SRGM Tool with Two Dimension Imperfect Debugging

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

Dr. S. Sridhar[2]
[2]Dean-Cognitive & Central Computing Facility, RV College of Engineering, Bangalore-560059
Email: drssridhar@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, Section II describes about traditional overviews of TEF. Section III describes about the Gompertz TEF and the integration of Yamada Delayed S-Shaped with GTEF. Cost and Reliability are estimated with Release time in Section IV and about two types of imperfect debugging on the proposed model in Section V

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 resting 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.

International Journal of Engineering Research & Technology (IJERT)
NCITSF`14 Conference Proceedings
ISSN: 2278-0181

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

$$R^2 = \frac{S(\hat{\hat{\alpha}},0,1) - S(\hat{\alpha},\hat{\beta},\hat{\theta})}{S(\hat{\hat{\alpha}},0,1)}$$

Where ά is the LSE of α for the model with only a constant term, where β =0 and θ =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 α, β, and θ of the Gompertz testing effort function, by applying the function the estimated parameters are obtained as

$$\hat{\alpha} = 1168.24323, \hat{\beta} = 0.004743, \hat{\theta} = 0.1257241$$

Using α, β and θ 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.

Model	A	r	AE (%)	MSE
Proposed Model	628.877	0.082432	69.97772	83.2873
Yamada Rayleigh Model	613.732	0.018553	63.9122	134.0952
Huang Logistic Model	474.083	0.03276	11.0612	128.5832

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

Table II shows the estimated value with the comparative results of different SRGM with Dataset 2.

Model	A	R	AE (%)	MSE
Proposed Model	652.07	0.019631	58.04	111.56
Yamada Rayleigh Model	565.35	0.019659	57.91	122.09
Huang Logistic Model	394.08	0.04272	10.06	118.59

Table II: Comparative results of different SRGM with DS-2

The graphical representation of AE and MSE for the proposed method, Yamada Rayleigh model and Huang Logistic model are shown in the following graphs.

Fig1 Comparison of AE with proposed,Yamada and Huang models for DS-1

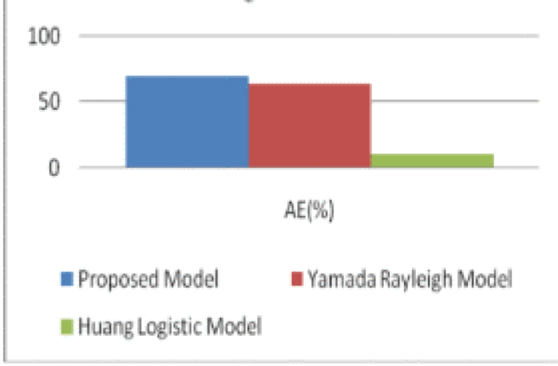
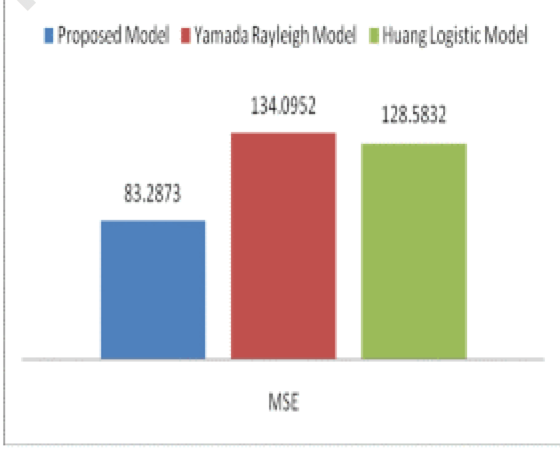


Fig2: Comparison of MSE with Proposed,Yamada & Huang Models for DS-2



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 Reliablity 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

[1] M.R.Lyu," Applying Reliability Model More effectively", University of Lowa, Jet Prolusion Laboratory", Caltech,July,2004,pp.43-52.

[2] W.K.Ehrlich-Thomas J.Emerson,"Modeling Software Failures and Reliablity growth during System testing" AT & T Bell Laboratories,8 Corporate Place,ACM,2005,pp.72-87

[3] J.Lian,"software Reliabilty Presentation",ECE-355 Tutorial ,2004,pp.1-24

[4] S,Yamada,M.Ohba,and S.Osaki,"S-Shaped Reliabity Growth Modeling for Error Detection," IEEE Transaction Reliabilty,Dec 2005,pp.475-478

[5] Carina Andersson, "A replicated empirical study of a selection method for software reliability growth models," Journal of Empirical Software Engineering, Vol. 12, No. 2, pp. 161–182, Apr 2007.

[6] Chin-Yu Huang, Sy-Yen Kuo and Michael R. Lyu, "An Assessment of Testing- Effort Dependent Software Reliability Growth Models,"IEEE Transactions on Reliability, Vol. 56, No. 2, pp. 198-211, Jun 2007.

[7] V. B. Singh1; Kalpana Yadav, Reecha Kapur and V. S. S. Yadavalli, "Considering the Fault Dependency Concept with Debugging Time Lag in Software Reliability Growth Modeling Using a Power Function of Testing Time," International Journal of Automation and Computing, Vol. 4, No. 4, pp. 359-368, Oct 2007.

[8] Han Seong Son, Hyun Gook Kang and Seung Cheol Chang, "Procedure for Application of Software Reliability Growth Models to NPP PSA," Journal of Nuclear Engineering and Technology, Vol. 41 No. 8,pp. 1065-1072, Oct 2009.

[9] Lev V. Utkin, Svetlana I. Zatenko and Frank P.A. Coolen, "Combining imprecise Bayesian and maximum likelihood estimation for reliability growth models," In Proc. of the Sixth International Symposium on Imprecise Probability: Theories and Applications, Durham, UK, 2009.

[10] N. Ahmad, S. M. K Quadri and Razeef Mohd, "Comparison of Predictive Capability of Software Reliability Growth Models with Exponentiated Weibull Distribution," International Journal of Computer Applications, Vol. 15, No. 6, pp. 40-43, Feb 2011.

[11] P. K. Kapur, H. Pham, Sameer Anand and Kalpana Yadav, "A Unified Approach for Developing Software Reliability Growth Models in the Presence of Imperfect Debugging and Error Generation," IEEE Transactions on Reliability, Vol. 60, No. 1, pp. 331-340, Mar 2011.

[12] S. M. K. Quadri, N. Ahmad and Sheikh Umar Farooq, "Software Reliability Growth modeling with Generalized Exponential testing –effort and optimal Software Release policy," Global Journal of Computer Science and Technology, Vol. 11, No. 2, pp. 27-42, Feb 2011.