

M. Xie, G. Y. Hong and C. Wohlin, "A Study of Exponential Smoothing Technique in Software Reliability Growth Prediction", *Quality and Reliability Engineering International*, Vol.13, pp. 247-353, 1997.

# **A STUDY OF EXPONENTIAL SMOOTHING TECHNIQUE IN SOFTWARE RELIABILITY GROWTH PREDICTION**

M.Xie, G.Y. Hong  
Industrial & Systems Engineering Department  
National University of Singapore, Singapore

and C. Wohlin  
Dept of Communication Systems  
Lund University, Lund, Sweden

## **ABSTRACT**

Software reliability models can provide quantitative measures of the reliability of software systems which are of growing importance today. Most of the models are parametric ones which rely on modelling the software failure process as a Markov or nonhomogeneous Poisson process. It has been noticed that many of them do not give very accurate prediction of the software failure process as the focus is on the fitting of past data. In this paper, we study the use of double exponential smoothing technique to predict software failures. The proposed approach is a non-parametric one and it has the ability of providing more accurate prediction compared with traditional parametric models. It is very easy to use and it requires very limited amount of data storage and computational effort. It can be updated instantly without much calculation. Hence, it is a tool that should be more commonly used in practice. Numerical examples are shown to highlight the applicability. Comparisons with other commonly used software reliability growth models are also presented.

**Key words:** Software reliability, double exponential smoothing, reliability prediction, repairable systems, model comparison

## 1. Introduction

As computer systems are widely utilized in various areas including those with safety-critical functions, producing reliable software is of increasing importance. For the past two decades, many reliability models have been developed, see Musa et al. (1987), Xie (1991) and Lyu (1996). These models enable software developers to evaluate software reliability in a quantitative manner. Software reliability models can assist during software testing where it could be used to track reliability improvements and to provide an aid for making decisions such as on when testing can be terminated and how much additional resource is need for the testing.

Traditional software reliability models, such as those well-known Goel-Okumoto model (Goel and Okumoto, 1979) and the s-shaped model (Yamada et al., 1983) have shown to be very useful in fitting software failure data. However, the predictive ability is questionable as the focus of such models are the fitting of the past data and it heavily relies on the early information available. In software development, the prediction of future failure behaviour is much more important. Although plenty of literature, see e.g. Keiller and Miller (1991) and Littlewood (1980, 1987), can be found on this issue, little study has been published in using other methods that have the potential of providing better predictive performance.

Exponential smoothing is a forecasting technique which is based on adjusting an early forecast with the latest observation. In statistical analysis of time series data, this technique has shown surprising accuracy with minimal computational effort. The model components and parameters have some intuitive meaning to the users. Only limited data storage and computational effort are required. Hence, for a more accurate prediction of future software failure behaviour, this is a technique worth further study.

In this paper, the use of double exponential smoothing technique is studied for prediction of the number of software failures that will occur for a continued period of testing. Actual data is used as a numerical example and a comparison to other prediction approach is carried out. From the results shown, the double exponential

smoothing technique has the ability of providing more accurate prediction compared with traditional parametric models. It is very easy to use and it requires very limited amount of data storage and computational effort. It can be updated instantly without much calculation. Hence, it is a tool that should be more commonly used in practice.

## 2. The Exponential Smoothing Technique

Exponential smoothing is a powerful forecasting technique which is characterized by its simplicity and nonparametric properties, see e.g., Gilchrist (1976) and Montgomery et al. (1990). A forecasting procedure, as derived from the model, is a procedure whose input is historical data and whose output is a predicted value. The predicted value is generated by the following equation:

$$\text{New Prediction} = \alpha (\text{current observation of Failures}) + (1 - \alpha)(\text{last Prediction})$$

Denote by  $N_i$  the observation of the number of software failures during the latest time period and  $F_i$  the last prediction of failures, we have that the predicted number of the failures for the next period is given by

$$F_{i+1} = \alpha N_i + (1 - \alpha)F_i,$$

where  $0 < \alpha \leq 1$  is the smoothing constant which determines the relative weight placed on the current observation of failures.

It has been noted that the simple exponential smoothing technique always lags behind the trend when it is present. The trend is the most important issue in software reliability prediction. In order to solve this problem, a type of double exponential smoothing designed to track time series with linear trend can be used, see e.g., Gilchrist (1976). The method requires two smoothing constants,  $\alpha$  and  $\beta$ . Two smoothing equations are used:

$$S_i = \alpha N_i + (1 - \alpha)(S_{i-1} + G_{i-1}),$$

$$G_i = \beta (S_i - S_{i-1}) + (1 - \beta)G_{i-1}.$$

where  $S_i$  is interpreted as the value of the intercept at time  $i$  and  $G_i$  as the value of the slope at time  $i$ . The first equation is very similar to that used for simple exponential smoothing and the second equation is for the trend.

When the most current failure numbers  $N_i$  becomes available, it is averaged with the prior prediction of the current failure numbers, which is the previous intercept  $S_{i-1}$  plus the previous slope  $G_{i-1}$ . The new estimate of the intercept  $S_i$  causes us to revise our estimate of the slope to  $S_i - S_{i-1}$ . This value is then averaged with the previous estimate of the slope  $G_{i-1}$ .

The predicted number of failures in the next interval based on the double exponential smoothing technique is then given as

$$F_{i+1} = S_i + G_i$$

The  $k$ -step-ahead prediction  $F_{i+k}$  can also be calculated as:

$$F_{i+k} = S_i + k G_i.$$

The exponential smoothing technique has not been commonly used in software failure prediction. The main reason for this seems to be the fact that simple exponential smoothing is not able to capture the trend of the process. However, by employing the double exponential smoothing technique, the simplicity remains and the problem of capturing the trend is solved.

It should be noted that the selection of the smoothing constants is a statistical issue. However, in practice, they depend on the weight on the latest information that we would like to give. As the from our experience, any reasonable value of the smoothing constants will give an accurate result for the predicted values. Unlike the traditional software reliability growth models which require accurate estimates of the model parameters, the exponential smoothing technique is much more robust.

On the other hand, the exponential smoothing technique requires an initial value to get started. This can be done by using the first two observations to determine an initial values. In most of practical situations, reasonable prediction can only be obtained after a certain amount of testing. This is especially the case when the existing software reliability models are used because little initial failure information usually implies inaccurate estimates of the parameters which makes the prediction inaccurate. Hence, this problem is not specifically for the method of double exponential smoothing. In fact, exponential smoothing techniques do not depend very much on early data, as their weight in the prediction is decreasing exponentially.

Although there are some discussion of time series methods, see e.g., Singpurwalla and Soyer (1985), Soyer (1986) and Goel (1996), most of them a related to the use of moving averages type of methods. The double exponential smoothing stressed the analysis of trend and it has the advantage that it does not require a keeping of a large number of observations. What is needed is the latest value of  $F_i$  and  $G_i$ . They can be updated easily whenever a new observation is available and suitable for automatic data collection system, see Murthy and Gent (1995). This paper is hence focused on the study of double exponential smoothing technique and its comparison with other traditional software reliability growth models.

### **3. Application to Software Reliability Data**

The application of the double exponential smoothing technique in predicting software failure failures is a straightforward issue. A numerical example based on an actual set of data is given in this section. The data displayed in Table 1 in this study is from a large communication software system. Table 2 shows, for the last 20 months, the one-step double exponential smoothing prediction and the error in the prediction. The values of smoothing constants used are and the method is initiated using the first two month of the data.

Table 1. Failure data of a large communication software system  
(read from left to right and up and down).

2	11	18	10	12	4	28	6	7	6
17	31	8	7	10	2	2	0	3	2
1	1	4	3	39	20	12	13	26	33
8	8	11	14	8	7	8	0	2	3
2	5	3	4	4	1	3	1	0	2

The plot of system failures together with the double exponential smoothing prediction is given in Figure 1. The smoothing constants used are 0.6 and 0.1, respectively. In this case, a large value for  $\alpha$  is used because it can be suspected that the number of failures in a month might depend more on the recent values than much earlier ones. We see that the double exponential smoothing technique can predict very well the trend in the data change.

Table 2. The results of the prediction and the prediction error using the double exponential smoothing technique

Month	# System failures	Prediction by DES	Error of DES
31	8	8.0	0.0
32	8	8.1	0.1
33	11	8.1	-2.9
34	14	10.1	-3.9
35	8	12.9	4.9
36	7	10.2	3.2
37	8	8.3	0.3
38	0	8.1	8.1
39	2	2.7	0.7
40	3	1.8	-1.2
41	2	2.0	0.0
42	5	1.5	-3.5
43	3	3.4	0.4
44	4	2.9	-1.1
45	4	3.3	-0.7
46	1	3.6	2.6
47	3	1.7	-1.3
48	1	2.2	1.2
49	0	1.2	1.2
50	2	0.1	-1.9

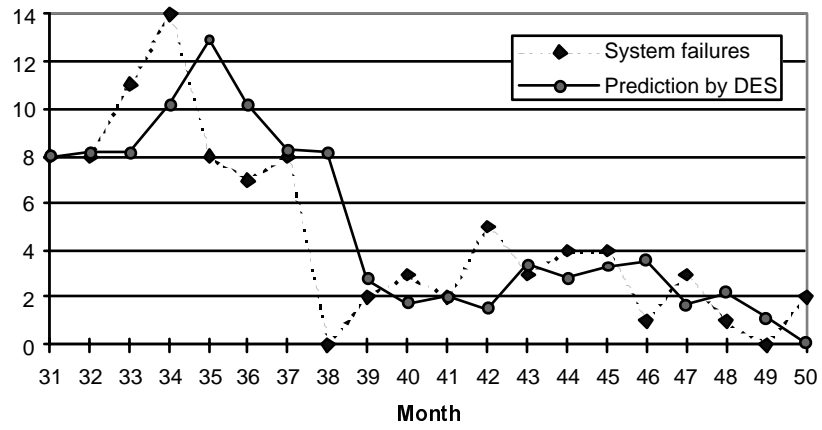


Figure 1. The double exponential technique applied to system failure prediction.



## **5. Comparison with some other software reliability growth models**

Many software reliability models have been developed and tested in practice. However, it is commonly agreed that there is no single model which is superior to all others all the time. The objective of the research of the time series models is to develop an analytical framework for forecasting future software failure times. The developed models are studied to provide a characterization of the underlying software failure phenomenon.

In this section, we compare our double exponential smoothing approach with some other commonly used software reliability growth models when applied to the same data set. The models used are the Goel-Okumoto model and the s-shaped NHPP model. These models all belong to the class of non-homogeneous Poisson process (NHPP) models and they are widely used by software reliability practitioners. They are also among the earliest models proposed, see Xie (1993). Compared with other early models, this type of models are known for their simplicity.

NHPP models are all characterized by a mean value function  $m(t)$  which is the expected cumulative number of failures during  $(0, t]$ . Given the set of data, the parameters in the model can be estimated using the maximum likelihood method. When the functional relationship is determined, the number of failures in the next period of time can be calculated as

$$m(i+1) - m(i)$$

This is the approach adopted in our comparison. The data set is as in Table 1 and the maximum likelihood estimates are obtained using the software accompanied with the book by Lyu (1996). The one-step ahead prediction using the three NHPP models is compared with our double exponential smoothing approach for the last 12 months and the results are shown in Table 3.

Table 3. Comparison of the double exponential smoothing with other software reliability growth models

<i>Month</i>	<i>System Failures</i>	<i>Prediction by DES</i>	<i>Prediction by GO Model</i>	<i>Prediction by S-Shaped Model</i>
39	2	2.0	10.5	7.5
40	3	2.2	10.2	6.8
41	2	2.6	9.5	6.2
42	5	2.5	8.8	5.6
43	3	3.7	8.4	5.3
44	4	3.6	7.9	5.0
45	4	4.0	7.5	4.7
46	1	4.2	7.2	4.4
47	3	3.0	6.6	4.0
48	1	3.1	6.3	3.7
49	0	2.2	5.9	3.4
50	2	1.2	5.4	3.1
Sum of Absolute Errors		13.3	64.2	29.7
Sum of Square Errors		28.03	377.26	102.09

The plot of the predicted number of failures using the two models and the double exponential smoothing technique is displayed in Figure 2 and the error plot is shown in Figure 3. From the comparison, it is clear that for this set of failure data, the double exponential smoothing technique provides a much more accurate prediction.

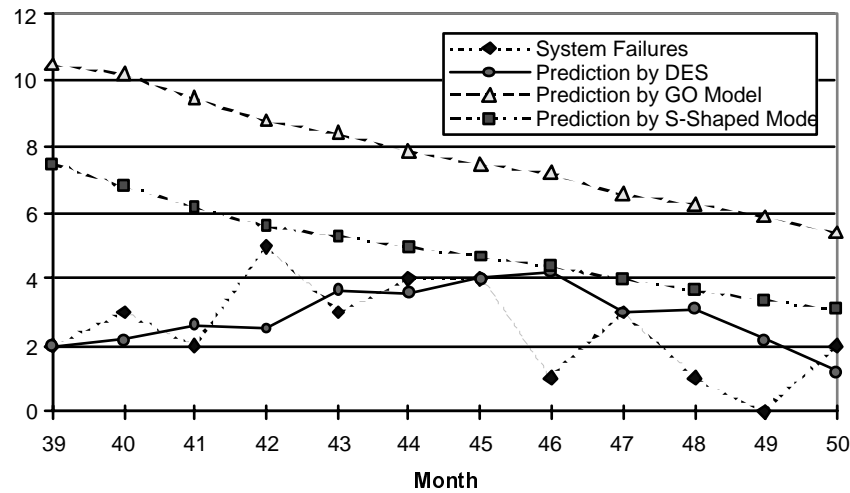


Figure 2. Comparison of the predicted values with the actual observation.

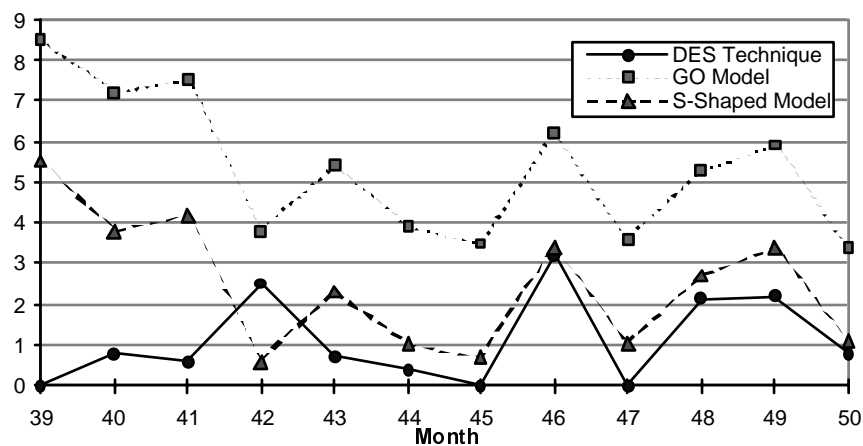


Figure 3. Error term comparison of the methods.

The above comparison uses the system failure data presented in Xie and Wohlin (1995). Although it is common that the system failure data is used in system reliability analysis, additional background for this data set is given in the original paper. From that, we know that the system is composed of two subsystems with one subsystem made available later than the other one. The later part of the failures are mainly from the second subsystem. This change can be observed in the data in Table 1 as there is a sudden increase in the number of failures around month 25.

It is argued in Xie and Wohlin (1995) that for this type of data, traditional models are not good and they suggested an additive model for the analysis. Most of the software reliability growth models, except the s-shaped NHPP model, cannot capture the increase in the failure intensity which is common for a short period of time. This probably explains why the s-shaped NHPP model performs better than the Goel-Okumoto model for this set of data. However, the additive model which is more reasonable than a single model makes use of two models, one for each system, and they are usually more complicated. In this case, we have improved the predictive ability significantly by simplifying the procedure as the same time.

One the other hand, the double exponential smoothing technique can be used even when the traditional software reliability models provides good results. We use the failure data from subsystem 2 as an example here. For this subsystem, the number of system failures for the 28 months the subsystem is available is displayed in Table 4. The plot of the predicted number of failures are compared for the three methods in Figure 4. Figure 5 shows the absolute error of the prediction.

Table 4. Number of subsystem 2 failures for each month since it is available.

3	3	38	19	12	13	26
32	8	8	11	14	7	7
7	0	2	3	2	5	2
3	4	1	2	1	0	1

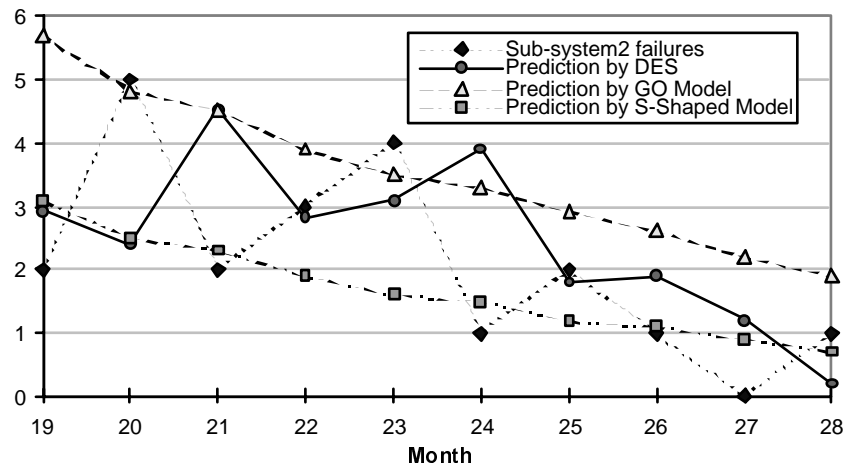


Figure 4. Comparison of the predicted values with the actual observation for a subsystem failure data.

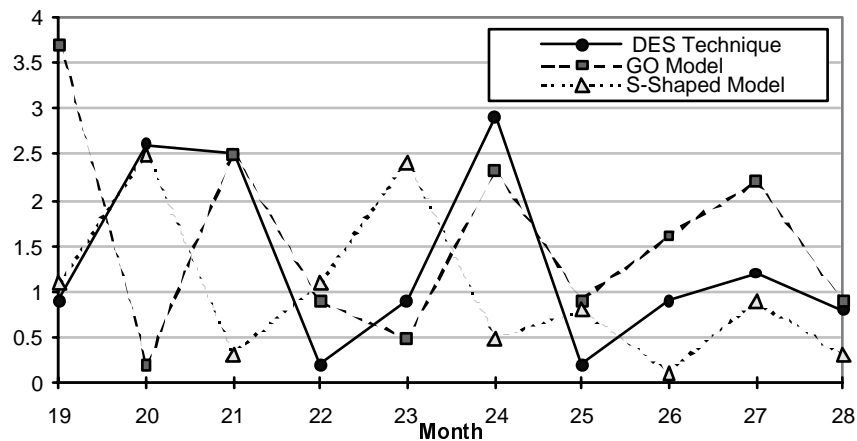


Figure 5. Error term comparison of the methods for a subsystem failure data.

In this case, the sums of the absolute errors are 13.1., 15.7 and 10.0 for the method of double exponential smoothing, the Goel-Okumoto model and the s-shaped model, respectively. The corresponding sum of square errors are 26.01, 35.35 and 16.32, respectively. The s-shaped model is a better one, but all three models have a similar performance in this case. This can be explained that there is a clear increasing failure intensity at the beginning, a case for which the s-shaped model is able to capture. On the other hand, the double exponential smoothing constant are

selected arbitrary to illustrate the simplicity of the approach and they may not be the optimum ones.

It can be noted that adding new subsystem to an existing system is a common situation for repairable systems. It can also be the case for combined software and hardware systems, see e.g., Keene and Chris (1992) for a discussion of the importance of this type of systems. The double exponential smoothing technique can be applied when analysing this type of systems.

### **5. Reliability related prediction based on the double exponential smoothing technique**

As discussed before, the double exponential smoothing technique is useful for predicting of the number of failures in the next time period. However, sometimes it is more useful to provide an estimate of failure intensity. It can be calculated as the reciprocal of the number of the failures. That is, an failure intensity estimate can be given as

$$\lambda_0 = \lambda(i) = 1 / F_{i+1}$$

This is also the estimate of the field failure intensity when the software is released without further testing.

On the other hand, if the testing is to be continued for an additional  $k$  periods of time, we can use the  $k$ -step ahead prediction of the number of failures to estimate the failure intensity  $\lambda_k$  expected to be achieved at that time. That is

$$\lambda_k = \lambda(i+k) = 1 / F_{i+1+k} = 1 / (S_i + (k+1)G_i)$$

where  $F_i$  and  $G_i$  are the current values.

If the software is released without additional testing, we can determine the field reliability  $R(T)$  which is the probability that there is no failure  $T$  time units after release. This is given as

$$R(T) = \exp\{-\lambda_0 T\}$$

where  $\lambda_0$  is given as before.

A sequential decision procedure on whether to terminate the testing or continue with it can also be based on this estimate. For example, if, after a certain amount of testing, the reliability estimate given above is not as high as required, we can determine additional amount of testing needed to reach a prescribed reliability level  $R(T)$ .

This can be done using the  $k$ -step ahead prediction of the failure intensity given previously. We have that

$$R(T) = \exp\{-\lambda_k T\} = \exp\{-T/(S_i + (1+k)G_i)\}$$

so that the number of time periods needed for further testing,  $k$ , can be determined based on this reliability requirement as  $F_i$  and  $G_i$  are known from the double exponential smoothing prediction procedure.

## 6. Conclusions

In this paper, we have studied the use of the double exponential smoothing technique in prediction of software failures during testing. This method is a reasonable approach as in software reliability analysis, it is important to be able to predict software failures and track its trend. Traditional models rely heavily on the fitting of past data which may not be representative because of the change of testing strategy and other testing conditions.

From the results of the comparison to other prediction models, we notice that the double exponential smoothing technique is a better method for predicting software failures in the testing process. It is very simple and the application is straightforward without complicated numerical computation, and yet the results are very accurate compared with other reliability growth models.

## References

- Gilchrist, W. (1976). *Statistical Forecasting*. Wiley, New York.
- Goel, A.L. (1996). Relating operational software reliability and workload: results from an experimental study. *Proc. Annual Reliability and Maintainability Symposium*, pp.167-172.
- Goel, A.L. and Okumoto, K. (1979). Time-dependent error-detection rate model for software reliability and other performance measures. *IEEE Transactions on Reliability*, R-28, 206-211.
- Keene, S. and Chris, L. (1992). Combining software and hardware aspects of reliability, *Quality and Reliability Engineering International*, 8 (4), 419-426.
- Keiller, P.A. and Miller, D.R. (1991). On the use and the performance of software reliability growth. *Reliability Engineering and System Safety*, 32, 95-117.
- Littlewood, B. (1980). Theory of software reliability: how good are they and how they can be improved? *IEEE Transactions Software Engineering*, SE-6, 489-500.
- Littlewood, B. (1987). How good are software reliability predictions. In *Software Reliability: Achievement and Assessment*. Ed. B. Littlewood, Blackwell, Oxford, pp.172-191.
- Lyu, M. (1996). *Handbook of Software Reliability Engineering*. McGraw-Hill, New York.
- Montgomery, D.C.; Johson, L.A. and Gardiner, J.S. (1990). *Forecasting & Time Series Analysis*. McGraw-Hill, New York.
- Murthy, B. and Gent, T. (1995). Measuring system and software reliability using automated data collection process. *Quality and Reliability Engineering International*, 11, 341-353.
- Musa, J.D. Iannino, A. and Okumoto, K. (1987). *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, New York.
- Singpurwalla, N.D. and Soyer, R. (1985). Assessing (software) reliability growth using a random coefficient autoregressive process and its ramifications. *IEEE Transactions on Software Engineering*, SE-11 (12), 1456-1464.



Soyer, R. (1986). Applications of time series models to software reliability analysis. In *Software Reliability: State of the Art Report*, Eds. A. Bendell and P. Mellor, Pengamon Infotech Ltd., pp.197-208.

Xie, M. (1991). *Software Reliability Modelling*. World Scientific Publisher, Singapore.

Xie, M. (1993). Software reliability models - a selected annotated bibliography, *Journal of Software Testing, Verification and Reliability*, Vol.3, 3-28 (1993).

Xie, M. and Wohlin, C. (1995). An Additive Reliability Model for the Analysis of Modular Software Failure Data. *Proc. Sixth Int. Symp. on Software Reliability Engineering*, pp.188-194.

Yamada, S.; Ohba, M. and Osaki, S. (1983). S-shaped software reliability growth models and their applications. *IEEE Transactions on Reliability*, R-33, 289-292.

Dr. Finn Jensen  
Chief Editor, QREI  
Pile Alle 11  
DK-2840 Holte  
Denmark

Singapore, 12 June 1996

Dear Dr.Jensen:

**RE: A paper submitted to QREI**

Please find enclosed a paper "A STUDY OF EXPONENTIAL SMOOTHING TECHNIQUE IN SOFTWARE RELIABILITY GROWTH PREDICTION" which we submit to your journal for possible publication.

In this paper, we have studied the use of an exponential smoothing technique in predicting software failures. Software reliability growth prediction has been a research area of mine for more than a decade and I am surprised to see that such a good method is not of a wide use. In this paper, we show that it is an accurate and simple prediction method. Comparison with other commonly used models are shown to support this.

We hope that you and the referees will find this paper acceptable and we are looking forward to your reply at your earliest convenience.

Sincerely yours

M. Xie  
Dept of Industrial and Systems Engineering  
National University of Singapore  
Kent Ridge, Singapore 0511  
Fax: (65) 777 1434  
E-mail: ISEXIEM@NUS.SG