

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

# INFORMATION TECHNOLOGY JOURNAL

**ANSI***net*

Asian Network for Scientific Information  
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

## Prediction Capability Analysis of Two and Three Parameters Software Reliability Growth Models

D.R. Prince Williams

Department of Information Technology, Salalah College of Technology, Salalah, Sultanate of Oman

**Abstract:** This study aims to analyze the predict capability of some of the popular Software Reliability growth models to understand the potential of having two or three parameters to facilitate the estimation process. The predict validity analysis will be on two key factors, one pertaining to the degree of fitment on available failure data and the other for its prediction capability. The validity analysis will be to arrive at a trade-off in choosing a simple model as compared to a complex model by determining their performances across multiple data sets. Data for the predict validity analysis has been taken from different time periods to understand the impact of these models across various technologies and processes used during that timeframe.

**Key words:** Software reliability, prediction, mean value function, error detection rate, estimation of parameters

### INTRODUCTION

Many Software Reliability Models have been developed, in the last many years, each having its own strength in facilitating reliability estimations (Yang and Xie, 2000). These models make use of two or three parameters to get the best fit against the actual failure data. The design of such models and the need for two or three parameters is based on the expectation of the trend in the failure data. Some of the assumptions made to identify such trends are as follows:

- Faults are initially rapidly identified but eventually reaches a steady state.
- Fault identification is initially slow, later rapidly increases and finally converges to a steady state.
- Fault identification initially has a steady growth then rapidly converges towards a steady state.

Each of the above trends can be true depending on one or more of the following issues.

- The testing teams performance capability
- Complexity/Size of the application domain
- The technology in use
- The software development process followed

The number of faults identified during a test cycle depends on the experience of the tester. A less experienced tester may not identify as many faults as an experienced tester. Also, when the application complexity and size increases, the pressure to perform a rigorous test/debug within a specific time period could lead to creating more faults or ignoring existing ones.

Change in the technology, has led to many changes being introduced in the software development process. During earlier times, the SSAD (Structured System Analysis and Design) technique was used as the popular approach to software development. Later, when distributed systems became popular and with the advent of distributed objects, OOAD (Object Oriented Analysis and Design) technique is used. So the testing/debug strategy also changes as there is more program integration required in OOAD based development as compared to the SSAD based development. Today, with the internet technology in place the ability to integrate heterogeneous systems, viz., legacy systems, client-server systems, new technology systems etc., across remote locations have become a possibility. In such a scenario, when old and new systems co-exist, it will be interesting to see the performance capability of software reliability growth models.

Software reliability growth models have been developed across various time periods. The complexity of the models, by having two or more parameters, has also increased to cater to the change in need with the change in time. In this study, we focus on the validity of reliability predictions of two and three parameter software reliability growth models. Also, the aim is to do the analysis on multiple data inputs, taken from the early 80's to the 90's. The approach is to study the degree of fitment using a specific percentage of the available failure data and to validate the predictive performance of each model using the balance percentage of failure data.

### Notation

a : Expected number of faults in the software when the testing begins.

- b : Fault detection rate per remaining fault in the software.
- m(t) : Expected number of observed failures during the time interval (0,t).
- R(x/s) : Software reliability.
- c : Inflection parameter.
- $\lambda(t)$  : Failure intensity function
- z<sub>i</sub> : Cumulative number of failures up to time t<sub>i</sub>

$$m(t) = a(1 - e^{-bt}) \tag{2}$$

**Logarithmic poisson model:** This model, proposed by Musa and Okumoto (1984), has its mean value function as given below:

$$m(t) = a \ln(1 + bt) \tag{3}$$

This model is classified as an infinite failure model as it is assumed that there is no upper bound to the number of failures.

**Three parameter models:** The following SRGM's, designed using three parameters, were selected for the performance analysis:

- Imperfect Debugging Model
- Inflection S-shaped Growth Model
- Logistic Model

**Imperfect debugging model:** Imperfect debugging occurs when the error debugging process does not lead to the removal of the software error. Though this concept was first introduced in the J-M model, we have, for present study, taken the same introduced in the Goel-Okumoto model by Kapur *et al.* (1990).

The mean value function for this model is given below:

$$m(t) = \frac{a}{p}(1 - e^{-pbt}) \tag{4}$$

where p is the probability of perfect debugging.

**Inflection s-shaped growth model:** The basic concept in this growth model (Obha, 1984a) is that the observed software reliability growth becomes S-shaped if faults in a program are mutually dependant. The mean value function for this model is given below:

$$m(t) = a \frac{1 - e^{-bt}}{1 + c e^{-bt}} \tag{5}$$

where c is the inflection parameter and is given by  $\frac{1-r}{r}$ , (r > 0), where r is the inflection rate.

**Logistic growth model:** This is another popular model (Yamada *et al.*, 1983) used by many software houses in Japan. The software failures are assumed to follow a logistic curve. The mean value function for this model is given below:

### SOFTWARE RELIABILITY GROWTH MODELS (SRGMS)

Six different software reliability growth models were selected for performing the predict capability analysis. The common assumptions made in these models are given below:

- The software system is subject to failure at random times caused by software faults
- There are no failures at time t = 0
- Failure intensity is proportional to the residual fault content
- Faults located are immediately corrected and will not appear again

**Two parameter models:** The following three SRGM's, each having a specific characteristic and designed using two parameters, were selected for the prediction capability analysis:

- Delayed S-shaped Growth Model
- Exponential Model
- Logarithmic Poisson Model

**Delayed s-shaped growth model:** The mean value function for this model (Yamada *et al.*, 1983) is given below:

$$m(t) = a(1 - (1 + bt)e^{-bt}) \tag{1}$$

One specific advantage of using this model is that it is designed for fault isolation data analysis. Fault isolation means that some of the failures can be intentionally reproduced, leading to the identification of the fault and its removal.

**Exponential growth model:** Though there are a few variants of the exponential growth model, the model proposed by Goel and Okumoto (1979) has been taken for this study. The failure data is assumed to take an exponential curve.

The mean value function for this model is given below:

$$m(t) = \frac{a}{1 + ce^{-bt}} \quad (6)$$

where c is an inflection parameter

**PREDICTION CAPABILITY**

**Approach:** The prediction capability of each model is analyzed using four different data sets. Each data set is of a different size and from different time periods (80's and 90's). The four data sets chosen have been taken from Yamada *et al.* (1983), Kapur *et al.* (1990), Musa and Okumoto (1984) and Obha (1984b), respectively. The predict validity process consists of the following steps:

- Estimation of the parameters of each model, using 80% of the failure data
- Model analysis:
  - Goodness of Fit for all models using the first 80% of the failure data
  - Comparison of prediction capability of the models selected by validating against the last 20% of the available failure data
- Model ranking

**Parameter estimation:** The model parameters for each of the models selected were estimated by maximizing the log likelihood function Obha (1984a) and is as follows:

$$\ln L = \sum_{i=1}^n (z_i - z_{i-1}) \ln [m(t_i) - m(t_{i-1})] - \sum_{i=1}^n \ln ((z_i - z_{i-1})!) - m(t_n) \quad (7)$$

The respective mean value function of each model is substituted in the above equation. The substituted function is then differentiated with respect to the number of parameters, to obtain the specific parameter based equations. These equations are set to zero and solved to arrive at the respective parameter values.

The details of the parameters estimated for each Model using published failure data sets are given in the Table 1-4.

**Data Set 1:** Yamada *et al.* (1983):

Number of Failures : 19

Number of Failures identified for estimation = 0.08\*19 = 15

**Data Set 2:** Kapur *et al.* (1990):

Number of Failures : 20

Number of Failures identified for estimation = 0.08\*20 = 16

Table 1: Parameters estimation of data set 1

Model	a	b	3rd parameter
Delayed S-shape	382.133	0.198	
Exponential	1542	0.015	
Logarithmic poisson	1465	0.015	
Imperfect debug	1156	0.02	p = 0.075
Inflection S-shape	454.267	0.133	r = 0.319
Logistic	332.86	0.336	m = 14.756

Table 2: Parameters estimation of data set 2

Model	a	b	3rd parameter
Delayed S-shape	105.461	0.270	
Exponential	138.09	0.077	
Logarithmic poisson	79.066	0.153	
Imperfect debug	103.567	0.103	p = 0.75
Inflection S-shape	137.203	0.079	r = 0.979
Logistic	101.741	0.348	p = 9.985

Table 3: Parameters estimation of data set 3

Model	a	b	3rd parameter
Delayed S-shape	5110	0.113	
Exponential	5808	0.041	
Logarithmic poisson	3277	0.075	
Imperfect debug	4356	0.054	p = 0.75
Inflection S-shape	5181	0.087	r = 0.359
Logistic	5000	0.147	m = 10.189

Table 4: Parameters estimation of data set 4

Model	a	b	3rd parameter
Delayed S-shape	561.864	0.05	
Exponential	731.825	0.014	
Logarithmic poisson	543.818	0.019	
Imperfect debug	548.869	0.019	p = 0.75
Inflection S-shape	545.205	0.058	r = 0.164
Logistic	531.819	0.08	m = 15.966

**Data Set 3:** Musa and Okumoto (1984):

Number of Failures : 59

Number of Failures identified for estimation = 0.08\*59 = 47

**Data Set 4:** Obha (1984b):

Number of Failures : 109

Number of Failures identified for estimation = 0.08\*109 = 87

Most of the above computations were done using a software tool (MathCAD).

**Model analysis:** The performance of each model was analyzed by examining the Goodness of Fit' and Predictive capability of each model. The first 80% of the failure data was used to derive the degree of fitness of each the model. The remaining 20% of failure data was then predicted using the estimated parameters. The validation of fitness and prediction capability of each model was measured by calculating their respective 'Sum of Square Errors' (SSE) values. The details are given in Table 5-8.

From the Table 5 it can be seen that the 3 parameter inflection s-shaped model has the best fit amongst all the models while the 2 parameter exponential

Table 5: SSE of the models using data set 1

SSE/MDL	Del'S'hape	Expo'tial	Log'Po'sn	Impl'Debug	Inf S'hape	Logistic
Fit	3462	1276	1613	1268	1151	2639
Predict	237	5606	2362	5562	1494	28

Table 6: SSE of the models using data set 2

SSE/MDL	Del'S'hape	Expo'tial	Log'Po'sn	Impl'Debug	Inf S'hape	Logistic
Fit	551	147	165	148	146	175
Predict	0.70	17.91	29.34	19.18	17.94	0.31

Table 7: SSE of the models using data set 3

SSE/MDL	Del'S'hape	Expo'tial	Log'Po'sn	Impl'Debug	Inf S'hape	Logistic
Fit	943	976	3023	1023	208	1646
Predict	67900	65436	577063	42050	33034	175983

Table 8: SSE of the models using data set 4

SSE/MDL	Del'S'hape	Expo'tial	Log'Po'sn	Impl'Debug	Inf S'hape	Logistic
Fit	23189	84630	81837	76690	10732	15941
Predict	1513	10705	50677	14449	435	81

model comes a close second. But in terms of prediction, the 3 parameter logistic model is way ahead as compared to the other models while the 2 parameter delayed-s-shape model comes second. It is followed by the inflection-s-shape and the logarithmic poisson model while the implicit-debug and exponential model, have, relatively, the least prediction capability.

From the Table 6 it can be seen that the 3 parameter inflection s-shaped model, the implicit-debug model and the 2 parameter exponential model have the same fitment characteristics. But in terms of prediction, the 3 parameter Logistic model is the leader very closely followed by the 2 parameter delayed s-shape model. In fact, the difference is negligible.

From the Table 7 it can be seen that the 3 parameter Inflection S-shaped model, has the best fit amongst all the models. It is followed by the 2 parameter delayed s-shaped model and the exponential model. In terms of prediction, it is the 3 parameter inflection s-shaped model, again, that has relatively the best capability. It is followed by the implicit-debug, delayed s-shape and the exponential model. The logarithmic poisson model, relatively, has the least prediction capability.

From the Table 8 data, it can be seen that the 3 parameter Inflection s-shaped model, the implicit-debug model and the 2 parameter exponential model have the same fitment characteristics. But in terms of prediction, the 3 parameter logistic model is the leader very closely followed by the 2 parameter delayed s-shape model. In fact, the difference is negligible.

**Ranking:** Figure 1 and 2 provide a visual presentation on the relative ranking between all the models analyzed. Figure 1 presents the ranking of each model in terms of its degree of fitness with actual failure data. Figure 2 presents the ranking of each model in terms of its short term prediction capability.

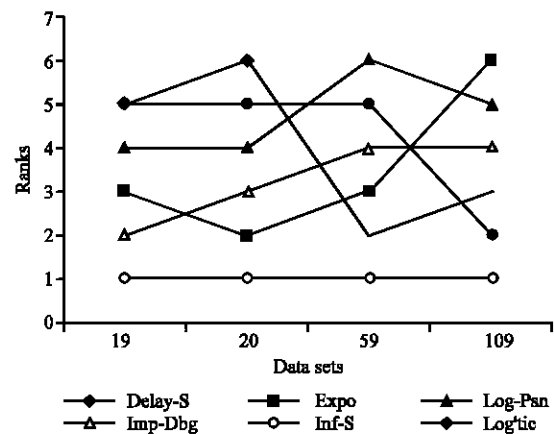


Fig. 1: Fitment of the models

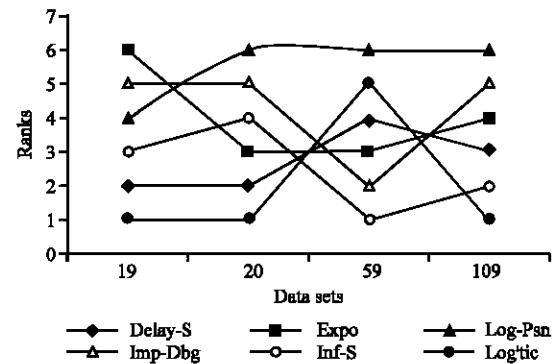


Fig. 2: Prediction of the models

**Goodness of fit**

- Three Parameter Models
  - The inflection s-shape model has the best fit amongst all the models analyzed.
  - Logistic model fitment improves as the failure data volume increases

- Implicit-debug model fitment decreases as the data volume increases
- Two Parameter Models
  - The delayed s-shape model has the best fit amongst all the two parameter models analyzed.
  - Exponential model fitment decreases as the failure data volume increases
  - Logarithmic poisson model fitment decreases as the data volume increases

### Prediction capability

- Three Parameter Models
  - The logistic model shows the best overall prediction capability across all models
  - Inflection s-shaped model prediction tends to improve as the failure data volume increases
  - Implicit-debug model prediction capability is about average
- Two Parameter Models
  - The delayed s-shape model has the best prediction capability amongst all the two parameter models
  - Exponential model's prediction capability has a tendency to improve as the failure data volume increases
  - Logarithmic poisson model prediction capability decreases as the data volume increases

### CONCLUSIONS

Though the three parameter Logistic model shows the best overall failure data prediction capability performance, the two parameter Delayed S-shaped model can be considered to be a close second. This is because the prediction variation between these two models when compared to actual failure data, for all failure data sets used, is less than 5%. In the case of failure dataset 3, the

delayed s-shaped model shows better performance. It has a prediction deviation of less than 3% while for the Logistic model it is 4%. Since the specific need for a project manager is a suitable software reliability growth model that is easy to implement and provides good failure data prediction, the delayed s-shaped model can be considered to meet this requirement.

### REFERENCES

- Goel, A.L. and K. Okumoto, 1970. Time dependent error detection rate model for software reliability and other performance measure. IEEE Trans. Reliab., 28: 206-211.
- Kapur, P.K. *et al.*, 1990. Optimal software release policies for software reliability growth models under imperfect debugging. RAIRO: Operations Res., 24: 295-305.
- Musa, J.D. and K. Okumoto, 1984. A logarithmic Poisson execution time model for software reliability measurement. Proceedings of the 7th International Conference on Software Engineering, pp: 230-238.
- Obha, M., 1984a. Inflection S-Shaped Software Reliability Growth Model. Stochastic Models in Reliability Theory. Springer-Verlag, Berlin, pp: 144-162.
- Obha, M., 1984b. Software reliability analysis models. IBM. J. Res. Deve., 28: 428- 443.
- Yamada, S., M. Obha and S. Osaki, 1983. S-Shaped reliability growth modeling for software error detection. IEEE Trans. Reliability, 12: 475-478.
- Yamada, S. *et al.*, 1992. Software reliability measurement and assessment based on non homogenous poisson process models: A survey. Micro Reliab., 32: 1763-1773.
- Yang, B. and M. Xie, 2000. A study of operational and testing reliability in software reliability analysis. Reliability Eng. Sys. Safety, 70: 323-329.