A Generalized Software Fault Classification Model

OMAR SHATNAWI
Prince Hussein bin Abdullah Information Technology College
Al al-Bayt University
P.O.BOX 130040, Mafraq 25113
JORDAN

Abstract: - Most non-homogenous Poisson process (NHPP) based software reliability growth models (SRGMs) presented in the literature assume that the faults in the software are of the same type. This assumption implies that the fault removal rate per remaining faults is independent of the testing time. However, this assumption is not truly representative of reality. It has been observed that the software contains different types of faults and each fault requires different strategies and different amount of testing-effort to remove it. This paper proposes a generalized model based on classification the faults in the software system according to their removal complexity. The removal complexity is proportional to the amount of testing-effort required to remove the fault. The testing-effort expenditures are represented by the number of stages required to remove the fault after the failure observation / fault isolation (with time delay between the stages). Therefore, it explicitly takes into account the faults of different severity and can capture variability in the growth curves depending on the environment it is being used and at the same time it has the capability to reduce either to exponential or S-shaped growth curves. Such modelling approach is very much suited for object-oriented programming and distributed development environments. Actual software reliability data have been used to demonstrate the proposed generalized model.

Key-Words: - software engineering, software testing, NHPP, SRGM, fault severity.

1 Introduction

Software reliability modelling is very important due to the fact that it is not possible to produce fault-free software. The faults in the software occur due to human imperfection. These faults manifest themselves in terms of failures when the software is run. Testing phase in the software development process aims at detecting and removing these faults and making the software more reliable. Thus it is very important to evaluate software reliability during testing phase, based on software fault data analysis. Models concerned with the relationship between time span of testing and the cumulative number of faults detected/removed through software testing are called software reliability growth models (SRGMs). Based on non-homogeneous Poisson process (NHPP), several SRGMs have been developed because the models can be easily applied in actual software development.

In general, among various SRGMs, two most important factors affect reliability: the number of initial faults and the fault removal rate [1]. The number of initial faults is the number of faults in the software at the beginning of the test. This number is usually a representative measure of software reliability. Knowing the number of residual faults can help to determine whether the software is suitable for customers to use or not, and how much more testing resources are required. It can provide an estimate of the number of failures that will eventually be encountered by the customers. The fault removal rate, on the other hand, is used to measure the effectiveness of fault removal by test techniques and test cases. In the literature [2, 3, 4, 5, 6, 7], most researchers assume a constant fault removal rate per fault in deriving their SRGM. That is, they assume that all faults have equal probability of being removed during the software testing process, and the rate remains constant. In reality, the fault removal rate strongly depends on the skill of test teams, program size and software testability.

The rest of this paper is organized as follows: Section 2 develops the proposed generalized model. Sections 3 and 4 give the method used for parameter estimation and the criteria used for validation and evaluation of the proposed approach respectively. The applications of the proposed model to actual software reliability data sets through data analysis and model comparisons are shown in Section 5. This paper concludes in Section 6.
2 Modelling In Software Reliability

2.1 Model Development
Most the SRGMs developed in the literature [2, 3, 4, 5, 6, 7] assume that the faults in the software are of the same type and a constant fault removal rate per fault in deriving their SRGM. This assumption is not truly representative of reality. It has been observed that the software contains different types of faults and each requires different strategies and different amount of testing-effort to remove it. If this assumption is not taken into account, the SRGM may give misleading results.

To address this problem several SRGMs have been developed in the literature [6, 8, 9, 10, 11]. The fault removal rate in all these models is assumed to be constant.

Through real data experiments and analyzes on several software development projects [6, 12, 13, 14, 15], it has been observed the fault removal rate has three possible trends as time progresses: increasing, decreasing or constants. It decreases when the software has been used and tested repeatedly, showing reliability growth. It can also increase if the testing techniques or requirements are changed, or new faults are introduced due to new software features or imperfect debugging. The learning-process of software developers has also been studied [8, 12, 13, 14]. The learning is closely related to the changes in the efficiency of testing during a testing phase. Learning usually manifests itself as a changing fault detection rate.

To capture the learning-process of the test-team, the researchers adopted a time-dependent fault removal rate. Therefore, the role of the learning process during the testing phase can be established. Kapur et al. proposed a model with three types of faults [16]. For each type, the fault removal rate is time-dependent. This paper extends the model to include n types of fault, explicitly take into account the faults of different severity, and can capture variability in the growth curves depending on the environment it is being used and at the same time it has the capability to reduce either to exponential or S-shaped growth curves.

2.2 Model Assumptions
1. Failure occurrence or fault removal phenomenon follows an NHPP.
2. The faults existing in the software are of finite types. They are distinguished by the amount of testing-effort needed to remove them. The testing-effort expenditures are represented by the number of stages required to remove the fault after the failure observation / fault isolation (with time delay or lag between the stages).
3. Each time a failure occurs, an immediate (delayed) effort takes place to decide the cause of the failure in order to remove it. The time delay between the failure observation and its subsequent fault removal is assumed to represent the severity of the faults. The more severe the fault, more the time delay.
4. The debugging process is prefect.
5. The fault removal rate for all fault types is a logistic function as it is expected the learning-process will grow with time.

2.3 Model Notations
\[ a \] Initial fault-content of the software.
\[ i \] Type of fault (\( i=1,2,\ldots,n \)).
\[ a_i \] Initial content of fault-type \( i \) \( (\sum_{i=1}^{n} a_i = a) \).
\[ b_i \] Proportionality constant represents failure rate / fault isolation(s) / fault removal rate per fault for fault-type \( i \).
\[ b_i(t) \] Logistic learning-process function, i.e., fault removal rate per fault for fault-type \( i \).
\[ m_i(t) \] Mean number of fault removed of type \( i \) in \( i \) number of processes (stages) by time \( t \).
\[ \beta \] Constant parameter in the logistic learning function.

2.4 Model Formulation
Assuming that the software consists of \( n \) different types of faults and on each type of fault a different strategy is required to remove the cause of failures due to that fault. Let for a type \( i (i=1,2,\ldots,n) \) fault, \( i \) different processes (stages) are required to detect/remove the cause of the failure.

The time-dependent fault removal rate per fault for \( i \) fault-type, which is a non-decreasing S-shape curve and capture the learning-process of the software testers given as

\[
b_i(t) = \frac{b_i}{1 + \beta \exp(-b_i t)} \quad (1)
\]

Assuming that the expected number of faults detected/removed in \( (t+\Delta t) \) is proportional to the number of faults remaining to be detected/removed, we may write the following differential equations:

For \( i=1 \)

\[
\frac{d}{dt} m_{i1}(t) = b_i(t)(a_i - m_{i1}(t)) \quad (2)
\]
For $i=2$
\[
\frac{d}{dt} m_{21}(t) = b_2(a_2 - m_{21}(t)) \\
\frac{d}{dt} m_{22}(t) = b_2(t)(m_{21}(t) - m_{22}(t))
\] (3)

For $i=3$
\[
\frac{d}{dt} m_{31}(t) = b_3(a_3 - m_{31}(t)) \\
\frac{d}{dt} m_{32}(t) = b_3(t)(m_{31}(t) - m_{32}(t)) \\
\frac{d}{dt} m_{33}(t) = b_3(t)(m_{32}(t) - m_{33}(t))
\] (4)

Similarly for $i=n$, we have
\[
\frac{d}{dt} m_{n1}(t) = b_n(a_n - m_{n1}(t)) \\
\frac{d}{dt} m_{n2}(t) = b_n(t)(m_{n1}(t) - m_{n2}(t)) \\
\frac{d}{dt} m_{n3}(t) = b_n(t)(m_{n2}(t) - m_{n3}(t)) \\
\hspace{2cm} \vdots \hspace{2cm} \vdots \hspace{2cm} \vdots \\
\frac{d}{dt} m_{nn}(t) = b_n(t)(m_{n-1}(t) - m_{nn}(t))
\]

Note that the first subscript stands for the type of fault and the second subscript stands for the number of processes (stages) required to remove the fault after its occurrence / fault detection and is dependent on the type of the fault, i.e., if the fault is of the type $k$, then it will be removed in $k$ stages because of its complexity.

Solving the differential equations (2), (3), (4), and (5) respectively we get
\[
m_i(t) = m_{i1}(t) = a_1 \frac{1 - \exp(-b_1 t)}{1 + \beta \exp(-b_1 t)}
\]
\[
m_{2}(t) = m_{22}(t) = a_2 \frac{1 - (1 + b_2 t)\exp(-b_2 t)}{1 + \beta \exp(-b_2 t)}
\]
\[
m_{3}(t) = m_{33}(t) = a_3 \frac{1 - (1 + b_3 t + \frac{b_3^2 t^2}{2})\exp(-b_3 t)}{1 + \beta \exp(-b_3 t)}
\]
\[
\hspace{2cm} \vdots \hspace{2cm} \vdots \hspace{2cm} \vdots \\
m_{n}(t) = m_{nn}(t) = a_n \frac{1 - \sum_{j=0}^{n-1} \exp(-b_j t)}{1 + \beta \exp(-b_j t)}
\]

The proposed generalized model is the superposition of all the NHPP with mean value functions given in equation (6). Thus, the mean value function of the superposed NHPP is
\[
m_{G\infty}(t) = \sum_{i=1}^{n} m_i(t) = \sum_{i=1}^{n} a_i \frac{1 - \sum_{j=0}^{i-1} \exp(-b_j t)}{1 + \beta \exp(-b_j t)}
\] (7)

The removal rate per faults for the first three types of faults is given as
\[
\hat{b}_i(t) = \frac{d}{dt} m_{i1}(t) = \frac{b_i}{1 + \beta \exp(-b_i t)}
\]
\[
\hat{b}_2(t) = -\frac{d}{dt} m_{21}(t) = \frac{b_2(1 + \beta + b_2 t) - b_2(1 + \beta \exp(-b_2 t))}{(1 + \beta \exp(-b_2 t))(1 + \beta + b_2 t)}
\]
\[
\hat{b}_3(t) = -\frac{d}{dt} m_{31}(t) = \frac{b_3(1 + \beta + b_3 t + \frac{b_3^2 t^2}{2}) - b_3(1 + \beta \exp(-b_3 t))}{(1 + \beta \exp(-b_3 t))(1 + \beta + b_3 t + \frac{b_3^2 t^2}{2})}
\] (8)

We observe that $\hat{b}_1(t)$, $\hat{b}_2(t)$, and $\hat{b}_3(t)$ increase monotonically with time $t$ and tend to the constants $b_1$, $b_2$ and $b_3$ respectively as $t \to \infty$. Thus, in the steady state, $m_{2}(t)$ and $m_{3}(t)$ fault growth curves behave similar to the $m_{1}(t)$ fault growth curve and hence there is no loss of generality in assuming the steady state rates $b_2$ and $b_3$ to be equal to $b_1$. After substituting $b_2 = b_3 = b_1$ in the right hand side of equation (8), one can see that $\hat{b}_2(t) > \hat{b}_3(t) > \hat{b}_1(t)$, which is in accordance with the severity of the faults [6, 16, 17]. Generalizing for arbitrary $n$, we can assume $b_1 = b_2 = b_3 = \ldots = b_n = b$ (say), then equation (7) can be written as
\[
m_{G\infty}(t) = \sum_{i=1}^{n} a_i \frac{1 - \sum_{j=0}^{n-1} \exp(-b_j t)}{1 + \beta \exp(-b_j t)}
\] (9)

The proposed generalized model given in equation (9) is very interesting from various points of view. Besides the above-mentioned interpretation as a general flexible S-shaped fault removal model, it can also reduce to the models [2, 3, 6, 11, 12, 13, 16]. Thus, it is able to model both cases of strictly decreasing failure intensity and the case of increasing-and-decreasing failure intensity.

3 Parameter Estimation Technique
The MLE method is used to estimate the parameters of the proposed model given in equation (9). Since all the data sets used are given
in the form of pairs \((t_i, x_i)(i=1,2,\ldots,k)\), where \(x_i\) is the cumulative number of faults detected by time \(t_i\) \((0 < t_1 < t_2 < \ldots < t_k)\) and \(t_i\) is the accumulated time spent to remove \(x_i\) faults. The Likelihood function \(L\) for the unknown parameters with the mean value function \(m(t)\) is given as

\[
L(\text{Parameter } | (t_i, x_i)) = \prod_{i=1}^{k} \left( \frac{[m(t_i) - m(t_{i-1})]^{-x_i}}{(x_i - x_{i-1})!} \right)^{x_i} \exp\left(-\frac{(m(t_i) - m(t_{i-1}))}{x_i}ight)
\]

(10)

The MLE of the SRGM parameters can be obtained by maximizing equation (10) with respect to the following constraints: \((a_i \geq 0, 0 < b < 1, \beta \geq 0)\).

4 Model Validation

To check the validity of the proposed model given in equation (9) to describe the software reliability growth, it has been tested on three actual software reliability data sets, DS-I, DS-II, and DS-III cited from a real software development projects [18], [8], and [4] respectively. DS-I were recorded from test of a Command and Control Software System during 15 weeks in which 1138 faults were detected. DS-II had been collected during 21 days of testing, 46 faults were detected during this period. DS-III had been collected during 21 weeks of testing System T1 (a Real-Time Command and Control Application), 136 faults were removed during the period.

The performance of an SRGM judged by its ability to fit the past software fault/failure data and to predict satisfactorily the future behavior from present and past data behavior [4, 6].

4.1 The Goodness of Fit Criteria

4.1.1 The Sum of Squared Error (SSE)

This metrics measures the distance of a model estimate value from the actual data, as follows

\[
SSE = \sum_{i=1}^{k} (\hat{m}_{GE-s}(t_i) - x_i)^2
\]

(11)

where \(k\) is the number of observations, \(\hat{m}_{GE-s}(t_i)\) is the estimated cumulative number of faults by time \(t_i\) obtained from the fitted mean value function of equation (9) and \(x_i\) is the total number of faults removed by time \(t_i\). Lower value of SSE indicates less fitting error, thus better goodness of fit.

4.1.2 The Akaike Information Criterion (AIC)

This criterion was first proposed as SRGM model selection tool by [19] and defined as

\[
AIC = -2 \times \log(\text{Max. of Likelihood function}) + 2 \times N
\]

where \(N\) is the number of the parameters used in the model. Lower value of AIC indicates more confidence in the model thus a better fit and predictive validity.

4.1.3 Coefficient of Multiple Determination (R^2)

This Goodness-of-fit measure can be used to investigate whether a significant trend exists in the observed failure intensity. We define this coefficient as the ratio of the Sum of Squares (SS) resulting from the trend model to that from a constant model subtracted from 1, that is

\[
R^2 = 1 - \frac{\text{residual SS}}{\text{corrected SS}}
\]

(13)

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

In other words, we evaluate the performance of the models using SSE, AIC and R^2 metrics. For SSE and AIC, the smaller the metric value the better the model fits relative to other models run on the same data set. For R^2, the larger the metric value the better the model fits relative to other models run on the same data set.

5 Data Analyses and Model Comparisons

The software reliability data and the comparison criteria in terms of goodness-of-fit defined in Section 4 are applied to check the performance of the proposed generalized model defined in equation (9) with ‘n’ types of faults \((n=2, 3, 4, 5)\). That is, the models under comparison are \(m_{GF:2}(t)\), \(m_{GF:3}(t)\), \(m_{GF:4}(t)\), and \(m_{GF:5}(t)\) that estimates the presence of two, three, four and five types of faults respectively. The model [8] that estimate the presence of just one type of fault i.e., \(m_{GF:1}(t)\), has been excluded.

DS-I

The results of the parameter estimation of the models under comparison are given in Table I for DS-I. On applying \(m_{GF:2}(t)\), the model reduces to the exponential model [2] and reveals the presence of only one type of faults. Accordingly, the fault removal phenomenon is described by just a fault type1. On applying \(m_{GF:3}(t)\), the mode reveals the presence of only two types 1 and 3 of faults where the majority of faults are of type 1. Accordingly,
the fault removal phenomenon is described by a combination of faults types 1 and 3. On applying \( m_{GF,1}(t) \), the model reveals the presence of only two types 1 and 2 of faults where the majority of faults are of type 1. Accordingly, the fault removal phenomenon is described by a combination of faults types 1 and 4. On applying \( m_{GF,2}(t) \), the model reveals the presence of only two types 1 and 5 of faults where the majority of faults are of type 1. Accordingly, the fault removal phenomenon is described by a combination of faults types 1 and 3.

6 Conclusion

Based on Section 5, it is observed that the introduction of more fault types increases the flexibility of the model. To increase the flexibility of the model, the model should include all the fault types available in the software. As the number of fault types may tend to be large, modelling each fault type individually is not practically possible. Therefore, the model has been tested for only 5 types of faults. However, if the faults categorized into a smaller number of groups, where each group contains the faults of common characteristics, the number of faults types can be reduced to the level that it can be practically handled by the proposed generalized model. Finally, the proposed modelling approach provides a broad framework for developing NHPP type of continuous time SRGMS.

References:


Table 1: Parameter estimation for DS-I

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>a₁, a₂, a₃, a₄, a₅, b, β</td>
<td>.995</td>
<td>265.2</td>
<td>122.1</td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.996</td>
<td>22006.2</td>
<td>122.1</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.986</td>
<td>15318.4</td>
<td>264.7</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.994</td>
<td>6723.0</td>
<td>236.7</td>
<td></td>
</tr>
<tr>
<td>mGF,5(t)</td>
<td>.996</td>
<td>4524.45</td>
<td>229.9</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Goodness of fit for DS-I

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>.981</td>
<td>22005.9</td>
<td>286.1</td>
<td></td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.986</td>
<td>15318.4</td>
<td>264.7</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.994</td>
<td>6723.0</td>
<td>236.7</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.996</td>
<td>4524.45</td>
<td>229.9</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Parameter estimation for DS-II

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>a₁, a₂, a₃, a₄, a₅, b, β</td>
<td>.995</td>
<td>264.7</td>
<td>122.1</td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.996</td>
<td>22006.2</td>
<td>122.1</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.986</td>
<td>15318.4</td>
<td>264.7</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.994</td>
<td>6723.0</td>
<td>236.7</td>
<td></td>
</tr>
<tr>
<td>mGF,5(t)</td>
<td>.996</td>
<td>4524.45</td>
<td>229.9</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Goodness of fit for DS-II

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>.994</td>
<td>25.0</td>
<td>78.8</td>
<td></td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.995</td>
<td>24.5</td>
<td>76.5</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.995</td>
<td>25.0</td>
<td>76.5</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.994</td>
<td>25.1</td>
<td>80.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Parameter estimation for DS-III

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>a₁, a₂, a₃, a₄, a₅, b, β</td>
<td>.994</td>
<td>80.7</td>
<td></td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.995</td>
<td>25.0</td>
<td>78.8</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.995</td>
<td>24.5</td>
<td>76.5</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.994</td>
<td>25.0</td>
<td>76.5</td>
<td></td>
</tr>
<tr>
<td>mGF,5(t)</td>
<td>.995</td>
<td>25.1</td>
<td>80.7</td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Goodness of fit for DS-III

<table>
<thead>
<tr>
<th>Models under Comparison</th>
<th>Parameter Estimation</th>
<th>R²</th>
<th>SSE</th>
<th>AIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>mGF,1(t)</td>
<td>.996</td>
<td>208.6</td>
<td>110.1</td>
<td></td>
</tr>
<tr>
<td>mGF,2(t)</td>
<td>.997</td>
<td>151.5</td>
<td>113.1</td>
<td></td>
</tr>
<tr>
<td>mGF,3(t)</td>
<td>.995</td>
<td>241.4</td>
<td>121.3</td>
<td></td>
</tr>
<tr>
<td>mGF,4(t)</td>
<td>.996</td>
<td>225.0</td>
<td>119.2</td>
<td></td>
</tr>
<tr>
<td>mGF,5(t)</td>
<td>.995</td>
<td>265.2</td>
<td>122.1</td>
<td></td>
</tr>
</tbody>
</table>