

<table>
<thead>
<tr>
<th>Testing Function</th>
<th>Effort Function Parameter Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$W$</td>
</tr>
<tr>
<td>Exponential</td>
<td>1030421</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>2873</td>
</tr>
<tr>
<td>Weibull</td>
<td>2669</td>
</tr>
</tbody>
</table>

Table 4.2: Estimation of testing Effort Function Parameters for DS-2

<table>
<thead>
<tr>
<th>Testing Function</th>
<th>Effort Function Parameter Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$W$</td>
</tr>
<tr>
<td>Exponential</td>
<td>8544</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>49</td>
</tr>
<tr>
<td>Weibull</td>
<td>799</td>
</tr>
</tbody>
</table>
## Table 4.3: Model Parameter Estimation Results (DS-1)

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>$b_1$</th>
<th>$b_2$</th>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>p</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM-1</td>
<td>1328</td>
<td>.076</td>
<td>.146</td>
<td>-</td>
<td>-</td>
<td>.016</td>
<td>.0003</td>
</tr>
<tr>
<td>SRGM-2</td>
<td>1308</td>
<td>.079</td>
<td>.127</td>
<td>-</td>
<td>-</td>
<td>.018</td>
<td>.0229</td>
</tr>
<tr>
<td>SRGM-3</td>
<td>1400</td>
<td>.328</td>
<td>.328</td>
<td>.335</td>
<td>.999</td>
<td>.004</td>
<td>.0950</td>
</tr>
</tbody>
</table>

## Table 4.4: Model Comparison Results (DS-1)

<table>
<thead>
<tr>
<th>Models</th>
<th>$R^2$</th>
<th>MSE</th>
<th>BIAS</th>
<th>VARAITION</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM-1</td>
<td>.99826</td>
<td>371.06</td>
<td>-2.886</td>
<td>19.32</td>
<td>19.54</td>
</tr>
<tr>
<td>SRGM-2</td>
<td>.99712</td>
<td>613.77</td>
<td>-7.442</td>
<td>23.98</td>
<td>25.10</td>
</tr>
<tr>
<td>SRGM-3</td>
<td>.99471</td>
<td>1127.1</td>
<td>-0.0003</td>
<td>34.06</td>
<td>34.06</td>
</tr>
</tbody>
</table>
Table 4.5: Model Parameter Estimation Results (DS-2)

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>b₁</th>
<th>b₂</th>
<th>β₁</th>
<th>β₂</th>
<th>p</th>
<th>α</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM-1</td>
<td>351</td>
<td>.127</td>
<td>.193</td>
<td>-</td>
<td>-</td>
<td>.236</td>
<td>.156</td>
</tr>
<tr>
<td>SRGM-2</td>
<td>328</td>
<td>.139</td>
<td>.122</td>
<td>-</td>
<td>-</td>
<td>.553</td>
<td>.179</td>
</tr>
<tr>
<td>SRGM-3</td>
<td>375</td>
<td>.153</td>
<td>.147</td>
<td>1.16</td>
<td>1.32</td>
<td>.278</td>
<td>.137</td>
</tr>
</tbody>
</table>

Table 4.6: Model Comparison Results (DS-2)

<table>
<thead>
<tr>
<th>Models</th>
<th>R²</th>
<th>MSE</th>
<th>BIAS</th>
<th>VARIATION</th>
<th>RMSPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM-1</td>
<td>.99247</td>
<td>77.72</td>
<td>-0.42</td>
<td>9.047</td>
<td>9.057</td>
</tr>
<tr>
<td>SRGM-2</td>
<td>.99029</td>
<td>141.7</td>
<td>-2.014</td>
<td>12.05</td>
<td>12.22</td>
</tr>
<tr>
<td>SRGM-3</td>
<td>.99081</td>
<td>94.79</td>
<td>-0.003</td>
<td>10.00</td>
<td>10.00</td>
</tr>
</tbody>
</table>
Chapter 4  
Software Reliability Growth Models with various modes

Figure 4.1: Goodness of fit curve for DS-1

Figure 4.2: Goodness of fit curve for DS-1