Considering Residual Faults of Burr Type XII Software Reliability Growth Model

B. Rama Devi¹, Dr. R. Satya Prasad² and Dr. G. Sridevi³
¹ Research Scholar, Department of CSE, Acharya Nagarjuna University, Guntur, India.
² Associate Professor, Department of CSE, Acharya Nagarjuna University, Guntur, India.
³ Professor, Dept. of CSE, KL University, Vaddeswaram, Guntur, India.

Abstract - Software Reliability Growth model (SRGM) is a mathematical model of how the software reliability improves as faults are detected and repaired. A large number of software reliability growth models have been proposed to analyze the reliability of software application during the testing phase, with the increasing demand to deliver high-quality software, more accurate software reliability models are required to estimate the optimal software release time and the cost of testing efforts. This paper proposes Burr type XII based Software Reliability growth model with Interval domain data. The unknown parameters of the model are estimated using the maximum likelihood (ML) estimation method. Reliability of a software system using Burr type XII distribution, which is based on Non-Homogeneous Poisson process (NHPP), is presented through estimation procedures. The performance of the SRGM is judged by its ability to fit the software failure data. How good does a mathematical model fit to the data is also being calculated. To access the performance of the considered SRGM, we have carried out the parameter estimation on the real software failure datasets.

Keywords – Software Reliability, Burr type XII distribution, NHPP, ML Estimation, Fault detection rate.

I. INTRODUCTION

Software reliability is defined as the probability of failure free software operation for a specified period of time in a specified environment (Lyu, 1996) (Musa et al., 1987)[3][4]. SRGM is a mathematical model of how the software reliability improves as faults are detected and required (Quadri and Ahmad, 2010)[12]. Among all SRGMs developed so far a large family of stochastic reliability models based on a Non-Homogeneous Poisson Process known as NHPP reliability model has been widely used. Software reliability is the most dynamic quality characteristic which can measure and predict the operational quality of the software system during its intended life cycle. If the selected model does not fit the collected software testing data relatively well. We would expect a low prediction ability of this model and the decision makings based on the analysis of this model would be far from what is considered to be optimal decision (Xie et al., 2001)[13]. This paper presents a method for model validation.

II. RELATED RESEARCH

This section presents the theory that underlies the proposed distributions and maximum likelihood estimation for complete data. If ‘t’ is a continuous random variable with pdf: \( f(t; \theta_1, \theta_2, ..., \theta_k) \). Where \( \theta_1, \theta_2, ..., \theta_k \) are k unknown constant parameters which need to be estimated, and cdf: \( F(t) \). Where, the mathematical relationship between the pdf and cdf is given by: \( f(t) = \frac{d(F(t))}{dt} \). Let ‘a’ denote the expected number of faults that would be detected given infinite testing time in case of finite failure NHPP models. Then, the mean value function of the finite failure NHPP models can be written as: \( m(t) = aF(t) \). Where, \( F(t) \) is a cumulative distributive function. The failure intensity function \( \lambda(t) \) in case of the finite failure NHPP models is given by: \( \lambda(t) = aF'(t) \) [8].

ISSN 2277-1956/V4N2-123-129 ©IJECSE
Considering Residual Faults of Burr Type XII Software Reliability Growth Model

A. NHPP Model

There are numerous software reliability growth models available for use according to probabilistic assumptions. The Non Homogenous Poisson Process (NHPP) based software reliability growth models are proved to be quite successful in practical software reliability engineering [4]. Model parameters can be estimated by using maximum Likelihood Estimate (MLE). NHPP model formulation is described in the following lines.

A software system is subjected to failures at random times caused by errors present in the system. Let \( \{N(t), t \geq 0\} \) be a counting process representing the cumulative number of failures by time ‘\( t \)’, where \( t \) is the failure intensity function, which is proportional to the residual fault content.

Let \( m(t) \) represent the expected number of software failures by time ‘\( s \)’. The mean value function \( m(t) \) is finite valued, non-decreasing, non-negative and bounded with the boundary conditions.

\[
m(t) = \begin{cases} 
0, & t = 0 \\
\lambda, & t \to \infty
\end{cases}
\]

Where ‘\( \lambda \)’ is the expected number of software errors to be eventually detected.

Suppose \( N(t) \) is known to have a Poisson probability mass function with parameters \( m(t) \) i.e.,

\[
P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, n = 0, 1, 2, \ldots \infty
\]

Then \( N(t) \) is called an NHPP. Thus the stochastic behaviour of software failure phenomena can be described through the \( N(t) \) process. Various time domain models have appeared in the literature that describes the stochastic failure process by an NHPP which differ in the mean value function \( m(t) \).

B. Proposed Model Description –

In this paper, we propose to access the software reliability based on Burr Type XII distribution model. The Burr distribution has a flexible shape and controllable scale and location which makes it appealing to fit to data. It is frequently used to model insurance claim sizes [5]. The mean value function and intensity function of Burr Type XII NHPP model are as follows.

The Cumulative distributive function (CDF) is given by

\[
m(t) = \int_0^1 \lambda(t)dt = a\left[1 - \left(1 + t^c\right)^{-b}\right]
\]

\[
= a \cdot F(t)
\]

The Probability Density Function (PDF) of Burr XII distribution are given, respectively by

\[
\lambda(t) = a\left(\frac{ct^{c-1}}{\left(1 + t^c\right)^{b+1}}\right) = a \cdot f(t)
\]

Where \( t>0, a>0, b>0 \) and \( c>0 \) denote the expected number of faults that would be detached given infinite testing time in case of finite failure NHPP models. In order to have an assessment of the software reliability, \( a, b \) and \( c \) are unknown parameters and estimated by using Newton Raphson method. Expressions are now delivered for estimating ‘\( a \)’, ‘\( b \)’ and ‘\( c \)’ for the Burr type XII model.
This is also a Poisson model with mean \( \lambda \).

Let \( S_k \) be the time between \( k-1 \)th and \( k \)th failure of the software product. Let \( X_k \) be the time up to the \( k \)th failure. Let us find out the probability that time between \( k-1 \)th and \( k \)th failures, i.e., \( S_k \) exceeds a real number 's' given that the total time up to the \( k-1 \)th failure is equal to \( x \).

i.e., \( p \left[ S_k > \frac{s}{X_{k-1}} = x \right] \)

\( R S_k / X_{k-1}(s/x) = e^{-[m(x+s)-m(s)]} \)

This Expression is called Software Reliability.

III. ILLUSTRATING THE MLE

In this section we develop expressions to estimate the parameters of the Burr type XII model based on Interval domain data. Parameter estimation is of primary importance in software reliability prediction.

A set of failure data is usually collected in one of two common ways, Interval domain data and time domain data. In this paper parameters are estimated from the Interval domain data.

The mean value function of Burr type XII model is given by

\[
m(t) = a \left[ 1 - \left(1 + t^c\right)^{-b} \right], \quad t \geq 0
\]

\[
\log L = \sum_{i=1}^{k} (n_i - n_{i-1}) \log \left[ m(t_i) - m(t_{i-1}) \right] - m(t_k)
\]

\[
\log L = \sum_{i=1}^{k} (n_i - n_{i-1}) \left\{ \log a + \log \left[ 1 - \left(1 + t_i^c\right)^{-b} \right] - \log \left[ 1 - \left(1 + t_i^c\right)^{-b} \right] \right\} - a + a \left(1 + t_i^c\right)^{-b}
\]

Taking the Partial derivative with respect to 'a' and equating to '0'.

\[
\left\{ \text{i.e.,} \quad \frac{\partial \log L}{\partial a} = 0 \right\}\]

\[
\therefore a = \sum_{i=1}^{k} \frac{(n_i - n_{i-1})}{(1 + t_i^c)^b - 1}
\]
Considering Residual Faults of Burr Type XII Software Reliability Growth Model

The parameter ‘b’ is estimated by iterative Newton Raphson Method using

\[ b_{n+1} = b_n - \frac{g(b)}{g'(b)} \]

Where \( g(b) \) and \( g'(b) \) are expressed as follows.

\[ g(b) = \frac{\partial \log L}{\partial b} = 0 \]

\[ \frac{\partial \log L}{\partial b} = g(b) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[ -\log(t_{i-1} + 1) - \log(t_i + 1) + \frac{(t_i + 1)^b \log(t_i + 1) - (t_{i-1} + 1)^b \log(t_{i-1} + 1)}{(t_i + 1)^b - (t_{i-1} + 1)^b} \right] \]

\[ + \sum_{i=1}^{k} \left( \frac{1}{(t_k + 1)^b - 1} \log \left( \frac{1}{1+t_k} \right) \right) \]

(5)

Again partial differentiating with respect to ‘b’ and equate to 0, we get

\[ g'(b) = \frac{\partial^2 \log L}{\partial b^2} = 0 \]

\[ \frac{\partial^2 \log L}{\partial b^2} = g'(b) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[ \frac{2(t_{i-1} + 1)^b (t_i + 1)^b \log(t_i + 1) \log \left( \frac{t_{i-1} + 1}{t_i + 1} \right)}{\left( (t_i + 1)^b - (t_{i-1} + 1)^b \right)^2} \right] \]

\[ + \sum_{i=1}^{k} (n_i - n_{i-1}) \log(1+t_k) \left( \frac{t_k + 1)^b \log(t_k + 1)}{\left( (t_k + 1)^b - 1 \right)^2} \right) \]

(6)

The parameter ‘c’ is estimated by iterative Newton Raphson Method using

\[ c_{n+1} = c_n - \frac{g(c_n)}{g'(c_n)} \]

Where \( g(c) \) and \( g'(c) \) are expressed as follows.

\[ g(c) = \frac{\partial \log L}{\partial c} = 0 \]

\[ \frac{\partial \log L}{\partial c} = g(c) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[ -\log(t_{i-1}) + \frac{t_{i-1}^c}{(1+t_{i-1})} - \log t_i^c + \frac{t_i^c \log t_i - t_{i-1}^c \log t_{i-1}}{(t_i^c - t_{i-1}^c)} \right] - \sum_{i=1}^{k} (n_i - n_{i-1}) \log \frac{t_k}{(1+t_k)} \]

(7)
\[ g'(c) = \frac{\partial^2 \log L}{\partial c^2} = 0 \]

\[
\frac{\partial^2 \log L}{\partial c^2} = g'(c) = \sum_{i=1}^{k} (n_i - n_{i-1}) \left[ \left\{ \log\left( \frac{t_{i-1}}{t_i} \right) \left( \frac{t_{i-1}^{c} t_i^{c}}{(t_i^{c}-t_{i-1}^{c})^2} \right) \{ \log t_i - \log t_{i-1} \} \right\} + \sum_{i=1}^{k} (n_i - n_{i-1}) \left( \frac{t_i^{c}}{(1+t_i^{c})^2} \right) \right]
\]

\[= \sum_{i=1}^{k} (n_i - n_{i-1}) \left( \frac{t_i^{c}}{(1+t_i^{c})^2} \right) \]

\[= \sum_{i=1}^{k} (n_i - n_{i-1}) \left( \frac{t_i^{c}}{(1+t_i^{c})^2} \right) \]

(8)

IV. DATA ANALYSIS

A set of failure data Phase 1 and Phase 2 taken from Pham (2005) and Release #1, #2, #3 and #4 datasets taken from Wood (1996) consists of the observation time(week), CPU Hours and the number of failures detected per week defects found [7][11].

Solving equations in Section III by Newton Raphson Method (N-R) method for all the data sets, the iterative solutions for MLEs of a, b, c of given software failure datasets are shown in Table-1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of samples</th>
<th>Estimated Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>a</td>
</tr>
<tr>
<td>Phase 1</td>
<td>21</td>
<td>25.994042</td>
</tr>
<tr>
<td>Phase 2</td>
<td>21</td>
<td>41.590454</td>
</tr>
<tr>
<td>Release #1</td>
<td>20</td>
<td>87.533224</td>
</tr>
<tr>
<td>Release #2</td>
<td>19</td>
<td>111.77847</td>
</tr>
<tr>
<td>Release #3</td>
<td>12</td>
<td>59.376054</td>
</tr>
<tr>
<td>Release #4</td>
<td>19</td>
<td>42.831021</td>
</tr>
</tbody>
</table>

V. METHOD OF PERFORMANCE ANALYSIS

The performance of SRGM is judged by its ability to fit the software failure data. The term goodness of fit denotes the question of “How good does a mathematical model fit to the data?”. In order to validate the model under study and to assess its performance, experiments on a set of actual software failure data have been performed. The performance evaluation of software reliability growth model is generally measured with sum of square errors (SSE) and correlation...
Considering Residual Faults of Burr Type XII Software Reliability Growth Model

index of regression curve equation (R-square). Among them, the model performance is better when SSE is smaller and R-square is close to 1.

SSE is used to describe the distance between actual and estimated number of faults detected totally, which is defined as

\[
SSE = \sum_{i=1}^{n} (y_i - m(t_i))^2
\]

Where \( n \) denotes the number of failure samples in failure data set, \( y_i \) denotes the number of faults observed to the moment \( t_i \), and \( m(t_i) \) denotes the estimated number of faults detected to the time \( t_i \) according to the proposed model. The model can provide a better goodness-of-fit when the value of SSE is smaller.

The equation of calculating the value R-square is written as:

\[
R - square = 1 - \frac{\sum_{i=1}^{n} (y_i - m(t_i))^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]

Where \( \bar{y} \) denotes the mean value of faults detected. The model can provide a better goodness-of-fit when the value of R-square is close to 1.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Reliability ((t_n+50))</th>
<th>SSE</th>
<th>R-Square</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 1</td>
<td>0.999985708</td>
<td>2603.5148</td>
<td>1.2859</td>
</tr>
<tr>
<td>Phase 2</td>
<td>0.999983394</td>
<td>10015.7216</td>
<td>1.3703</td>
</tr>
<tr>
<td>Release #1</td>
<td>0.999973211</td>
<td>20913.2148</td>
<td>0.3464</td>
</tr>
<tr>
<td>Release #2</td>
<td>0.999967139</td>
<td>42882.6562</td>
<td>0.9822</td>
</tr>
<tr>
<td>Release #3</td>
<td>0.999902327</td>
<td>8491.8095</td>
<td>0.8617</td>
</tr>
<tr>
<td>Release #4</td>
<td>0.999989656</td>
<td>5918.2451</td>
<td>1.1753</td>
</tr>
</tbody>
</table>

From the Table -2 it can be seen that the value of SSE is smaller and the value of R-square is more close to 1. The results indicate that our NHPP Burr type XII model based on fault detection rate fits the data in the given datasets, best and predicts the number of residual faults in software most accurately.

VI. CONCLUSION

Software reliability growth model can estimate the optimal software release time and the cost of testing efforts [14]. And SRGM can help project managers to determine the testing resources and manpower needed to achieve desired reliability requirements. So more accurate model is needed to decrease the testing cost and increase the profit of releasing software [15][16][17]. In this paper the fault detection rate is calculated with the number of faults remaining in the software. Considering the two factors jointly the fault detection rate is more realistic and accurate. Moreover, we have discussed the
performances of 6 datasets by using our new Burr type XII SRGM. The experiment result shows that the Phase 1 data set can provide a better goodness-of-fit compared with other datasets are given in Table 2. The reliability of the model over Release #4 data is high among the data sets which were considered.

REFERENCE