ISSN:0975-9646

Joint effect of Learning and Testing Effort in SRGM with Fault Dependent Correction Delay

Shaik Mohammad Rafi[#1] , Dr.K Nageswara Rao[#2] , Dr. S . Pallam Setty[#3] , Dr. Shaheda Akthar[#4]

[#1] *Assoc. Professor, Dept. of C.S.E, S.M.C.E, Affiliated to JNTU Kakinada. A.P, INDIA.*
[#2] *H.O.D, Dept. of C.S.E, P.V.P.S.I.T, Vijayawada, Affiliated to JNTU Kakinada. A.P, INDIA.*
[#3] *Professor, Dept.of C.S. &T, Andhra University, A.P, India.*
[#4] *Principal, Sri Mittapalli Institute of Technology for women, Affiliated to JNTU Kakinada. A.P, INDIA.*

Abstract—Software Reliability growth models are helping the software society in predicting and analyzing the software product in terms of quality. In this context several software reliability growth models are proposed in the literature. Majority of models concentrated on fault detection process, ignoring the correction. Error detection, correction and dependency are the important phenomenon for the software reliability models. In this paper we proposed a new SRGM model based on correction lag and error dependency in SRGM with incorporating the testing effort and learning. All numerical calculations are carried out on real datasets and results are analyzed. By analyzing the results we can say that our proposed model fits well for the datasets.

Index Terms—Non homogeneous possoin process, software reliability growth model, correction lag, testing effort.

ACRONYMS

NHPP	: Non Homogeneous Poisson Process
SRGM	: Software Reliability Growth Model
MVF	: Mean Value Function
MLE	: Maximum Likelihood Estimation
TEF	: Testing Effort Function
LSE	: Least Square Estimation
MSE	: Mean Square fitting Error

NOTATIONS

m (t)	: Expected mean number of faults detected in time (0,t]
λ (t)	: Failure intensity for m(t)
n (t)	: Fault content function
m_1 (t)	: Cumulative number of leading faults detected up to t.
m_2 (t)	: Cumulative number of dependent faults isolated up to t.
W (t)	: Cumulative testing effort consumption at time t.
W*(t)	: W (t)-W (0)
w(t)	: Current testing effort
a	: Expected number of initial faults
r (t)	: Failure detection rate function
r	: Constant fault detection rate function.
a_1	: Total number of initial leading faults
a_2	: Total number of dependent faults.
p	: Probability factor
θ	: Detection rate of initial faults.
Ψ	: Inflection factor

I. Introduction

Software is one important vehicle which is driving the several electronic and commercial products. The development in products, software uses makes the increase in the need of the software. Testing is one important phase of software development life cycle. Testing is being done intended to find the errors in software products which are being added during the upper phases. Reliability is one important element for software testing where it defines quality. As the testing proceeds, errors are identified and removed from the software product to make the quality improvement in the software product. As the testing continues every bug is identified and fixed, it increases the reliability of the software product. Reliability is defined as a software product has to perform its functionality under given environmental conditions before it fails .reliability is one important measure of quality. Software
reliability models are helping the industries from decades in terms of economically and quantitatively. Software reliability growth models are described by mathematical models to show the real time testing environment. Basically software testing is complex in nature in order to understand to software testing environment completely
Mathematical models are constructed which re useful in describing the real time testing environment. Several papers are proposed in the literature. Generally software reliability growth models are classified as both analytical and data driven models [16]. Analytical models have three major categorized as non homogenous poison process models (NHPP), markovian models and Bayesian models. A non homogenous poison process model adopts a stochastic process to describe the software failure phenomenon. Software reliability growth models are based on assumptions depends upon bug fixing. The reliability growth models are classified as perfect and imperfect debugging models. Plenty of reliability are proposed to measure the software failure process successfully [2][4] [8][9][20]. Some of them based on non homogenous poison process model (NHPP) [2][4][18][19] are proposed to predict the future failures. The dependency among software failures can affect our software reliability [1][3][15][21][22]]. Software cost models and release policies are being analyzed by Xie et al, 2003, Yamada et al, 1985 , Pham et al, 2000 ,Huang et al, 2005) Non homogenous poison process models are influenced by a parameter m(t) which is a cumulative in number of failures exposed up to time t. Generally the reliability, models are

categorized into both concave and S shaped models. In 1979 Goel and Okumoto proposed exponential software reliability growth model based on the shape of cumulative number of failure parameter m(t).Yamada and Ohba have made [18][19]simple modifications to the existing reliability growth models to capture the testing environment and they proposed delayed s shaped model and inflection s shaped model, generally the S shaped models are derived by a factor that as the testing proceeds the testing people will learn about their environment, which effect the testing. Testing consumes many resources like time-test cases and man power. Many reliability growth models are proposed in literature which considers testing effort (Yamada has considered exponential and Raleigh testing effort functions into software reliability growth models[17]. Huang (2005) has proposed a reliability growth model by considering logistic testing effort function. Software reliability is dynamic in nature. Software may non monotonically increases or decreases due to dynamic nature of the software. Remaining section of the paper contains section II describes the review of error correction and detection models. Section III will explain testing effort dependent SRGM with correction lag and model derivation. Section IV describes the numerical calculations and goodness of fit techniques and the performance analysis of the models.

II Review Of Software Error Detection And Correction

Software reliability models are mathematical models which describes the realistic phenomenon of software testing during software development life cycle. These models are embedded with fault detection , correction and fault introduction. Many papers are proposed in the literature in this context. But several of them are just concentrated on fault detection process Marjory. Among several papers they assume faults are corrected as soon as they are detected. But it is not always true. Software product is complex in nature. So correcting the detected faults is quite cumbersome task for the correct correction people. So fault correction is time lag phenomenon. (Huang et al, 2004, Yanjun shu et al , 2009).

The number of remaining uncorrected faults is the difference between number of detected faults and corrected faults (Yanjun Shu et al, 2009). Xie et at,2007 studied and analyzed the number of corrected and detected faults based on the medium based software. Schneidewnd et al,2003) first person proposed a fault correction as constant delay in the software reliability model. Musa et al,1987 made an analysis on the dataset consisting of correction and detection fault of real time data control T1 project.

III Software Reliability Growth Model With Testing Effort And Correction Delay.

Many NHPP software reliability growth models are proposed to access the software reliability. Software reliability measures the how long a software can give correct service before it deviates from required service in a given conditional environment. Before software released into market an extensive test is conducted. Software with more errors when released into the market incurs high failure costs [Pham et al, 2000]. For that more sophisticated testing is needed to track the errors. During the software development many resources are consumed like manpower, test cases. TEF describes test expenditure in testing process. The TEF, which gives the effort, required in testing and CPU time the software for better error tracking.

A) SRGM with Testing effort functions

The following assumptions are made for software reliability growth modeling (Yamada et al, 1993,, Huang and Kuo et al, 2002, Huang et al, 2007)

(i) The fault removal process follows the Non-Homogeneous Poisson process (NHPP)

(ii) The software system is subjected to failure at random time caused by faults remaining in the system.

(iii) The mean time number of faults detected in the time interval (t, t+Δt) by the current test effort is proportional for the mean number of remaining faults in the system.

(iv) The proportionality is constant over the time.

(v) Each time a failure occurs, the fault that caused it is immediately removed and no new faults are introduced.

We can describe the mathematical expression SRGM with a testing-effort based on following

$$\frac{dm(t)}{dt}\times\frac{1}{w(t)} = r\times(a-m(t)) \qquad (1)$$

Solving the above equation based on condition m(0)=0; we get

$$m(t) = a\times(1-\exp(-r\times W^*(t))) \qquad (2)$$

And intensity function

$$\lambda(t) = a\times r\times w(t)\times\exp(-r\times W^*(t)) \qquad (3)$$

From the above (v) condition states that faults detected immediately removed but due to environmental conditions and complexity of the software faults cannot be removed as quickly. Every fault will take some time to remove it so it impacts a delay in the assumed model. Let us consider the delay factor function is given by [3][15][22]

$\varphi(t)$.

From the above we can modified the (v) assumption and new modified MVF is

$$\begin{aligned} m(t) &= m(t-\varphi(t)) = \\ &a\times(1-\exp(-r\times t)\times\exp(r\times\varphi(t))) \end{aligned} \qquad (4)$$

Now comparing the eq (2) and eq(4) we get

$$\begin{aligned} t-\varphi(t) &= W^*(t)\, and \\ t &= \varphi(t)+W^*(t) \end{aligned} \qquad (5)$$

Now new modified SRGM with testing effort is given by

$$m(t) = a\times(1-\exp(-r^*[W^*(t)+\varphi(t)])) \qquad (6)$$

Theorem 1:
(a) Now new intensity function is given by

$$\lambda(t) = \frac{dm(t)}{dt}$$
$$= a \times r \times \exp(-r \times [W^*(t) + \varphi(t)]) \times (w(t) + \frac{d\varphi(t)}{dt}) \quad (7)$$
$$a > 0, r > 0.$$

(b) $\frac{d\varphi(t)}{dt} < 1$ (8)

(c) the reliability of above SRGM with TEF and delay factor is given by

$$R(\Delta t / t) = \exp(a(\exp(-rW^*(t+\Delta t))\exp(-r\varphi(\Delta t + t)) - \exp(-rW^*(t))\exp(-r\varphi(t)))) \quad (9)$$

Model 1: NHPP SRGM with testing effort :
This is very popular model many authors have used it.
Now let us consider equation (4) and take initial condition as

$\varphi(t) = 0 \, and \, \frac{d\varphi(t)}{dt} < 1$ then
(10)

We obtain the equation as [31]

$$m(t) = a \times (1 - \exp(-r \times W^*(t)))$$
(11)

Model 2 : Delayed S shaped model with Testing effort
Popular S shaped model is proposed by Yamada which takes the testers capability in to consideration during testing[18][19]. It states that when the testing begins testers are not familiar with environment as the testing continues testers gain the knowledge. In this model failure rate initially increases and later decays.

If $\varphi(t) = \frac{1}{r} \times -\ln(1 + r \times W^*(t))$ (9)

Then $\frac{d\varphi(t)}{dt} = -\frac{w(t)}{1 + r \times W^*(t)} < 1$ (12)

Because w(t) < W*(t). equation (9) satisfies the theorem 1(b)
Now from the eq.(4) we get[9]

$$m(t) = (1 - \exp(-rW^*(t))(1 + rW^*(t))) \quad (13)$$

The intensity for the above equation is given by
$\lambda(t) = a \times r^2 \times w(t) \times W^*(t) \times \exp(-r \times W^*(t))$ (13)

Model 3: Inflection S shaped model with testing effort
Inflection S shaped model proposed by Ohba et al 1984. it states that some faults are not detected before other faults are identified. Now from delay function

$$\varphi(t) = \frac{1}{r} \times -\ln\left[\frac{\psi + 1}{1 + \psi \times \exp(-r \times W^*(t))}\right] \quad (14)$$

And $\frac{\varphi(t)}{dt} = -\left(\frac{\psi \times w(t) \times \exp(-rW^*(t))}{1 + \psi \times \exp(-rW^*(t))}\right) < 1$ (15)

The MVF for above model [25]

$$m(t) = a \times \left(\frac{1 - \exp(-r \times W^*(t))}{1 + \psi \times \exp(-r \times W^*(t))}\right) \quad (16)$$

B. Considering Fault dependency and debugging time lag in testing effort dependent software reliability growth model [3]

The following are the assumptions for the NHPP model
1) Fault detection and correction model follow the NHPP.
2) Software product under goes failure at random times and causing the failure of the product.
3) All faults are categorized as either leading faults. The total number of faults in the system is finite.
4) The mean number of leading faults in the interval (t , t+ Δt] is proportional to the product of current testing effort and remaining leading faults in the system. This proportionality is constant over the time.
5) The mean number of detected faults in the time interval (t , t+Δt] is proportional to the product of current testing effort and remaining dependent faults.
6) The dependent faults are not removed immediately but delayed by the function φ(t).
7) No new faults are introduced in to the system.
8) Testing effort is inversely proportional to the pth power of learning factor [Xia et al ,1992]

According to the assumption 8 the current testing effort is given by

$$w(t) = \frac{k}{[f(t)]^p} \quad (17)$$

Where f(t) is a learning factor and an increasing function of t. Here we took the same learning function used by the Xia et al,1992 [23]

$$f(t) = \frac{(\alpha + \beta t)}{(1 + bt)} \quad (18)$$

And cumulative testing effort is given by eq (17)

$$\int_0^t w(t)dt = \int_0^t \frac{k(1+bt)^p}{(\alpha+\beta t)^p} \quad (19)$$

For the mathematical simplification we took p value as 2 now cumulative testing effort is given by

$$W(t) = \frac{k b^2 t}{\beta^2} + \frac{2 k b \ln(\alpha + \beta t)}{\beta^2} - \frac{2 k b^2 \ln(\alpha + \beta t)\, \alpha}{\beta^3} - \frac{k}{\beta(\alpha + \beta t)} + \frac{2 k b \alpha}{\beta^2 (\alpha + \beta t)} - \frac{k b^2 \alpha^2}{\beta^3 (\alpha + \beta t)} - \frac{2 k b \ln(\alpha)}{\beta^2} + \frac{2 k b^2 \ln(\alpha)\, \alpha}{\beta^3} + \frac{k}{\beta \alpha} - \frac{2 k b}{\beta^2} + \frac{k \alpha b^2}{\beta^3} \quad (20)$$

So from the above assumption (3) we have

$$a = a_1 + a_2 \quad (21)$$

We assume [15]

$$m(t) = m_1(t) + m_2(t) \qquad (22)$$

And also we assume testing effort function is same for the both leading and dependent faults.
Based on the above assumption (4) we get the following equation

$$\frac{dm_1(t)}{dt} \times \frac{1}{w(t)} = r \times [a_1 - m_1(t)] \qquad (23)$$

Solving the above equation based on the conditions $m_1(t) = 0$ the MVF we get

$$m_1(t) = a_1 \times (1 - \exp(-rW^*(t))) \qquad (24)$$

From the assumption 5 and 6 we have

$$\frac{dm_2(t)}{dt} \times \frac{1}{w(t)} = \theta \times [a_2 - m_2(t)] \times \frac{m_1(t - \varphi(t))}{a}$$

(25)

Case 1: if $\varphi(t) = 0$ then from the equation (20) we have

$$\frac{dm_2(t)}{dt} \times \frac{1}{w(t)} = \theta \times [a_2 - m_2(t)] \times \frac{m_1(t)}{a}$$

(26)

And

$$\frac{dm_2(t)}{dt} \times \frac{1}{w(t)} = \theta \times [a_2 - m_2(t)] \times \frac{a_1 \times (1 - \exp(-rW^*(t)))}{a}$$

(27)

Solving the above equation under boundary conditions $m_2(0) = 0$ we get

$$m_2(t) = a_2 \times [1 - e^{-p\theta \times (W^*(t) - \frac{(1 - \exp(-rW^*(t)))}{r})}]$$

(28)

Now let $a_1 = ap\, and\, a_2 = (1-p)a$ and from the equation 17 we get total MVF

$$m(t) = a_1 \times (1 - e^{-rW^*(t)}) + a_2 \times [1 - e^{-p\theta \times (W^*(t) - \frac{(1 - \exp(-rW^*(t)))}{r})}] \qquad (29)$$

And

$$m(t) = ap \times (1 - e^{-rW^*(t)}) + a(1-p) \times [1 - e^{-p\theta \times (W^*(t) - \frac{(1 - \exp(-rW^*(t)))}{r})}]$$

(30)

Case 2 : if $\varphi(t) = \frac{1}{r} \times \ln(1 + rW^*(t))$

(31)

From the equation 20 we obtain the following equation

$$\frac{dm_2(t)}{dt} \times \frac{1}{w(t)} = \theta \times [a_2 - m_2(t)] \times \frac{a_1 \times (1 - (1 + rW^*(t)) \exp(-rW^*(t)))}{a}$$

(32)

Solving the above equation based on condition $m_2(0) = 0$ we get

$$m_2(t) = a_2 \times [1 - e^{-[\frac{(\theta \times a_1)}{a} \times \left(W^*(t) + \frac{a_1 \times (1 - \exp(-rW^*(t)))}{r} \right)]}]$$

(33)

Required MVF

$$m(t) = a \times p \times \left(1 - (1 + rW^*(t)) \exp(-rW^*(t))\right) + a \times (1-p) \times [1 - e^{-[(\theta \times p) \times \left(W^*(t) + \frac{a \times p \times (1 - \exp(-rW^*(t)))}{r} \right)]}]$$

(34)

Case 3: if $\varphi(t) = \left(\frac{\ln(\psi + 1)}{(1 + \psi \times \exp(-rW^*(t)))} \right)$ then from the equation 20 we get

$$\frac{dm_2(t)}{dt} \times \frac{1}{w(t)} = \theta \times [a_2 - m_2] \times \frac{1}{a} \times \left(\frac{a_1 \times (1 - \exp(-rW^*(t)))}{(1 + \psi \times \exp(-rW^*(t)))} \right)$$

(35)

Solving the above equation based on the boundary conditions m(0)=0 we get the equation

$$m_2(t) = a_2 \times \left[1 - \frac{\left(\frac{(1+\psi)}{(1 + \psi \times \exp(-rW^*(t)))} \right)^{\left(\frac{\theta \times p}{\psi \times r} \right)}}{\left(\frac{(1 + \psi \times \exp(-rW^*(t)))}{(1+\psi) \times \exp(-rW^*(t))} \right)^{\left(\frac{\theta \times p}{r} \right)}} \right]$$

(36)

From above MVF

$$m(t) = \left(\frac{a \times p \times (1 - \exp(-rW^*(t)))}{(1 + \psi \times \exp(-rW^*(t)))} \right) + a \times (1-p) \times \left[1 - \frac{\left(\frac{(1+\psi)}{(1 + \psi \times \exp(-rW^*(t)))} \right)^{\left(\frac{\theta \times p}{\psi \times r} \right)}}{\left(\frac{(1 + \psi \times \exp(-rW^*(t)))}{(1+\psi) \times \exp(-rW^*(t))} \right)^{\left(\frac{\theta \times p}{r} \right)}} \right] \qquad (37)$$

IV Numerical Examples

A. Software failure data

First set (DS1) of actual data is from the study by Ohba(1984)[20]. The system is PL/1 data base application software , consisting of approximately 1,317,000lines of code .During nineteen weeks of experiments, 47.65 CPU hours were consumed and about 328 software errors are removed. Second dataset (DS2) used here is taken from the technical report for the project of Rector vessel Level Indication system used to monitor the level of water with in the rector vessel [4]. It took 25 weeks to complete the test. During the test phase, 230 software faults are removed.

Table 1: Failure dataset 2 (DS2)

Week	CNF	Week	CNF	Week	CNF	Week	CNF
1	44	8	100	15	197	22	230
2	75	9	124	16	205	23	230
3	75	10	130	17	214	24	230
4	75	11	130	18	215	25	230
5	75	12	159	19	225		
6	75	13	175	20	227		
7	75	14	181	21	228		

B Model comparison criterion

We use the following model comparison criterion where these will describe the best fit for given model.

1) Means Square Error :

Here we used MSE [M.Xie et al, 1991, C.Y Huang& Kuo 2007, H.Pham 2000] which gives real measure of the difference between actual and predicted values. The MSE defined as

$$MSE = \frac{\sum_{i=1}^{n}[m(t_i)-m_i]^2}{n} \quad (38)$$

A smaller MSE indicate a smaller fitting error and better performance.

2) Coefficient of Multiple determinations (R^2):[13]

Which measures the percentage of total variation about the mean accounted for the fitted model and tells us how well a curve fits the data. it is frequently employed to compare model and access which model provides the best fir to the data. The best model is that which provides higher value for R^2., which is close to 1.

3) SSE : it is calculates as [8]

$$SSE = \sum_{i=1}^{n}[y_i - m(t_i)]^2 \quad (39)$$

Here y_i is the total number of failures observed at a time t_i according to the actual data and $m(t_i)$ is the estimated cumulative number of failures at a time t_i.

C. Performance analysis

This section mainly constitutes the comparison of proposed model with some other models based on the above discussed datasets. Here we estimated the parameters by using LSE. Due to the complexity of the models we used SPSS package for the model parameter estimation. The estimated parameters of model 1 from eq(28) for dataset 1 is a=331.3, alpha=3, b=3.341, β=0.02767, k=0.25, p=0.248, r=0.5, θ=0.02828. The estimated parameters for model 2 from eq.(36) is 333.5, alpha=3.175, b=1.069, β=0.03 k=0.9996, p=0.245, ψ=0.5, r=0.99 and θ=0.07046 .

Fig.1 plots the comparison between observed failure data and the data estimated by model 1 eq(28). All estimated values for various models are listed in Table 2. From the Table 2 we can see that our proposed model has less MSE and SSE. The lower the value of MSE and SSE, it represents the best fit.

Fig 2 plots the cumulative numbers of failures versus time for the estimated and actual datasets (DS2). The comparisons for proposed and other models are shown in the table 3.

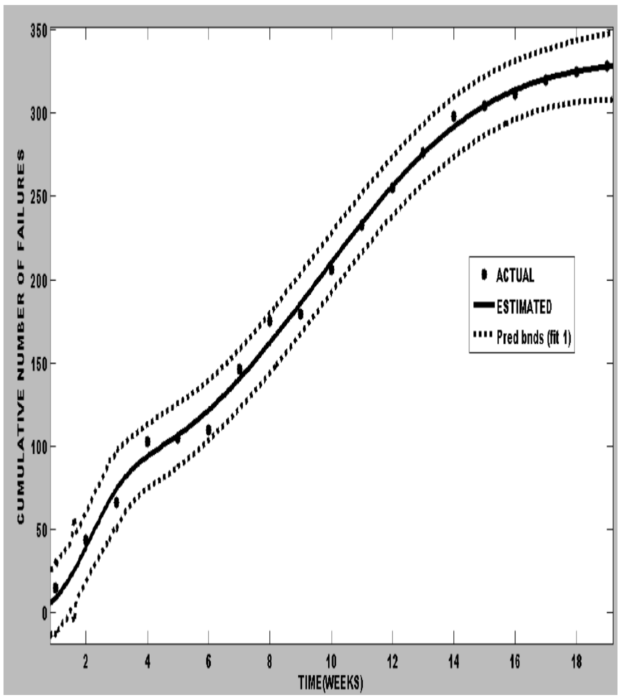

Fig 1. Cumulative Number of errors Versus Time for DS 1.

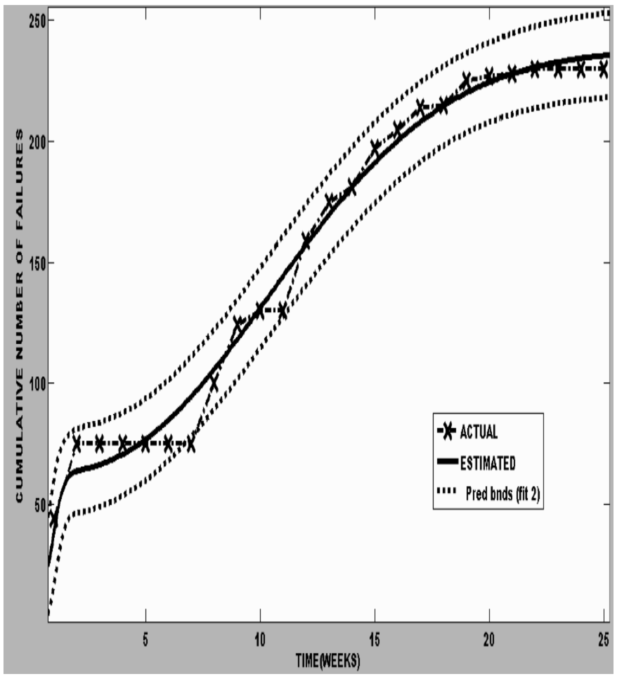

Fig 2. Cumulative number of errors Versus Time for DS2.

Table 2 ESTIMATED PARAMETER VALUES AND MODEL COMPARISION FOR DS1

Models	a	r	SSE	R^2	MSE
Model 1 from eq.(30)	331.3	0.5	662.8	0.9966	55.23
Model 2 from eq. (37)	333.5	0.99	631.5	0.9968	48.58
SRGM with exponentiated weibull TEF[13]	359.5	0.1416	1505	0.9923	88.36
Model 1 by Huang (et al, 2006)[3]	420.1	0.0759	--	--	89.36
Model 2 by Huang (et al, 2006)[3]	481.3	0.0199	--	--	92.83
SRGM with Logistic TEF[6]	395.6	0.0416	2167	0.989	127.46
Delayed S shaped model with Logistic TEF	319.3	0.1339	11060	0.9436	650.25
SRGM with Rayleigh TEF	459.1	0.02734	5100	0.974	299.98
Delayed S shaped model with Rayleigh TEF	333.2	0.1004	15170	0.9226	892.2
G-O model	760.5	0.03227	2656	0.9865	156.2
Yamada Delayed S shaped model	374.1	0.1977	3205	0.9837	188.51

Table 3 ESTIMATED PARAMETER VALUES AND MODEL COMPARISION FOR DS2

Models	a	r	SSE	R^2	MSE
Model 1 from eq.(30)	238.4	0.99	1217	0.9885	57.95
G-O model	326.4	0.5569	6330	0.9404	275.2
Yamada Delayed S shaped model	247.2	0.191	10230	0.9037	448.78

From the table 3 we can see that our proposed model has less MSE and SSE value, which shows our model best fit for the dataset (DS2). The estimated model 1 parameters for the dataset 2 is a=238.4, alpha=5.814, b=5.6, β=0.25, k=2, p=0.2634, r=0.98, and θ=0.004946.

The SSE values and MSE values for these models are very less than other models. Lower values of predicted values indicates our models better fit for the datasets than other models.

Based on above parameters we made an analysis that the SRGM is totally influenced by the nature of the faults. The number of initial faults are influencing the system more compared with other factors. And other two important factors line fault detection and correction also play an important role.

Here we have added a new concept testing effort we defines as number test cases , manpower and time of testing. There is substantial relation between testing effort and the parameter p.The dynamic changes in the value of p indicates and give the information to managers how much effort they required. Depending on the p value they can assign the required resources

V CONCLUSION

In this paper we have designed a new models testing effort and learning dependent SRGM with correction lag. Correction delay is considered to be an important factor in software reliability models where time of correction delay is not been neglected. Parameters of new models are estimated on real datasets, and comparisons are done to see the goodness of fit techniques. Our proposed models are better fit than other models.

REFERENCES

[1] A. Wood, Predicting software reliability, IEEE computers 11 (1996) 69–77.

[2] A.L Goel, and Okumoto, K., "*Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures*", *IEEE Trans. Reli. 20,206-2110979).*

[3] C.Y.Huang, and Lin, C. D., 2006, "*Software reliability analysis by considering fault dependency and debugging time lag*", *IEEE Transactions on Reliability*, vol. 55, no. 3, pp. 436-450.

[4] C.Y Huang ,C. T. Lin , H. K.Lo, Y.S. Su and B.T Lin " *introduction to software Reliability and its applications*" Technical Report ,NHTU EECS industrial affiliates program , Jan 2004.

[5] C.Y.Huang, Lyu M.R "*Optimal Release time for Software systems Considering Cost, Testing-effort and Test efficiency*" IEEE Transaction on reliability VOL 54 No , December 2005.

[6] C.Y Huang and S. Y. Kuo, "*Analysis and assessment of incorporating logistic testing effort function into software reliability modeling,*" *IEEE Trans. Reliability*, vol. 51, no. 3, pp. 261–270, Sept. 2002.

[7] C.Y Huang, Lyu and Kuo "*An Assesment of testing effort dependent software reliability Growth model*". IEEE transactions on Reliability Vol 56, No: 2, June 2007

[8] H.Pham, (2000), "*Software Reliability, Springer-Verlag"*, New York, NY.

[9] J.D Musa, Iannino, A., Okumoto, K., 1987. "*Software Reliability, Measurement, Prediction and Application"*. McGraw Hill.

[10] M. Xie, 1991, *Software reliability modeling,* World Scientific, Singapore.

[11] M. Xie, and Yang, B. (2003). "*A study of the effect of imperfect debugging on software development cost.*" *IEEE Trans. Software Eng.* 29, 471-473.

[12] M. Xie, Hu, Q. P., Wu, Y. P., Ng, S. H., 2007, "*A study of the modeling and analysis of software fault-detection and fault-correction processes*", Quality and Reliability Engineering International, vol. 23, no. 4, pp. 459-470.

[13] N. Ahmad, Bokhari M U, Quadri S M K, Khan MGM "*The exponentiated Weibull software reliability growth model with various testing-efforts and optimal release policy*" *International Journal of Quality Reliability Management*, 2008, 25(2) : 11-235 .

[14] N.F Schneidewind, 2003, "*Fault correction profiles",* Proceedings of the 14thInternational Symposium on Software Reliability Engineering, pp. 257-267.

[15] P.K Kapur, Younes, S., 1995. "*Software reliability growth model with error dependency"*. Microelectronics and Reliability 35 (2), 273–278..

[16] S. Yamada, and Osaki,S. (1985), "*Cost-reliability optimal release policies for software systems*", IEEE Transactions on Reliability, 34(5), 422-424.

[17] S. Yamada, Hishitani, J., Osaki, S., 1993. "*Software reliability growth model with Weibull testing effort: a model and application*". IEEE Transactions on Reliability 42, 100–105.

[18] S. Yamada, Ohba, M., Osaki, S. (1983). "*S-shaped reliability growth modeling for software error detection". IEEE Trans. Reliability.* 12, 475–484.

[19] S. Yamada, Ohba,M., and Osaki.S. (1984), "S–shaped software reliability growth models and their applications", IEEE Transactions on Reliability, 33(4), 289–292.

[20] M.Ohba, *"Software reliability analysis models".* IBM Journal of Research and Development 1984, 28 (4), 428–443.

[21] Y.P. Wu, Hu, Q. P., Xie, M. and Ng, S. H., 2007, 'Modeling and analysis of software fault detection and correction process by considering time dependency', *IEEE Transactions on Reliability*, vol. 56, no. 4, pp.629-642.

[22] Yanjun Shu, Hongwei Liu, Zhibo Wu and Xiaozong Yang " Modeling of software fault detection and correction Peocess Based on Correction lag, 2009.

[23] Xia G, Zeephongsekul P, Kumar S " Optimal Software release policies for models incorporating learning in testing " Asia apecific journal of Operation research , 9, 221-234, 1992.

SK.MD.Rafi: Received Bachelor of Technology in Computer Science And Engineering, Master of Computer Science and Engineering, and pursuing PhD(computer science and Engineering) from J.N.T university Kakinada. Presently working as Assoc. Professor in the Department of Computer science and engineering in Sri Mittapalli College of Engineering. Area of interest software engineering, reliability and quality control, Software Architecture Recovery. Published many research papers in various International journals (mdrafi.527@gmail.com)

Dr. K. Nageswara Rao: Received PhD from computer science and engineering from Andhra University. Presently working as Head & Professor in PVP Siddhartha engineering college. Published many papers all over the world and guiding many research scholars. Here is a member of IEEE and CSI.

Dr. S. Pallam Setty : Received PhD from computer science and engineering from Andhra University Presenty working as a Professor in the department of computer science and engineering , Andhra university Visakhapatanam. Here is member of many professional bodies and guiding the many research scholars.

Dr. Shaheda akthar : received Bachelor of Computer Science and Master of Computer Science from Acharya Nagarjuna University, M.S from B.I.T.S Pilani. Ph.D from Acharya Nagarjuna University.Presently working as Principal in Sri Mittapalli Institute of technology for women. Area of interest software engineering, reliability and quality control, Software Architecture Recovery. Published many research papers in various International journals