Parameter Estimation and Validation Testing Procedures for Software Reliability Growth Model

Manohar Singh[1], Dr. Vaibhav Bansal[2]

Associate Professor, Department of Computer Science, OPJS University, Churu, Rajasthan

Abstract: ***In this age of technology, computers are being used to monitor and control safety critical and civilian systems with a great demand for higher quality software products. So reliability of the Software is the primary concern for both software developer and software user. The Software Reliability Growth Model can be used to predict the number of failures that may be encountered during software testing process. The Software Reliability Growth Model (SRGM) is a mathematical model of how the software reliability improves as faults are detected and repaired. Several Software Reliability Growth Models have been proposed during the past three decades for assessing the reliability of software product. In this paper we provide procedures to estimate the parameters of SRGMs and a critical analysis of prediction of Goodness of Fit using some existing Software Reliability Growth Models.***

Keywords: SRGM, Parameters of SRGM, Goodness of Fit, Least Square Estimation, Maximum Likelihood Estimator, SSE, MSE

1. Introduction

The utility of a SRGM is related to its stability and predictive ability. Stability means that model parameters should not significantly change as new data is added. Predictive ability means that the number of remaining defects predicted by the model should be close to the number found in field use.

Software Reliability is the most important aspect in measuring the quality of software. Software Reliability Growth Model is used to estimate the reliability change in software products and make the reliability growth prediction for making testing resource allocation decisions.

Software Reliability is the probability of failure of free software operation for a specified period of time in a specified environment. [1]. SRGM is one of the fundamental models to access Software Reliability Quantitatively [2]. All the reliability growth models are based upon the hypothesis that the reliability of a program is a function of the number of faults that it contains. The Software Reliability Growth Model assures good performance certainly in terms of goodness-of-fit, certainty. In order to estimate as well as to predict the reliability of software systems, failure data need to be properly measured using various metrics during software development and operational phases. [3] SRGM is a mathematical model of how the Software Reliability improves as faults are detected and corrected [4]. A number of NHPP based SRGMs have been reviewed and compared on their fit and predictive power by Pham [5]. Wood [6] studied the comparison of Software Reliability Growth Models on defect inflow data and found it correlated with past released defects. Although a number of studies have compared and evaluated SRGM within different context. We are not proficient to make a consensus on how to choose SRGMs for specified purpose and which models are best for given process characteristics.

In the remaining paper there are three more Sections. In the Section-2, we discuss the parameter estimation methods and various comparison criteria for prediction of SRGMs. Numerical implementation of parameter estimation of some existing SRGM with Data Set of Time-Domain data for a real time control system and the analysis of the results regarding prediction of SRGMs is presented in Section-3. Finally the conclusions are presented in Section-4.

2. Parameter Estimation of SRGM Models

Software Reliability Growth Models are an abstract form which contains several parameters of unknown value. They have to be estimated based on the basis of the input failure data. So that the resulting function describes the data as closely as possible. SRGMs are a statistical interpolation to detect defected data by mathematical function. These are used to predict future failure rates in software. The achievement of mathematical modelling approach to reliability evaluation depends a lot upon quality of failure data collected. The parameters of the SRGM are calculated based upon these data. There are two common types of the failure data: Time-Domain data and Interval-Domain data. Time-Domain data are characterised by recording the individual time at which the failure occurred, i.e. the time between two consecutive failures is recorded. Interval domain data are characterised by counting the number of failures during a fixed period. Some existing software reliability models can handle both type of data but Time-Domain data provides better accuracy of parameters estimation with current existing Software Reliability Models [5]. IEEE standard 1633: recommended practice on Software Reliability [7] provides a 13 steps procedure to access and predict the Software Reliability and commonly used methods for parameters estimation of SRGMs.

2.1 Methods to estimate the parameters of SRGMs

Parameter Estimation is of primary importance in Software Reliability predication. Once the mean value function $m(t)$ of analytical model is known, the parameter in the solution is required to be determined. During the testing and early operational phases of Software Development Life Cycle (SDLC), failure events are encountered. They are recorded and underlying faults that caused them are removed, which results in process called Reliability Growth. The basic idea behind the SRGM is simple; if the history of fault detection and removal follows a certain recognizable pattern, it is possible to describe the mathematical form of the pattern.

The function that represent this pattern is called mean value function $m(t)$, which is cumulative number of faults describe in a given time t. If we are able to fit this function to the existing historical fault detective data, we can predict the future failure behaviour of software. The mean value function is often transferred to failure intensity (rate) function $\lambda(t)$ by formula $\lambda(t) = \frac{dm(t)}{dt}$. The parameters of SRGMs are estimated by one of the following methods:

- Least Square Estimation (LSE)
- Maximum Likelihood Estimation (MLE)

The MLE is more suitable for the large sample of data and the LSE for small to medium size sample.

2.1.1 Least Squares Estimation (LSE)

In Least Square Method (LSM), the sum of squares of the difference between observed response and value predicted by the model is minimized. If the expected value of the response variable is given by $m(t)$, then the least square estimators of the parameters of the model may be obtained from n pairs of sample values $(t_1, y_1), (t_2, y_2) \ldots, (t_n, y_n)$, by minimizing S given by :

$$S = \sum_{k=1}^{n} [y_k - m(t_k)]^2 \tag{1}$$

where t_k and y_k are the observed values of explanatory and dependent variables respectively. For small and medium size samples least squares estimation is preferred [8]. For estimation of the parameters of the analytical models, Method of Least Square (Non Linear Regression) has been used. Non Linear Regression is a technique to find a nonlinear model of the relationship between the dependent variable and a set of independent variables. Unlike conventional linear regression, which is restricted to estimate linear models, nonlinear regression can estimate models with arbitrary relationships between independent and dependent variables.

2.1.2 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) method is one of the most useful techniques for estimation of parameters of SRGM based upon NHPP [8]. We briefly discuss below the MLE procedure for two types of software failure data as discussed above. For the first type of data, the estimation is to be performed at a specified time t_k, not necessarily corresponding to a failure, and with total of m_k failures being experienced at time t_1, t_2, ..., t_{mk}. Then the likelihood function for the NHPP is:

$$L = \left[\prod_{i=1}^{k} \lambda(t_i)\right] e^{-\int_0^{tk} \lambda(x)dx} \tag{2}$$

The MLE of the Parameters can be obtained by maximizing Likelihood function or its log likelihood function (log L).
If the software failure data is grouped into k points (t_i, y_i) ; $i = 1, 2, \ldots\ldots, k$, where y_i is the cumulative number of failure reports at time t_i. Then the Likelihood functions L is given as follows:

$$L = \prod_{i=1}^{k} \frac{[m(t_i) - m(t_{i-1})]^{y_i - y_{i-1}}}{(y_i - y_{i-1})!} e^{-\{m(t_i) - m(t_{i-1})\}} \tag{3}$$

Taking natural logarithm of (3) we get the log likelihood function as:

$$LogL = \sum_{i-1}^{k} (y_i - y_{i-1}) ln[m(t_i) - m(t_{i-1})] - m(t_k) - \sum_{i-1}^{k} ln[(y_i - y_{i-1})!] \tag{4}$$

The MLE of the parameters of SRGM can be obtained by maximizing (4) with respect to the model parameters. Estimation of parameters using MLE requires solving set of simultaneous equations to maximize the likelihood defect data coming from given function to find parameters.

Wood [6] applied both MLE and LSE and found LSE to be more stable and better correlated to field data although MLE results were more reasonable. It can be safely assumed that statistically MLE is much better parameter predictions method than LSE as LSE is much easier and provide consistent results in wider data sets than preferred methods [9] [10] [5]

2.2 Comparison Criteria for SRGM

Once the parameters of mean value function $m(t)$ are estimated by any of the methods mentioned above, the performance of SRGM are judged by their ability to fit the past software fault data (goodness of fit) and to predict satisfactorily the future behaviour of the software fault removal process (predictive validity) [11] [12]. Musa et al. [8] have suggested the following attributes for choosing an SRGM.

a) **Capability**: The model should possess the ability to estimate with satisfactory accuracy, the metrics needed by the software managers.
b) **Quality of assumptions:** The assumptions should be plausible and must depict the testing environment.
c) **Applicability:** A model can be adjudged as the better one if it can be applied across software products of different sizes, structures, platforms and functionalities.
d) **Simplicity**: The data required for an ideal SRGM should be simple and inexpensive to collect. The parameter estimation should not be too complex and should be easy to understand and apply even for persons without extensive mathematical background. Other than the above qualitative aspects the goodness of fit and predictive validity criteria help to compare SRGMs.

2.3 Goodness of Fit Criteria

The commonly used criteria for model comparison of Goodness of Fit and the predictive power are given in the Table 1.

2.4 Predictive Validity Criteria

Predictive validity is the ability of the model to determine the future failure behaviour from present and past failure behaviour. This criterion was proposed by Musa et al. [8]. Suppose t_k be the time, x_k is number of faults detected during the interval $(0, t_k]$, and $m(t_k)$ is the estimated value of the mean value function $m_r(t)$ at t_k, which is determined using the actually observed data up to an arbitrary time t_e ($0 < t_e \leq t_k$), in which $\frac{t_e}{t_k}$ denotes the testing progress ratio. In other words, the number of failures by t_k can be predicted by the SRGM and then compared with the actually observed number x_k. The difference between the predicted value $m(t_k)$ and the reported value x_k measures the prediction fault. The ratio $\left\{\frac{m(t_k)-x_k}{x_k}\right\}$ is called Relative Prediction Error (RPE). If the RPE value is negative/positive the SRGM is said to underestimate/overestimate the future failure phenomenon. A value of RPE is closer to zero indicates more accurate prediction, thus more confidence in the model and better predictive validity [8] [12]. The various metrics such as R^2, MSE, AIC, SSE, Variation, Bias, RMSPE and PRR shown in Table 1 can be used for predicting the validation of SRGM.

Table 1: Goodness of Fitness Metrics

Metrics	*Formula*	*Remarks*
Sum of Squared Error (SSE)	$SSE = \sum_{i=1}^{n}[m(t_i) - \hat{m}(t_i)]^2$	Lower the value of SSE, is the better goodness of fit [12]
Mean Square Fitting Error (MSE)	$MSE = \frac{\sum_{i=1}^{n}[m(t_i) - \hat{m}(t_i)]^2}{N}$	Lower the value of MSE, is the better goodness of fit [12]
Akaike Information Criterion (AIC)	$AIC = -2$ (value of the maximum log likelihood function) +2 N	Lower values of AIC indicate the preferred model [12].
Coefficient of Multiple Determination (R^2)	$R^2 = 1 - \frac{residual\ SS}{corrected\ SS}$	Model provides a better Goodness of Fit for R^2 is close to1. [12].
Predictive-Ratio Risk (PRR)	$PRR = \sum_{i=1}^{n}\left[\frac{m(t_i) - \hat{m}(t_i)}{\hat{m}(t_i)}\right]^2$	Lower the value of PRR better is the goodness of fit [13].
Prediction Error (PE)	$PE_i = OE_i - PE_i$	Lower the value of Prediction Error better is the goodness of fit [13].
BIAS	$BIAS = \frac{\sum_{i=1}^{n} PE_i}{n}$	Lower the value of BIAS better is the goodness of fit [13]
Variation	$Variation = \sqrt{\frac{\sum(PE - BIAS)^2}{N-1}}$	Lower the value of variation better is the goodness of fit [13]
Root Mean Square Prediction Error (RMSPE)	$RMSPE = \sqrt{(Bias^2 + Variation^2)}$	Lower the value of RMPSE better is the goodness of fit [13].

3. Numerical Implementation of Parameters Estimations

We now illustrate the procedure for estimation of unknown SRGM model parameters with practical considerations.

3.1 Data Set and Models

The data set used for this study is Time-Domain data for a real time control system provided by Ohba [14]. In data 15 faults has been reported with their time between the failures. In this study we use the following four early and widely used SRGMs as given in Table 2.

Table 2: SRGM Model used in this study

Model	Name of SRGM	Mean Value Function $m(t)$	Parameters to be Estimated
Model-1	Exponential Goel-Okumoto (G-O) [15]	$m(t) = a(1 - e^{-bt})$	Two, (a, b)
Model-2	Inflection S-Shaped Model [16]	$m(t) = \frac{a(1 - e^{-bt})}{1 + \beta e^{-bt}}$	Three, (a, b, β)
Model-3	Yamada Imperfect Debugging Model [17]	$m(t) = a(1 - e^{-bt})\left(1 - \frac{\alpha}{b}\right) + \alpha at$	Three, (a, b, β)
Model-4	PNZ Model [18]	$m(t) = \frac{a(1 - e^{-bt})\left(1 - \frac{\alpha}{b}\right) + \alpha at}{1 + \beta e^{-bt}}$	Four, (a, b, α, β)

3.2 Estimation of Parameters and Analysis

In this study the parameters of software reliability growth model mentioned in the Table 2 are estimated by using Least Square Estimation (Non Linear Regression). We use the statistical package IBM SPSS for estimation of parameters for goodness of fit and prediction of the models. The values of parameters estimated by using Least Square Estimation (LSE) of Non Linear Regression (NLR) and parameters values using Maximum Likelihood Estimation (MLE) obtained with same data and models by Pham-Zhang [14] are summarized in Table 3 for comparison.

The fitting of the models given in Table 2 using LSE and MLE estimators to the observed data is also presented in Figure 1, Figure 2, Figure 3 and Figure 4. It is observed from these figures the curves of the fitted models using LSE and MLE Estimators are almost consistent which show Goodness of Fit. It is also noted that for both curve and growth rate our estimates using LSE are much closer to the parameters estimates obtained using MLE.

Table 3: Estimated Parameters

Model	Mean Value Function $m(t)$	Estimated values of Parameters	
		LSE	MLE [14]
Model-1	$m(t) = a(1-e^{-bt})$	$\hat{a} = 18.539$ $\hat{b} = 0.005$	$\hat{a} = 19.54$ $\hat{b} = 0.0049$
Model-2	$m(t) = \frac{a(1-e^{-bt})}{1+\beta e^{-bt}}$	$\hat{a} = 28.306$ $\hat{b} = 0.000023$ $\hat{\beta} = -0.994$	$\hat{a} = 28.58$ $\hat{b} = 0.00013$ $\hat{\beta} = -0.965$
Model-3	$m(t) = a(1-e^{-bt})\left(1-\frac{\alpha}{b}\right) + \alpha a t$	$\hat{a} = 10.873$ $\hat{b} = 0.010$ $\hat{\alpha} = 0.002$	$\hat{a} = 11.40$ $\hat{b} = 0.0094$ $\hat{\alpha} = 0.001925$
Model-4	$m(t) = \frac{a(1-e^{-bt})\left(1-\frac{\alpha}{b}\right) + \alpha a t}{1+\beta e^{-bt}}$	$\hat{a} = 10.146$ $\hat{b} = 0.011$ $\hat{\alpha} = 0.002$ $\hat{\beta} = 0.084$	$\hat{a} = 12.1363$ $\hat{b} = 0.0144$ $\hat{\alpha} = 0.00146$ $\hat{\beta} = 0.9629$

3.3 Predictive validity and Accuracy

To predict the goodness of fit and validity of Model using LSE and MLE estimators, the various metrics are evaluated such as Sum of Squared Errors (SSE), Mean Squared Error (MSE), Predictive-Ratio Risk (PRR) and Root Mean Square Predictive Error (RMPSE) summarized in Table 4. We now compare the predictive accuracy of the value of a metrics obtain using MLE and NLR estimators as shown in Table 4. It is observed that the values of SSE, MSE, PRR and RMSPE obtained with MLE estimator are less than the values obtained with LSE estimator in Model-1, Model-2 and Model-3 which indicates that MLE is better estimator.

Table 4: Comparison of Goodness of Fit Metrics

Metrics →	SSE		MSE		PRR		RMSPE	
Model ↓	LSE	MLE	LSE	MLE	LSE	MLE	LSE	MLE
Model-1	2.07	0.53	0.14	0.04	0.09	0.04	0.52	0.21
Model-2	0.41	0.30	0.03	0.02	0.01	0.01	0.21	0.17
Model-3	0.44	0.10	0.03	0.01	0.01	0.01	0.21	0.08
Model-4	3.38	6.83	0.23	0.46	0.03	0.10	0.59	0.84

The coefficients of Multiple Determination (R^2) are calculated using the parameters estimated using LSE estimators. The Table 5 summarized the values of R^2 using LSE and growth rate of the models with LSE and MLE estimators to predict the validity of the models. The values of R^2 obtained for various models lie close between 0 and 1. The values are very close to 1 or equal to 1 indicate the validity of the models. It is also observed that growth rates in both the cases of various models are almost equal which indicates the Goodness of Fit of models

Table 5: R^2 and Growth Rate to predict validity and accuracy of the Model

Model	Model-1	Model-2	Model-3	Model-4
Value of R^2 using LSE	0.999	0.999	1.000	1.000
Growth Rate with LSE	0.005	0.000023	0.010	0.011
Growth Rate with MLE	0.0049	0.00013	0.0094	0.0144

3.4 Asymptote for Goodness of Fit of Models

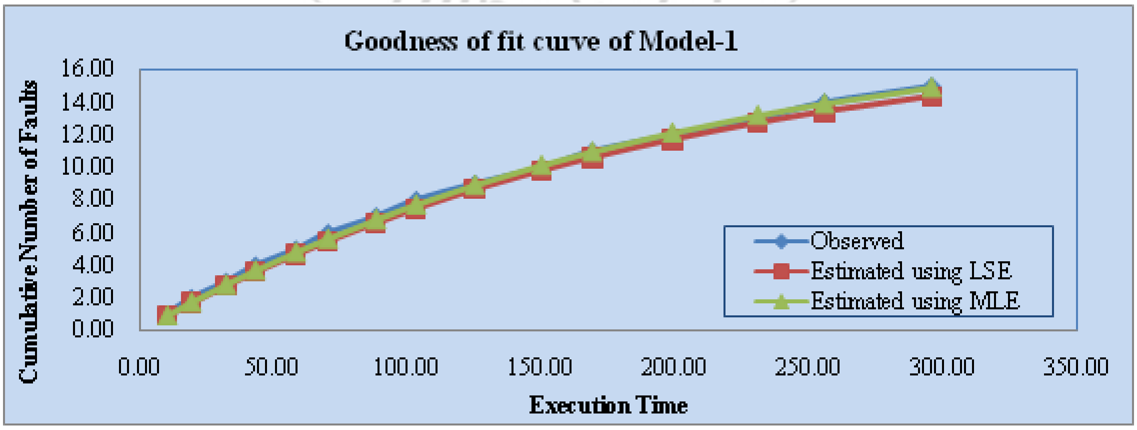

Figure 1: Model 1 - Goel-Okumoto (G-O)

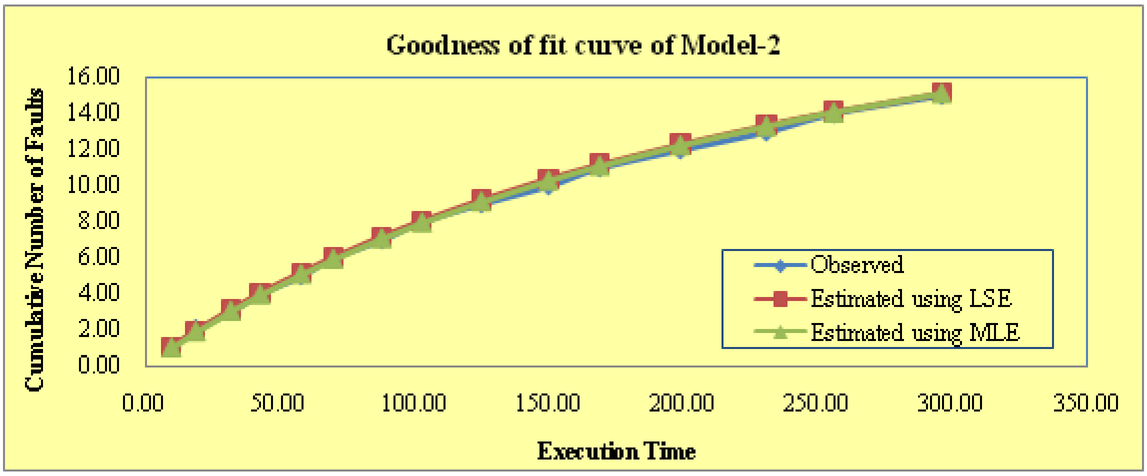

Figure 2: Model 2 - Inflection S-Shaped Model

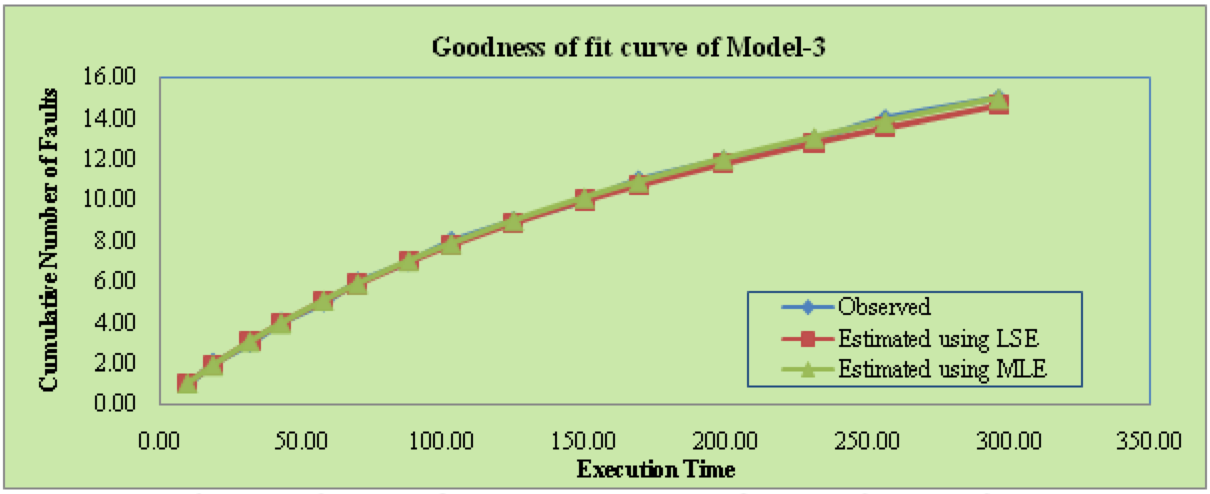

Figure 3: Model 3 - Yamada Imperfect Debugging Model

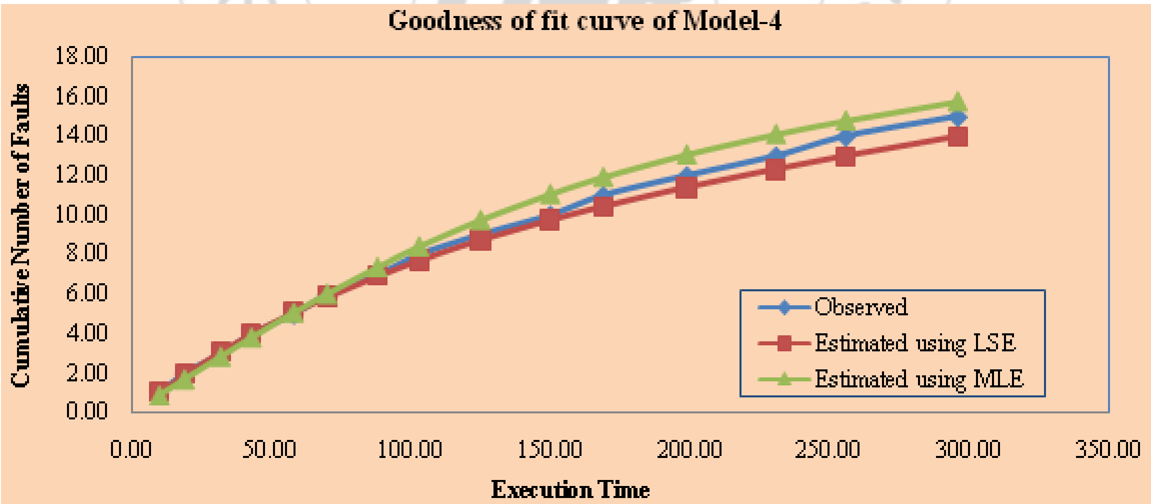

Figure 4: Model 4 - PNZ Model

4. Conclusion

In this paper we discussed most important methodologies, Least Square Estimation and Maximum Likelihood Estimation for estimating parameters of Software Reliability Growth Models and the various metrics used for comparison of Goodness of Fit and predictive validity. Here we used the failure data set from the Literature and existing SRGMs for implementation of LSE and MLE practically for comparison of Models. It was noted that MLE gives the better results as compared to LSE for Goodness of Fit and predictive validity of Models. It was observed that MLE is difficult to apply which limits its use in industry, especially due to lack of tools support whereas LSE is easy to use due to availability of compatible tools. It was concluded from the results presented here and the properties of LSE and MLE estimators suggested that Least Square Method for Non Linear Regression is a good estimator for fitting the data to observed failure data whereas Maximum Likelihood Estimator is better for prediction of Reliability Growth Models.

References

[1] Kaswan Singh, K., Choudhary, S. and Sharma, K., "A New Classification and Applicability of Software Reliability Models", International Journal of Advance Research in Computer Science and Management Studies., Vol. 2 Issue 7, July 2014.

[2] Mohd. R., Nazir, M., "Software Reliability Growth Models: overview and Applications", Journal of Emerging Trends in Computing and information Science, vol. 3 No. 9, Sep 2012

[3] Singh, M. "Software Reliability Testing Tools: An Overview and Comparison" IJECS Volume 05 Issue 11 Nov, Page No.18886-18891, 2016

[4] Quadri, S.M.K. and Ahmad, N., "Software Reliability Growth Modelling with new modified Weibull testing-effort and optimal release policy", International Journal of Computer Applications, Vol. 6, No. 12, 2010.

[5] Pham, H., "Software reliability and cost models: Perspectives, comparison and practice", European Journal of Operation Research, Vol. 149, No. 3, 475-489, 2003.

[6] Wood, A., "Predicting Software reliability," Computer, Vol. 29, No.11, pp. 69-77, 1996

[7] I.R. Society, IEEE recommended practice on Software Reliability, Std, 2008

[8] Musa, J.D., Iannino, A. and Komodo, K., Software Reliability: Measurement, Prediction and Application, McGraw-Hill, 1987.

[9] Ullah, N., Moriso, M. and Vetro, A., "A comparative analysis of software reliability growh models using defects data of closed and open source software," in Software Engineering Workshop (SEW), 2012 35th Annual IEEE, pp 187-192, 2012".

[10] Rana, R., Staron, M., Berger, C., Hanson, J., Nilsson, M. and Torner, F., "Evaluating the long-term predictive power of standard reliability growth models on automotive systems," Pasadena, CA, USA, 2013."

[11] Musa JD, Software Reliability Engineering, McGraw-Hill, 1999.

[12] Kapur PK, Garg RB and Kumar S, Contributions to Hardware and Software Reliability, World Scientific, Singapore, 1999.

[13] Pillai K and Nair VSS, "A Model for software Development effort and Cost Estimation", IEEE Transactions on Software Engineering, Vol. 23(8), pp. 485-497, 1997.

[14] H. Pham, "System Software Reliability, Implementation of PZ Coverage Model", Springer, ISBN-10: 1852339500, pg 231-232, 2006

[15] Goel AL and Okumoto K, "Time Dependent Error Detection Rate Model for Software Reliability and Other Performance Measure", IEEE Transactions on Reliability, R-28 (3), pp. 206-211, 1979.

[16] Ohba, "Software reliability-analysis models," IBMJ. Res. Develop.,vol. 28, pp. 428-443, July 1984.

[17] Yamada S, Ohba M and Osaki S, "S-shaped software reliability growth models and their applications", IEEE Trans. On Reliability, Vol. R-33, October, 1984.

[18] Pham, H., Zhang, X. "An NHPP Software reliability model and its comparison", International Journal of Reliability, Quality and Safety Engineering, Vol. 4, No. 3, pp 269-282, 1997.