SRGM Tool with Two Dimension Imperfect Debugging

B.Anni Princy¹,  
¹Research Scholar, Department of Computer Science & Engineering, Sathyabama University, Chennai. Email: shiyaprincy27@gmail.com

Dr. S. Sridhar²,  
²Dean-Cognitive & Central Computing Facility, RV College of Engineering, Bangalore-560059 Email: drsrsridhar@yahoo.com

Abstract—This paper describes the SRGM tool with two dimension imperfect debugging. Even though there are various testing effort functions found to improve the software reliability growth models, we are still more lacking in eliminating imperfect debugging. We will consider the two dimensional as testing time and testing coverage, where the time behavior of testing-effort are described by Gompertz and analyzed by optimal release time. To isolate the software error, the error detection is implemented by Yamada delayed S-Shaped model. The model parameters such as the number of persons involved, Number of test cases and time are estimated by AE and MSE and the software measures are tested through experimental data and with some real data from various projects. The performance results are analyzed and compared with other models to show the proposed testing-effort has better fault detection. The optimal release time on cost and reliability is also evaluated.

Keywords—Software Reliability Growth Model, Optimal Software release time, Testing Effort Function, Cost and Reliability.

I. INTRODUCTION

Now-a-days, all applications are machine dependent and it’s mostly wants to be more reliable for usage. SRGM is a mathematical model that shows the improvement of software reliability by reducing and detecting the failure and the error. Software Reliability is one of the most important functions which correctly predict the whole system[5][7]. Various SRGMs has been modeled during the past few years (MUSA et al 1987; LYU,1996; PHAM,2000). There are more SRGM available for evaluating some important parameters metrics like Time, Number of faults, MTBF, MTTF.(Yamada et al,1986,1987,1990,1991,1993,Kapur and Karg 1990,...).Most of the SRGMs are based on NHPP which models the distribution of testing-effort.

In most research models they are assuming the consumption rate as constant. The function which describes the effort over specific time and finding how effective are referred as testing effort-function(TEF)[6][12].Many found issue of the TEF(Musa,Yamada et al,Bokhari and Ahmed,Kapur et al and Huang et al) proposed SRGM to define various metrics like Testing time, the amount of Testing-effort and the number of Fault detected during testing.

This paper integrates Gompertz TEF and S-Shaped SRGM. Further we extended them using Yamada delayed S-Shaped SRGM with two types of Imperfect Debugging[3][4]. Real data from different applications are applied and analyzed. The paper is organized as follows,

II. SOFTWARE RELIABILITY GROWTH MODELING

SRGMs are grounded upon the theory of programs reliability and are a gathering of the faults found over it. Mostly statistical techniques are observed to identify failures during software testing and operation to forecast the software reliability[10][11]. The faults are identified and removed to make the system more reliable. More models are utilized to estimate the characteristics about the number of faults remaining in the system[1][2]. Unluckily, applying reliability growth models to the software faults is inhibited by a significant problem like the lack of quality data. The previous literature on reliability growth models generally assumes that they have been applied during pre-release testing. In order to be more effective, reliability growth models require the environment from which the data is obtained during the testing environment and employed after deployment. However it depends on an abnormal data, the data outside the bounds of a normal operational profile[8][9]. For a long period of time it is considered that the real world environments and the operational profile comprises all possible input. These models also required to reduce the time and to normalize the testing effort. If calendar time is used then it should be normalized for the number of testers.

\[ AE = \left| \frac{M_a - a}{M_a} \right| \]

III. PERFORMANCE ANALYSIS

To estimate the performance of the software reliability growth model, we move for the following.

1. The accuracy of estimate is defined as (AE)

Where \( M_a \) the actual cumulative number of is is detected errors after the test, and \( a \) is the estimated number of initial errors. For practical purposes, \( M_a \) is obtained from software error tracking after software testing.
2. The Mean of Squared Error (MSE)

\[ MSE = \frac{1}{k} \sum_{i=1}^{k} [m(t_i) - m_i]^2 \]

3. The Coefficient of multiple determination is defined as

\[ R^2 = \frac{S(\hat{\beta}, \theta) - S(\hat{\beta}, \hat{\theta})}{S(\hat{\beta}, \hat{\theta})} \]

Where \( \hat{\alpha} \) is the LSE of \( \alpha \) for the model with only a constant term, where \( \beta = 0 \) and \( \theta = 1 \). Therefore, \( R^2 \) measures the percentage of total variation about the model and as well as the data fitness. The best model is the one which provides the best data fitness with higher \( R^2 \).

IV RESULTS AND DISCUSSIONS

Numerical Examples:

Dataset 1: The actual data set consists of 1,57,1000 lines of code in the application. During the experiment we found 405 software error and it has been removed with estimated amount of time period. To estimate the parameters \( \alpha, \beta, \) and \( \theta \) of the Gompertz testing effort function, by applying the function the estimated parameters are obtained as

\[ \hat{\alpha} = 1168.2432, \hat{\beta} = 0.004743, \hat{\theta} = 0.1257241 \]

Using \( \alpha, \beta \) and \( \theta \) parameters the value of \( a \) and \( r \) is obtained as

\[ \hat{a} = 628.877, \hat{r} = 0.082432 \]

Table I shows the estimated value with the comparative results of different SRGM with Dataset 1.

<table>
<thead>
<tr>
<th>Model</th>
<th>A</th>
<th>r</th>
<th>AE (%)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed Model</td>
<td>628.877</td>
<td>0.082432</td>
<td>69.97772</td>
<td>83.2873</td>
</tr>
<tr>
<td>Yamada Rayleigh Model</td>
<td>613.732</td>
<td>0.018553</td>
<td>63.9122</td>
<td>134.0952</td>
</tr>
<tr>
<td>Huang Logistic Model</td>
<td>474.083</td>
<td>0.03276</td>
<td>11.0612</td>
<td>128.5832</td>
</tr>
</tbody>
</table>

Table I: Comparative results of different SRGM with DS-1

From the above tables and graphs, it is shown that AE is very high and MSE is low than previous methods. Thus the proposed method is very effective.

V CONCLUSION

Software Reliability system is one of the essential aspects of the software testing phase of the software development life cycle. In this paper, we have concluded that the proposed SRGM has better ability as compared to the other SRGMs and produces a reasonable predictable capability for the actual software failure data.
REFERENCES


