Testing Effort Dependent Delayed S-shaped Software Reliability Growth Model with Imperfect Debugging

M.U. Bokhari
Department of Computer Science, Aligarh Muslim University, Aligarh, India
mubokhari@gmail.com

Md. Ashraf Siddiqui
Department of Computer Science, Aligarh Muslim University, Aligarh, India
ashrafkfu@gmail.com

N. Ahmad
University Department of Statistics and Computer Applications
T.M. Bhagalpur University, Bhagalpur, India
nesar_bgp@yahoo.co.in

***Abstract*— In software development process, testing is one of the most important aspects and hence, software reliably is very important factor of software systems. In the last four decades many software reliability growth model based on non-homogeneous Poisson process (NHPP) have been developed which incorporates testing effort function. However, the previous models are quite helpful for software engineers/ developers and commonly applicable in the industries and research institution. Still more testing-effort functions are required to incorporate into software reliability growth model. In this paper, we develop delayed S-shaped software reliability growth model with imperfect debugging which incorporates new modified Weibull testing–effort function (NMWTEF). We estimate the testing effort model parameters by least square and S-shaped software reliability growth parameters by maximum likelihood estimation techniques. We also present confidence interval of the software reliability growth parameters. Various software reliability measures are investigated through three numerical experiments. The numerical results are compared with other existing models in the literature. It is shown that the proposed Delayed S-shaped Software Reliability Growth Model with imperfect debugging with NMWTEF has a fairly better errors prediction capability.**

Keywords- Software Testing, Software Reliability Growth, Testing Effort Functions, Least Square estimation, Maximum Likelihood estimation

I. INTRODUCTION

Software reliability is one of the important factors of software quality. Before software delivered in to market it is thoroughly checked and errors are removed. Every software industry wants to develop software that should be error free. Software reliability growth models use measured trends of failure rates (or change in intervals between failures) and extrapolate them to future operation. In most cases, they evaluate the reduction in failure frequency during successive developmental test intervals to estimate the software reliability at the conclusion of the test (and sometimes into operational deployment).

Software reliability is defined according to as the probability of failure free operation of a computer program in a specified environment for a specified period of time. A software reliability growth model (SRGM) explains the time dependent behavior of fault removal. The objective of software reliability testing is to determine probable problems with the software design and implementation as early as possible to assure that the system meets its reliability requirements. Numerous SRGMs have been developed during the last three decades and they can provide very useful information about how to improve reliability (Musa et al., 1987; Xie, 1991; Lyu, 1996). Among these models, exponential growth model, inflection S-shaped and delayed S-shaped growth model have been shown to be very useful in fitting software failure data. Many authors has incorporated testing-effort into exponential type and inflection S-shaped SRGM based on the NHPP to get a better description of the fault detection phenomenon(Pham, 2007; Pham et al. 1999; Yamada et al., 1984; 1986; 1987; 1993; Kapur and Garg, 1996; Kapur and Younes, 1994; Huang et al., 1997; 2007; Kuo et al., 2001; Huang and Kuo, 2002; Huang, 2005; Bokhari and Ahmad, 2006; 2007; 2014; Quadri and Ahmad, 2010; Quadri et al., 2006; 2008; Ahmad et al., 2008; 2009; 2010; 2010a 2011). Some authors have incorporated testing-effort into delayed S-shaped SRGM.

This paper incorporates New Modified Weibull Testing–effort function (NMWTEF) into delayed S-shaped NHPP growth models (Ohba, 1984; 1984a). The parameters of the model are obtained by Least Square Estimation (LSE) and Maximum Likelihood Estimation (MLE) methods. We present the analysis of real data in this paper and the results are compared with other models from literature.

II. SOFTWARE RELIABILITY GROWTH MODEL WITH NMWTEF

SRGM help to measure and track the growth of reliability as software is being improved. There is an extensive body of literature on software reliability growth modeling, with many detailed probability models purporting to represent the probabilistic failure process. The objective of software reliability testing is to determine probable problems with the software design and implementations. Often SRGM may also yield information on physical properties of the code, such as the number of faults remaining in a software system, etc. In general, the exposure time over which reliability is being assessed may be expressed as calendar time, clock time, CPU execution time, number of test-runs, or some other suitable measures. Exponential Growth Model and delayed S-shaped growth model have been shown to be very useful in fitting real software data.

A. NMW Testing Effort Function

The testing-effort indicates how the errors are detected effectively in the software and can be modeled by different distributions (Musa *et al.,* 1987; Yamada *et al.,* 1986; 1993; Kapur *et al.*, 1999).

Testing Effort Function describes the relationship between the effort expended to test software (e.g., in person-months), and the physical characteristics of the software, such as LOC, exposure time (which can take many forms, and can be expressed either as total effort), etc.

The Cumulative NMWTEF expenditure consumed in *(0, t)* is depicted in the following (Quadri and Ahmad, 2010):

$$W(t) = \alpha \cdot (1 - e^{-\beta \cdot t^m . e^{\delta \cdot t}}), \alpha > 0, \beta > 0, m \geq 0, \delta > 0. \tag{1}$$

Therefore, the current NMWTEF expenditure attesting t is given by:

$$w(t) = \frac{dW(t)}{dt} = \alpha \cdot \beta \cdot (m + \delta \cdot t) \cdot t^{m-1} \cdot e^{\delta \cdot t} \cdot e^{-\beta \cdot t^m \cdot e^{\delta \cdot t}} \tag{2}$$

Where α, β, m, δ are constant parameters, α is the total amount of testing-effort expenditures; β and δ are the scale parameters, and m is shape parameter of NMW.

B. Delayed S-Shaped SRGM with NMWTEF

The Delayed S-shaped SRGM was originally proposed by Yamada et al. (1984).It was a simple modification of the NHPP to obtain an S-shaped growth curve for the cumulative number of failures detected. Software fault detection process can be viewed as a learning process that the software testers become familiar with the testing environments and tools as time progresses, these testers' skills gradually improve and then level off as the residual faults become more difficult to uncover. Because the original S-shaped model is for the analysis of fault isolation data, i.e. the testing process contains not only a fault detection process, but also a fault isolation process. Recently, Huang et al. (2007) modified delayed S-shaped SRGM and testing effort function has also been incorporated in the SRGM.

The extended delayed S-shaped model with NMWTEF is formulated on the following assumptions,

1. The fault removal process follows the NHPP.
2. The software system is subject to failures at random times caused by faults remaining in the systems.
3. The mean number of faults detected in the time interval *(t, t +Δt),* by the current testing effort is proportional to the mean number of remaining faults in the software.
4. The proportionality of fault detection is constant.
5. The mean number of faults isolated in the time interval (*t, t*+ Δ*t*) by the current testing effort is proportional to the current number of faults isolated in the software.
6. The proportionality of fault isolation is constant.
7. The consumption of testing-effort is modeled by NMWTEF.
8. Each time a failure occurs, the fault which caused it is immediately removed, and no new faults are introduced.

According to these assumptions, the extended Delayed S-Shaped SRGM can be formulated as:

$$\frac{dm_d(t)}{dt} \times \frac{1}{w(t)} = r_1 \times [a - m_d(t)] \tag{3}$$

and

$$\frac{dm_r(t)}{dt}\times\frac{1}{w(t)}=r_2\times\left[m_d(t)-m_r(t)\right] \tag{4}$$

If $r_2 \neq r_1$ then the solution of equations (3) and (4) under the boundary condition $m_d(0) = 0$ and $m_r(0) = 0$, are

$$m_d(t)=a\times\left(1-\exp\left[-r_1W(t)\right]\right) \tag{5}$$

and

$$m_r(t)=a\times\left\{1-\frac{r_1\exp\left[-r_2W(t)\right]-r_2\exp\left[-r_1W(t)\right]}{r_1-r_2}\right\} \tag{6}$$

Let $r_2 \cong r_1 \cong r$, then, the delayed S-shaped SRGM with TEF is given by

$$m(t)=a\left(1-\left(1+rW(t)\right)\exp\left[-rW(t)\right]\right) \tag{7}$$

In other way, if we suppose

$$r(t)=\frac{r^2t}{1+rt} \tag{8}$$

Now substituting in to

$$\frac{dm(t)}{dt}\times\frac{1}{w(t)}=r(t)\times\left[a-m(t)\right](a>0) \tag{9}$$

We get the following SRGM,

$$\frac{dm(t)}{dt}\times\frac{1}{w(t)}=\frac{r^2t}{1+rt}\times\left[a-m(t)\right] \tag{10}$$

Solving under the boundary condition $m(0) = 0$, obtain the same MVF as

$$m(t)=a\left(1-\left(1+rW(t)\right)\exp\left[-rW(t)\right]\right) \tag{11}$$

III. ESTIMATION OF MODEL PARAMETERS

The success of a software reliability growth model depends heavily on the quality of the failure data collected. The parameters of the Software Reliability Growth Model are estimated based upon these data. In order to validate the proposed model and to compare its performances with other existing models, experiments on actual software failure data will be performed. MLE and LSE techniques are used to estimate the model parameters (Musa *et al.,* 1987; Musa, 1999; Lyu, 1996). Sometimes, however, the likelihood equations may be complicated and difficult to solve explicitly. In that case one may have to solve with some numerical methods to obtain the estimates. On the other hand, Least Square Estimation (LSE), like Maximum Likelihood Method (MLE), is fairly general technique which can be applied in most practical situations for small or medium sample sizes and may provide better estimates (Musa *et al.*, 1987; Huang et al., 1997; Huang and Kuo, 2002). It minimizes the sum of squares of the deviations between what we expect and what we actually observe.

A. Least Square Estimation Method

The parameters $\alpha, \beta, m, and\, \delta$ in NMWTEF can be estimated by LSE method. These parameters are determined for n observed data pairs in the form (t_k, W_k) $(k=1,2,..,n; 0<t_1<t_2<_{t3}...<t_n)$. where $\hat{\alpha}, \hat{\beta}, \hat{m}, and\, \hat{\delta}$ can be obtained by minimizing (Bokhari and Ahmad, 2006; Ahmad et al. 2008; 2009; 2010; 2011) the following equations:

$$\mathrm{S}(\alpha,\beta,\mathrm{m},\delta)=\sum_{k=1}^{n}\left[W_k-W(t_k)\right]^2 \tag{12}$$

B. Maximum Likelihood Method

Once the estimates of α, β, m and δ are known, the parameters of the SRGMs can be estimated through Maximum Likelihood Estimation method. The estimators of a and r are determined for the n observed data pairs in the form (t_k, y_k) $(k=1,2,...,n;\ 0<t_1<t_2<......<t_n)$ where y_k is the cumulative number of software faults detected up to time t_k or $(0,t_k]$. Then the likelihood function for the unknown parameters a *and* r in the SRGM model (7) is given by (Musa et al. 1987).

$$L=\prod_{k=1}^{n}\frac{[m(t_k)-m(t_{k-1})]^{(m_k-m_{k-1})}}{(m_k-m_{k-1})!}.e^{-[m(t_k)-m(t_{k-1})]} \tag{13}$$

where, $m_0 \equiv 0$ and $t_0 \equiv 0$.

MLequations can be solved by numerical method to obtain the values of a and r.

C Confidence Interval Estimation

Now the two sided confidence interval is as follows:

The $(1-\alpha)100\%$ confidence limits for a and r is obtained as (Yamada and Osaki, 1985):

$$\hat{a}-z_{a/2}\sqrt{Var\left(\hat{a}\right)}\le a\le \hat{a}+z_{a/2}\sqrt{Var\left(\hat{a}\right)} \tag{14}$$

and

$$\hat{r}-z_{a/2}\sqrt{Var(\hat{r})}\le r\le \hat{r}+z_{a/2}\sqrt{Var(\hat{r})} \tag{15}$$

where $z_{\alpha/2}$ is the $(1-\alpha/2)$quartile of the standard normal distribution.

IV. Comparison Criteria

We can use the following three criteria to check the performance of proposed Software Reliability Growth Model with NMWTEF.

A. The Accuracy of Estimation (AE):

The Accuracy of Estimation is defined (Musa *et al.,* 1987; Kuo *et al.*, 2001) as

$$AE=\left|\frac{M_a-a}{M_a}\right| \tag{16}$$

Where M_a is the actual cumulative number of detected error after the test, and a is the estimated number of initial errors. For practical purpose, a M is obtained from software error tracking after software testing.

B. The Mean of Squared Errors (MSE): The Mean Square Errors defined as

$$MSE=\frac{1}{k}\sum_{i=1}^{i=k}\left[m\left(t_i\right)-m_i\right]^2 \tag{17}$$

where $m(t_i)$, is the expected number of errors at time t_i estimated by a model, and m_i is the expected number of errors at time t_i . MSE gives a quantitative comparison for long-term predictions. A smaller MSE indicates a minimum fitting error and better performance.

C. The Relative Error(RE): The Relative Error can be calculated by given formula:

$$\text{RelativeError}\left(\text{RE}\right)=\frac{\hat{m}\left(t_q\right)-q}{q} \tag{18}$$

Assume that we have observed q failures by the end of test time t_q. We use the failure data up to time $t_e(\le t_q)$ to determine the parameters of $m(t)$.

V. Data Analysis

In this section we are going to estimate the parameters of NMWTEF with real data sets. The models parameters are estimated by Least Square Estimation (LSE) and Maximum Likelihood Estimation (MLE) methods. The liner equation formed by LSE and MLE are solved numerically. Also, evaluate the different comparison criteria to check the performance of proposed software reliability growth models.

A. Data Set (DS1): This data is from the study of Ohba (1984). The system is PL/1 data base application software, consisting of approximately 1,317, 000 lines of code. During the nineteen weeks experiments, 47.65 CPU hours were consumed and about 328 software errors were removed. The study reports that the total cumulative number of detected faults after a long period of testing is 358. In order to estimate the parametersα, β, δ and m of the NMWTEF; we use the method of least squares. The following estimates are obtained

$$\hat{\alpha} = 64.4667, \hat{\beta} = 0.0341, \hat{\delta} = 0.885, \textit{ and } \hat{m} = 0.056 \quad (19)$$

Figure 1, shows the fitting of the estimated NMWTEF with the above estimates. The fitted curves and the actual software data are shown by solid and dotted lines, respectively.

Using the estimated parameters α, β, m and δ, the parameters a, r in (7) can be obtained by the MLE method. The estimated values of the parameters α and r with confidence interval are given by:

$$\hat{a} = 352.719, \hat{r} = 0.080 \quad (20)$$

Table I: Confidence Interval of Parameter Estimates for DS1

Parameter	Estimate	95 percent Confidence Interval	
		Lower bound	Upper limit
a	352.719	327.073	378.364
r	0.090	0.080	0.100

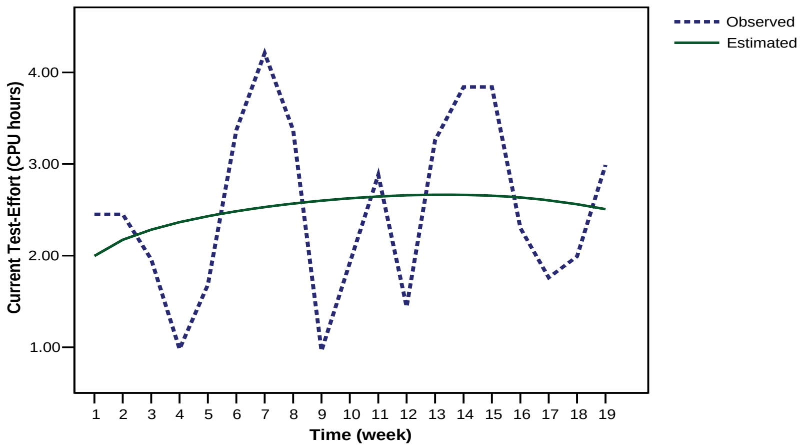

Fig. 1: Observed/estimated current testing-effort function vs. time.

Figure 2, illustrates a fitted curve of the estimated cumulative failure curve with the actual software data. The R^2 value for NMWTEF is 0.99674.Therefore, it can be said that the proposed curve is suitable for modeling the software reliability.

Table II presents the comparisons of proposed model with different SRGMs which reveal that the proposed model has a good performance.

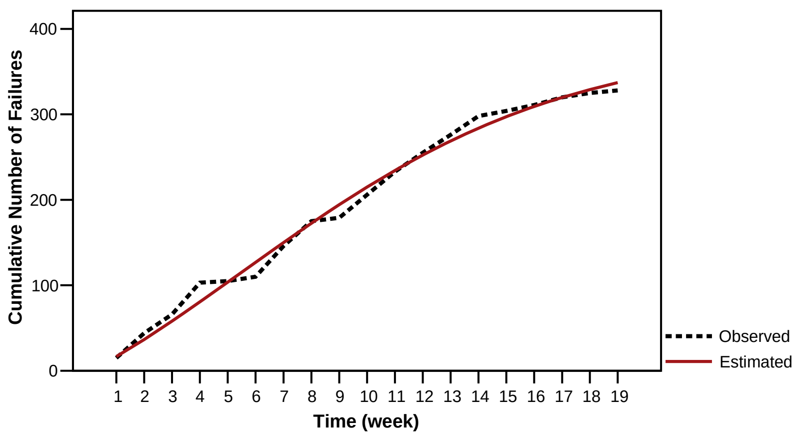

Figure 2: Observed/estimated cumulative number of failures vs. time

Table II: Comparison of different SRGMs for Data Set 1.

Model	***a***	***r***	***AE%***	***MSE***
Proposed model	352.72	0.09	1.7	197.4
Yamada delayed S-shaped model	374.05	0.1976	4.48	168.67
Delayed S-Shaped with Logistic TEF	346.55	0.0936	3.20	147.61
Huang Logistic model	394.08	0.0427	10.69	118.59
G-O model	760.0	0.0323	112.29	139.82

Furthermore, the relative error in prediction of proposed model for the data set is calculated and illustrated by Figure 3. It is shown that relative error approaches zero as $t_e$goes to. Therefore, Figure 2 to 3 and reveal that the proposed model has better performance than the other models.

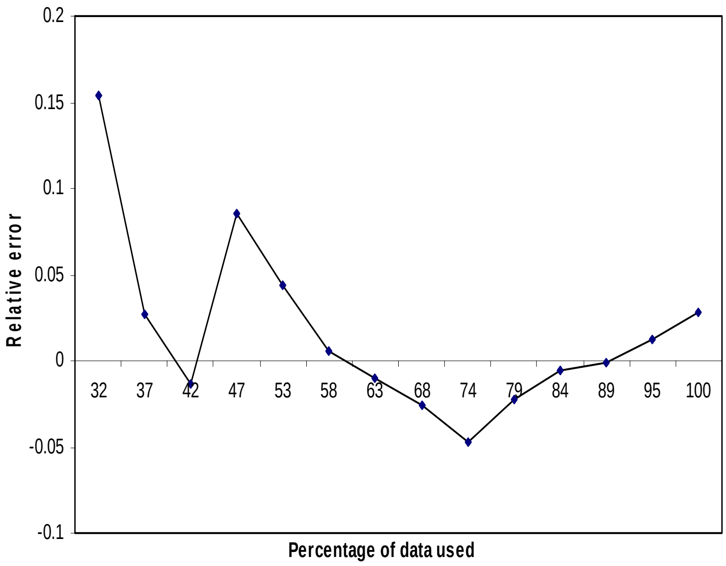

Figure 3: Predictive Relative Error Curve

A. Data Set Second (DS2)

The second set of actual data in this research is the System T1data of the Rome Air Development Center (RADC) projects and cited from Musa et al. (1987), Musa (1999). The number of object instructions for the system T1 which is used for a real-time command and control application. The size of the software is approximately 21,700 object instructions and developed by Bell Laboratories. The software was tested for twenty one weeks with 9 programmers. During the testing phase, about 25.3 CPU hours were consumed and 136 software errors were removed. The number of errors removed after 3.5 years of test was reported to be 188 (Huang, 2005). Similarly, parameters α, β, δ *and* m of NMWTEF for this data set can be obtained by using the method of LSE. The estimated values are

$$\hat{\alpha} = 25.587868, \hat{\beta} = 0.00009838, \hat{\delta} = 1.140123 \ and \hat{m} = 0.33891 \tag{21}$$

Using the estimated parameters α, β, δ *and* m , the parameters a, r in (7) can be obtained by the MLE method:

$$\hat{\alpha} = 124.824 \quad \hat{r} = 0.4 \tag{22}$$

Table III: Confidence interval of Parameter for DS2

Parameter	Estimate	95% Confidence Interval	
		Lower Bound	Upper Bound
a	124.824	112.530	137.118
r	0.400	0.296	0.505

Figure 4 shows the fitting of the NMWTEF whereas Figure 5 illustrates a fitted curve of the estimated cumulative number of failures. The fitted curve and the actual software data are shown by solid and dotted lines, respectively. The R^2 value for proposed NMWTEF is 0.9974. Therefore, it can be said that the proposed curve is suitable for modeling the software reliability.

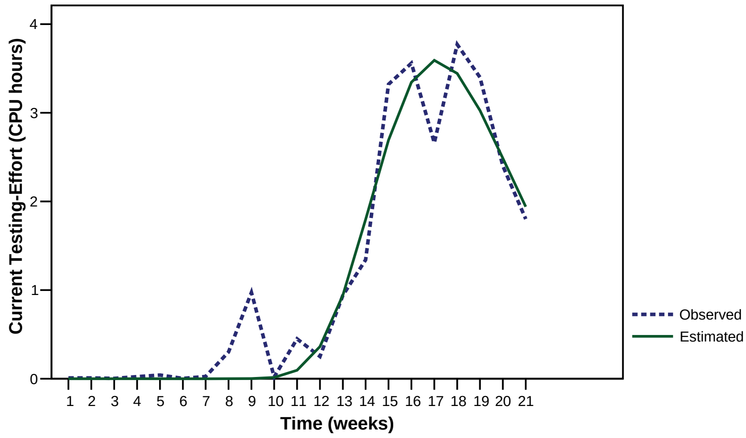

Figure 4: Observed/estimated current testing-effort function vs. time

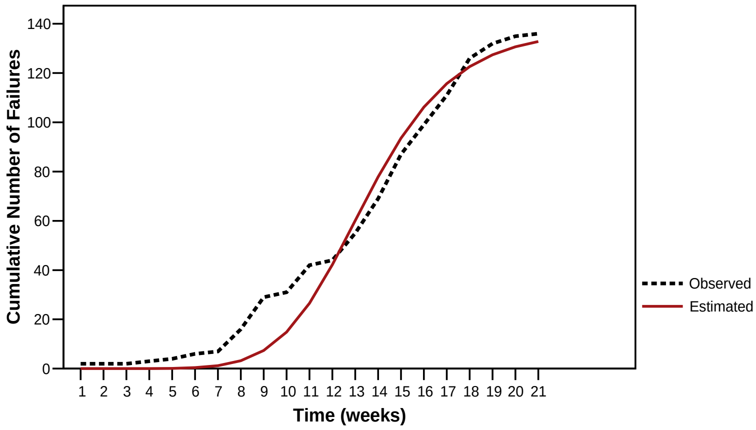

Figure 5: Observed/estimated cumulative number of failures vs. time

Table IV gives the comparisons of proposed model with different SRGMs which reveal that the proposed model has better performance. Kolmogorov Smirnov goodness-of-fit test shows that the proposed SRGM fits pretty well at the 5 percent level of significance.

Table IV. Comparison results of different SRGMs for DS2

Model	a	r	AE (%)	MSE
Proposed model	124.8	0.40	33.62	155.3
Yamada delayed S-Shaped model	237.19	0.0963	26.16	245.24
Delayed S-Shaped model with Logistic TEF	124.11	0.411	33.98	180.02
G-O model	142.32	0.1246	24.29	2438.3

Lastly, the relative error in prediction of proposed model for this data set is calculated and shown graphically in Figure 6. The relative error is plotted against the percentage of data used (that is, t_e / t_q). Consequently, from the Figures 4 to 6 and Table IV, it can be concluded that the proposed model gets reasonable prediction in estimating the number of software faults and fits the observed data better than the others.

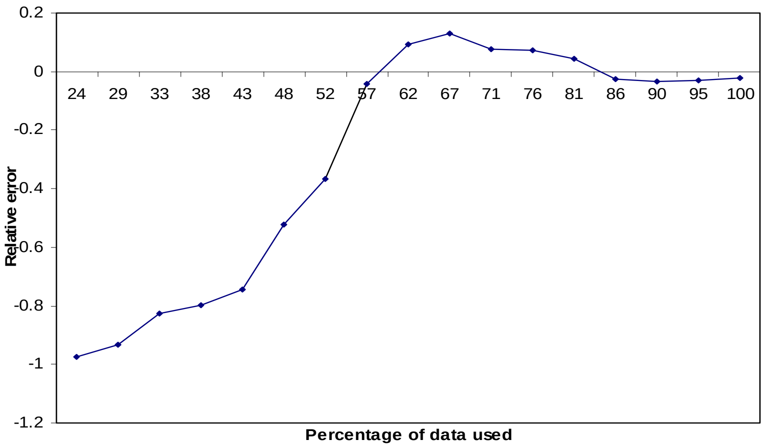

Figure 6: Predictive Relative Error Curve

VI. IMPERFECT DEBUGGING

Software Testing is very much important in assuring the quality of the software by identifying and removing faults in software, to make the software more efficient. But testing of the software for a long time may not ensure a bug free software and high reliability. Optimum amount of code also needs to be covered to make sure that the software is of good quality. Testing time alone may not give the correct picture of the number of faults removed in the software. The faults in the software may not be removed perfectly; this is mainly due to complexity of software or nature of testing team. This phenomenon is known as imperfect debugging. When the faults are not removed perfectly and leads to further generation of faults, this process is known as error generation (Pham et al. 1999).

In this section, we discuss a relaxation of the perfect debugging assumption. We modify the assumption (8) of section 3:

$$\frac{dm(t)}{dt}\times\frac{1}{w(t)}=r(t)\times\left[n(t)-m(t)\right] \tag{23}$$

$$n(t) = a + \Psi \times m(t)$$

$$\text{or} \quad \frac{dn(t)}{dt} = \Psi \frac{dm(t)}{dt} \tag{24}$$

$$\text{when} \quad r(t)=\frac{r^2 t}{1+rt}$$

Solving equation (23) using equation (24) under the boundary condition *m(0)=0, n(0)=a*, and *W(0)=0*, we obtain the delayed S-shaped MVF under imperfect debugging

$$m(t)=\frac{a}{1-\Psi}\left[1-\left(1+rW(t)\right)^{1-\Psi}\exp\exp\left(-r(1-\Psi)W(t)\right)\right] \tag{25}$$

A. *Illustrated Example:* From the above imperfect debugging model, summary of estimated parameters with their 95% confidence interval for PL1(DS1) are presented in the Table V:

Table V. Parameter Estimates under Imperfect debugging

Parameter	Estimate	95% Confidence Interval	
		Lower Bound	Upper Bound
a	7.746	-11.918	27.410
Ψ	0.984	0.940	1.028
r	1.688	-2.135	5.510

Table VI shows the estimated parameters of the proposed SRGM and some selected models for comparison under imperfect debugging. It also gives the value of MSE. We observed that the value of MSE of the proposed SRGM with NMW testing-effort function is the lowest among all the models considered. Moreover, the estimated values Ψ of all the models is close to but not equal to zero, thus the error removal phenomenon may not be pure perfect debugging process. A fitted curve of the estimated cumulative number of failures with the actual software data and the RE curve for the proposed SRGM with NMWTEF under imperfect debugging is illustrated by Figure 7 and 8.

Table VI. Comparison with different SRGMs under Imperfect Debugging

Models	***a***	***r***	Ψ	MSE
Proposed model	7.746	1.688	0.984	91.04
Huang Logistic model	391.62	4.20×10^{-2}	1.16×10^{-2}	114.09
Huang Rayleigh model	399.02	3.16×10^{-2}	1.23×10^{-1}	268.55
Delayed S-Shaped model with logistic TEF	335.39	1.24×10^{-1}	1.15×10^{-2}	634.60
Delayed S-Shaped model with Rayleigh TEF	346.09	9.88×10^{-2}	1.39×10^{-2}	880.49
Extended Goel-Okumoto model	365.85	7.53×10^{-2}	2.87×10^{-1}	222.09

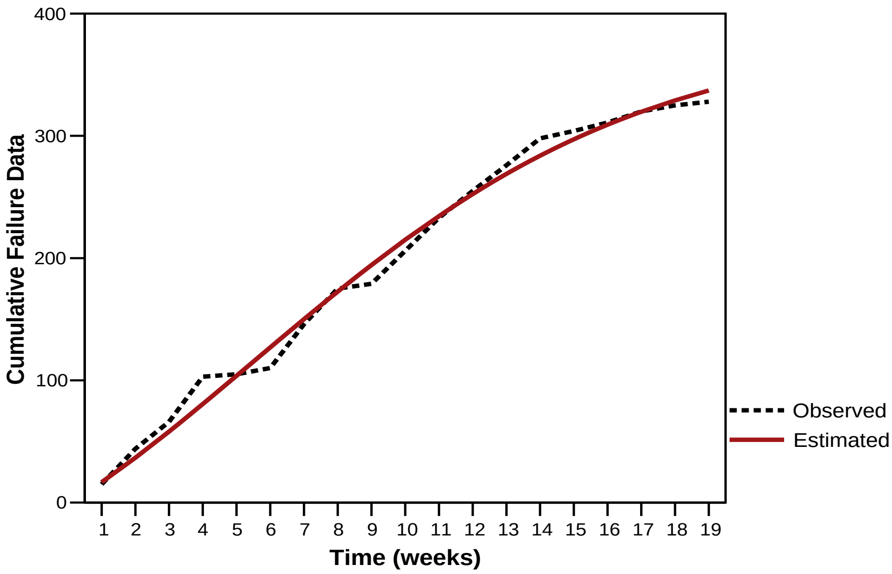

Figure 7: Observed/estimated cumulative number of failures v. time

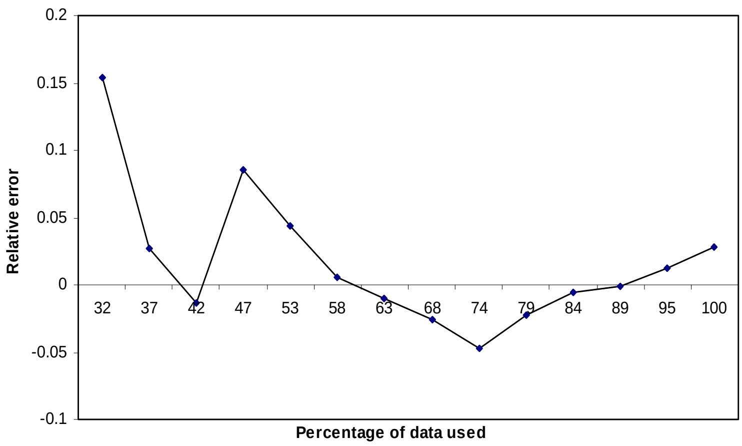

Figure 8: Predictive Relative Error Curve

VII. CONCLUSION

This paper proposed a SRGM based on NHPP model, which incorporates NMW testing-effort function into delayed S-shaped model. The performance of the proposed SRGM is compared with other existing SRGMs using different criteria. The results show that the proposed model has good performance, better fit and wider applicability based on two real data applications. We conclude that the incorporated NMWTEF into delayed S-shaped model is a flexible and can be used to describe the actual expenditure patterns more faithfully during software development.

REFERENCES

[1] Ahmad, N., Bokhari, M.U., Quadri, S.M.K. and Khan, M.G.M. (2008), "The Exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy: a performance analysis", International Journal of Quality and Reliability Management, Vol. 25 (2), pp. 211-235.

[2] Ahmad, N., Khan, M. G. M., Quadri, S. M. K. and Kumar, M. (2009), "Modeling and analysis of software reliability with Burr type X testing-effort and release-time determination", Journal of Modelling in Management, Vol. 4 (1), pp. 28-54.

[3] Ahmad, N.,Khan, M. G. M and Rafi, L. S. (2010), "A Study of Testing-Effort Dependent Inflection S-Shaped Software Reliability Growth Models with Imperfect Debugging", International Journal of Quality and Reliability Management, Vol. 27 (1), 89-110.

[4] Ahmad, N.,Khan, M. G. M and Rafi, L. S. (2010a), "Software Reliability Modeling Incorporating Log-Logistic Testing-Effort with Imperfect Debugging", in Proceedings of the International Conference on Modeling, Optimization and Computing (ICMOC-2010), Durgapur, India, Published by American Institute of Physics, pp. 651-657.

[5] Ahmad, N., Khan, M. G. M and Rafi, L. S. (2011), "Analysis of an Inflection S-shaped Software Reliability Model Considering Log-logistic Testing-Effort and Imperfect Debugging", International Journal of Computer Science and Network Security, Vol. 11 (1), pp. 161 – 171.

[6] Bokhari, M.U. and Ahmad, N. (2006), "Analysis of a software reliability growth models: the case of log-logistic test-effort function", In: Proceedings of the 17th IASTED International Conference on Modeling and Simulation (MS'2006), Montreal, Canada, pp. 540-545.

[7] Bokhari, M.U. and Ahmad, N. (2007), "Software reliability growth modeling for Exponentiated Weibull functions with actual software failures data", Advances in Computer Science and Engineering: Reports and Monographs, World Scientific Publishing Company, Singapore, Vol. 2, pp. 390-396.

[8] Bokhari, M. U. and Ahmad, N. (2014), "Incorporating Burr Type XII Testing-efforts into Software Reliability Growth Modeling and Actual Data Analysis with Applications", Journal of Software, Vol. 9 (6), pp. 1389-1400.

[9] Goel, A.L. and Okumoto, K. (1979), "Time dependent error-detection rate model for software reliability and other performance measures", IEEE Transactions on Reliability, Vol. R- 28, No. 3, pp. 206-211.

[10] Huang, C.Y. (2005), "Performance analysis of software reliability growth models with testing-effort and change-point", Journal of Systems and Software, Vol. 76, pp. 181-194.

[11] Huang, C.Y. and Kuo, S.Y. (2002), " Analysis of incorporating logistic testing-effort function into software reliability modeling", IEEE Transactions on Reliability, Vol. 51, no. 3, pp. 261-270.

[12] Huang, C.Y., Kuo, S.Y. and Lyu, M.R. (2007), "An assessment of testing-effort dependent software reliability growth models", IEEE Transactions on Reliability, Vol. 56, no.2, pp. 198-211.

[13] Kapur, P.K., Garg, R.B. and Kumar, S. (1999), Contributions to Hardware and Software Reliability, World Scientific, Singapore.

[14] Kapur, P.K. and Younes, S. (1994), "Modeling an imperfect debugging phenomenon with testing effort", In: Proceedings of 5th International Symposium on Software Reliability Engineering (ISSRE'1994), pp. 178-183.

[15] Kumar, M., Ahmad, N. and Quadri, S.M.K. (2005), "Software reliability growth models and data analysis with a Pareto test-effort", RAU Journal of Research, Vol., 15 (1-2), pp. 124-128.

[16] Kuo, S.Y., Hung, C.Y. and Lyu, M.R. (2001), "Framework for modeling software reliability, using various testing-efforts and fault detection rates", IEEE Transactions on Reliability, Vol. 50, no.3, pp 310-320.

[17] Lyu, M.R. (1996), Handbook of Software Reliability Engineering, McGraw- Hill.

[18] Musa, J.D., Iannino, A. and Okumoto, K. (1987), Software Reliability: Measurement, Prediction and Application, McGraw-Hill.

[19] Ohba, M. (1984), "Software reliability analysis models" IBM Journal. Research Development, Vol. 28, no. 4, pp. 428-443.

[20] Ohba, M. (1984a), "Inflection S-shaped software reliability growth models", Stochastic Models in Reliability Theory (Osaki, S. and Hatoyama, Y. Editors), pp. 144-162, Springer-Verlag Merlin.

[21] Pham, H. (2007), "An imperfect-debugging fault-detection dependent-parameter software", International Journal of Automation and Computing, Vol. 4, no. 4, pp. 325-328.

[22] Pham, H., Nordmann L. and Zhang, X. (1999), "A general imperfect software debugging model with S-shaped fault detection rate", IEEE Transactions on Reliability, Vol. R-48, no. 2, pp. 169-175.

[23] Quadri, S.M.K., Ahmad, N., Peer, M.A. and Kumar, M. (2006), "Non-homogeneous Poisson process software reliability growth model with generalized exponential testing effort function", RAU Journal of Research, Vol., 16 (1-2), pp. 159-163.

[24] Quadri, S. M. K. and Ahmad, N. (2010), "Software Reliability Growth Modeling with New Modified Weibull Testing-effort and Optimal Release Policy", International Journal of Computer Applications, Vol. 6 (12), pp.1–10, September 2010. Published by Foundation of Computer Science.

[25] Xie, M. (1991), Software Reliability Modeling, World Scientific Publication, Singapore.

[26] Yamada, S., Hishitani J. and Osaki, S. (1993), "Software reliability growth model with Weibull testing-effort: a model and application", IEEE Transactions on Reliability, Vol. R-42, pp. 100-105.

[27] Yamada, S., Ohtera, H. and Norihisa, H. (1986), "Software reliability growth model with testing-effort", IEEE Transactions on Reliability, Vol. R-35, no. 1, pp.19-23.

[28] Yamada, S., and Osaki, S. (1985), "Software reliability growth modeling: models and applications", IEEE Transaction on Software Engineering, Vol. SE-11, no. 12, pp. 1431-1437.

[29] Yamada, S., Ohba, M., and Osaki, S. (1984) "S-shaped software reliability growth models and their applications," IEEE Trans. Reliability, vol. R-33, no. 4, pp. 289–292.

AUTHORS PROFILE

Prof. Mohammad Ubaidullah Bokhari is presently working as a Professor in the Department of Computer Science, AMU, Aligarh, (INDIA) and Principal Investigator (PI) of the ambitious project, NMEICT ERP Mission Project (Govt. of India). He worked as Chairman Department of Computer Science from Jan 2010 to Dec 2012 and March 2014 to March 2017. He has also worked as an Associate Professor and Director of Studies in Australian Institute of Engineering & Technology, Victoria, Melbourne (Australia). He has a vast teaching experience of more than twenty three years. He obtained his doctorate in the field of Software Reliability and Master's degree in computer science from Aligarh Muslim University. During his academic pursuit he has visited different countries like Australia, France, Canada, South Africa, etc. for research purpose and acquiring academic excellence. He is recipient of Merit Scholarship from high school to graduation level. He was awarded prestigious Australian Postgraduate Award (Industry) Scheme Scholarship in 2004. His research interests include Software Reliability, Security and Cryptography, Software Engineering, Database Management System, Artificial Intelligence, Image Processing and E-Learning. He has published more than 85 research papers in reputed National and International Journals and Conferences. He has also authored three (03) books on different areas of computer science. Prof. Bokhari is lifetime member of Computer Society of India (C.S.I.) and member of IEEE. He has the privilege of starting the PhD program in the Department of Computer Science, under his supervision 4 research scholars completed his Doctorate Degree and more than 5 scholars are currently working.

Md Ashraf Siddiqui is research scholar in the Department of Computer Science, A.M.U. Aligarh. He received the B. Sc. and M. Sc. degree in Physics from A.M.U. Aligarh, in 1989 and 1991 respectively. And further MCA degree from Aligarh Muslim University, Aligarh, in 1995. He joined King Faisal University, Hofuf, Saudi Arabia as a lecturer in 2003 and worked till 2014. He has worked in various origination as a computer professional from 1995 to 2003. His research interests include reliability analysis, software testing, software reliability engineering and optimization Techniques. He has about 19 years of experience in teaching and software development.

Prof. Nesar Ahmad is a University Professor in the University Department of Statistics and Computer Applications and Director/Course Coordinator of University Department of Computer Applications, Computer Centre at Tilka Manjhi Bhagalpur University, Bhagalpur, India. He received the B. Sc. degree in Mathematics from Bihar University, Muzaffarpur, in 1984 and the M. Sc., M. Phil., and Ph. D. degree in Statistics from Aligarh Muslim University, Aligarh, in 1987, 1990, and 1993, respectively. He joined the University Department of Statistics and Computer Applications at Tilka Manjhi Bhagalpur University, Bhagalpur, in 1996. From 1995 to 1996 he was a research associate of UGC/CSIR at Aligarh Muslim University, Aligarh. He has been a Lecturer at the University of the South Pacific, Suva, Fiji Islands from January 2006 to December 2009. He has about 22 years of experience in teaching and research. His research interests include life testing, statistical modeling, reliability analysis, software testing, software reliability engineering and optimization Techniques. He has published more than 80 papers in journals, and conferences in these areas. Prof. Ahmad is an executive member of Indian Society of Information Theory and Its Applications (ISITA) and life member of Indian Science Congress. He is in the editorial board of International Journal of Scientific and Statistical Computing (IJSSC) and Journal of Convergence Information Technology (JCIT).