Proceedings of the National Conference; INDIACom-2007
Computing for Nation Development, February 23 – 24, 2007
Bharati Vidyapeeth's Institute of Computer Applications and Management, New Delhi

A Flexible Software Reliability Growth Model Incorporating Change Point with Imperfect Debugging and Fault Generation

V.K.Sehgal[1] **Reecha Kapur[1]** **Kalpana Yadav [2]** **Deepak Kumar[3]**

[1] *Department of Mathematics and Computer Application, Bundelkhand University, Jhansi, richakapur1@rediffmail.com*

[2] *Indira Gandhi Institute, G.G.S.I.P.University, kyadav11@yahoo.com*

[3] *J.S.S. Academy Of Technical Education, Noida, U.P., deepakgupta_du@rediffmail.com*

ABSTRACT

There exist various Software Reliability Growth models(SRGMs) in literature but they rarely differentiate between the failure observation and fault removal process. In real software development environment, the number of failures observed need not be same as the number of faults removed. As the testing grows and testing team gains experience, additional number of faults are removed without their causing any failure resulting in lesser failure observation than the fault removal. If the number of failures observed is more than the number of faults removed then we have the case of imperfect debugging. Due to the complexity of the software system and the incomplete understanding of the software, the testing team may not be able to remove the fault perfectly on the detection of the failure and the original fault may remain or get replaced by another fault. While the former phenomenon is known as imperfect fault debugging, the latter is called fault generation. In case of imperfect fault debugging the fault content of the software remains same while in case of fault generation the fault content increases as the testing progresses and removal results in introduction of new faults while removing old ones. Attempts have been made to study the above cases separately. Most of the SRGMs are based upon constant or monotonically increasing Fault Detection Rate (FDR). In practice, as the testing grows, so does the skill and efficiency of the testers. With the introduction of new testing strategies and new test cases, there comes a change in FDR. The time point where the change in removal curve appears is termed as 'change point'. In this paper we incorporate the concept of change point in Software Reliability Growth in the presence of imperfect debugging and fault generation. The model has been validated, evaluated and compared with other existing non homogenous poisson process(NHPP) models by applying it on actual failure / fault removal data sets cited from real software development projects. The results show that the proposed model provides improved goodness of fit and predictive validity for software failure / fault removal data.

KEYWORDS

Non homogenous poisson process, software reliability growth models, fault detection rate, imperfect debugging, change point

1.INTRODUCTION

It is virtually impossible to conduct many day-to-day activities without the aid of computer systems controlled by software. As more reliance is placed on these software systems to operate in a reliable manner, the failure to do so can result in high property, monetary or human loss. So, there is a need for effective and well planned testing. Testing is an important part of the software development process.

Several SRGMs have been developed in the literature to estimate the fault content and fault removal rate per fault in software. Goel and Okumoto [6] have proposed NHPP based SRGM assuming that the failure intensity is proportional to the number of faults remaining in the software describing an exponential failure curve. Ohba [13] refined the Goel-Okumoto model by assuming that the fault detection / removal rate increases with time and that there are two types of faults in the software. SRGM proposed by Bittanti et al. [1] and Kapur and Garg [10] have similar forms as that of Ohba [13] but are developed under different set of assumptions. These models can describe both exponential and S-shaped growth curves and therefore are termed as flexible models.

In most of the models discussed above it is assumed that whenever an attempt is made to remove a fault it is removed with certainty i.e. a case of perfect debugging. In practical software development scenario, the number of failure observed can be less than or more than the number of error removed. Kapur and Garg [10] has discussed the first case in their Error removal phenomenon flexible model which shows as the testing grows and testing team gain experience, additional number of faults are removed without them causing any failure. But if the number of failure observed is more than the number of error removed then we are having the case of imperfect debugging.

The testing team may not be able to remove the fault perfectly on the detection of the failure and the original fault may remain or replaced by another fault because of the incomplete understanding of the internal structure of the software. While the first phenomenon is known as imperfect debugging, the second is called fault generation. In case of imperfect debugging the fault content of the software is not changed, but because of incomplete understanding of the software the detected fault is not removed completely. But in case of error generation the fault content increases as the testing progresses and removal results in introduction of new faults while removing old ones.

It was Goel [5] who first introduced the concept of imperfect debugging. He introduced the probability of imperfect debugging in Jelinski and Moranda [7]. Model due to Chou and Obha [14] is a fault generation model as applied on GO model and has been also named as Imperfect debugging model. Kapur and Garg [11] introduced the imperfect debugging in Goel and Okumoto [6]. Recently, Pham et. al.[19] proposed a testing efficiency model which includes both imperfect debugging and fault generation, modeling it on the number of failures experienced or observed, but both imperfect debugging and fault generation is actually seen during fault removal. Recently, Kapur et. al.[9] proposed a flexible SRGM with imperfect debugging and fault generation using a logistic function for fault detection rate which reflects the efficiency of the testing team.

Most of the SRGMs developed till yet are based upon constant or monotonically increasing Fault Detection Rate (FDR). As the testing progresses, the testing team gains experience and with the employment of new tools and techniques, the fault detection rate gets changed. This change can also be caused by shift in testing strategy, defect density, introduction of new test cases, and induction of skilled personnel in team or simply by the increase in efficiency of present team. The point of time where the change in FDR is observed can be termed as 'Change Point'. Very few attempts have been made to incorporate the 'Change point' in failure growth phenomenon. The work in this area started with Zhao [20] who introduced the change point analysis in Hardware and Software reliability. Some pioneering work has been done in the area by Shyur [17], Chang [4], Wang [18]. The position of the change point can be judged from the graph of the actual failure data.

In this paper, a flexible SRGM incorporating change point in fault generation and imperfect debugging with learning has been proposed. The proposed model has been validated and evaluated on two actual software failure / fault removal data sets and compared with other existing models under perfect and imperfect environment. For estimation of parameters of the proposed model, SPSS is used. **SPSS** is a Statistical package for Social Sciences.

The paper is organized as follows: Section 2 describes the Non Homogenous Poisson Process and the assumptions for the proposed model. Section 3 discusses the model development under imperfect debugging and fault generation incorporating the concept of change point. Sections 4 and 5 provide the method used for parameter estimation and the criteria used for validation and evaluation of the developed model respectively. Section 6 gives the estimated results of the developed model to actual software reliability data sets collected from real software development projects. Section 7 concludes the paper.

Notation

$m(t)$	: the mean value function or the expected number of faults detected by time t
a	: constant, representing the initial number of faults lying dormant in the software when the testing starts.
$a(t)$	: total fault content of the software dependent on the time.
$\lambda(t)$	: intensity function or fault detection rate per unit time
$b1, b2, c$	: constants
p	: the probability of fault removal on a failure (i.e., the probability of perfect debugging).
$\alpha 1$, $\alpha 2$	: the rate at which the faults may be introduced during the debugging process per detected fault before and after change point.
β	: a constant parameter in the logistic function.
$b(p,t)$	: rate of fault removal per remaining fault for a software under probability of perfect debugging p.
τ	: change point i.e. a change in removal curve from where the FDR change.

2. Basic Assumption

The SRGM presented in this paper is based upon NHPP. The NHPP models are based on the assumption that the software system is subject to failures at random times caused by manifestation of remaining faults in the system. Hence NHPP are used to describe the failure phenomenon during the testing phase. The counting process $\{N(t), t \geq 0\}$ of an NHPP process is given as follows.

$$\Pr\{N(t)=k\} = \frac{(m(t))^k}{k!} e^{-m(t)}, \qquad k = 0,1,2... \quad (1)$$

and $$m(t) = \int_0^t \lambda(x)\,dx \quad (2)$$

The intensity function $\lambda(x)$ (or the mean value function $m(t)$) is the basic building block of all the NHPP models existing in the software reliability engineering literature.

The proposed model is based upon the following basic assumptions:

1. Failure observation / fault removal phenomenon is modeled by NHPP.
2. Software is subject to failures during execution caused by faults remaining in the software.
3. Each time a failure is observed, an immediate effort takes place to find the cause of the failure in order to remove it.

4. Failure rate is equally affected by all the faults remaining in the software.
5. When a software failure occurs, an instantaneous repair effort starts and the following may occur:
   (a) Fault content is reduced by one with probability p
   (b) Fault content remains unchanged with probability 1-p.
6. During the fault removal process, whether the fault is removed successfully or not, new faults are generated with a constant probability α.
7. Fault removal rate is assumed to be non-decreasing inflection S-shaped logistic function to describe the learning effect of the fault removal team

Assumption 5 and 6 captures the effect of imperfect debugging and error generation respectively, whereas assumption 7 incorporates the learning of testing team.

3. Modeling Software Reliability

3.1. Proposed SRGM

Assuming that fault removal rate per additional fault removed is not only a function of time but is a function of both time and probability of perfect debugging and a constant proportion of removed faults are generated while removal, the differential equation describing the removal phenomenon can be given by

$$\frac{dm_r(t)}{dt} = b(p,t)(a(t) - m_r(t)) \qquad (3)$$

where

$$b(p,t) = \begin{cases} \dfrac{b_1 p}{1+\beta e^{-b_1 pt}} & \text{when } 0 \le t \le \tau \\ \dfrac{b_2 p}{1+\beta e^{-b_2 pt}} & \text{when } t > \tau \end{cases}$$

and

$$a(t) = \begin{cases} a + \alpha_1 m_r(t) & \text{when } 0 \le t \le \tau \\ a + \alpha_2 m_r(t) & \text{when } t > \tau \end{cases}$$

where τ is the change point.

Case 1: For $0 \le t \le \tau$

$$\frac{dm_r(t)}{dt} = \frac{b_1 p}{1+\beta e^{-b_1 pt}}((a + \alpha_1 m_r(t)) - m_r(t)) \qquad (4)$$

Solving the above differential equation (4) under initial condition $m_r(0)=0$, we get mean value function as

$$m_r(t) = \frac{a}{(1-\alpha_1)}\left[1 - \left(\frac{(1+\beta)\exp(-b_1 pt)}{1+\beta\exp(-b_1 pt)}\right)^{(1-\alpha_1)}\right] \qquad (5)$$

Case 2: For $t > \tau$

$$\frac{dm_r(t)}{dt} = \frac{b_2 p}{1+\beta e^{-b_2 pt}}((a + \alpha_2 m_r(t)) - m_r(t)) \qquad (6)$$

Solving the above differential equation (6) under initial condition at $t=\tau$, $m_r(t)= m_r(\tau)$, we get mean value function as

$$m_r(t) = \frac{a}{(1-\alpha_2)}\left[1 - \left(\frac{(1+\beta\exp(-b_2 p\tau))\exp(-b_2 p(t-\tau))}{1+\beta\exp(-b_2 pt)}\right)^{(1-\alpha_2)}\right] + \frac{a}{(1-\alpha_1)}\left[1 - \left(\frac{(1+\beta)\exp(-b_1 p\tau)}{1+\beta\exp(-b_1 p\tau)}\right)^{(1-\alpha_1)}\right]\left(\frac{(1+\beta\exp(-b_2 p\tau))\exp(-b_2 p(t-\tau))}{1+\beta\exp(-b_2 pt)}\right)^{(1-\alpha_2)} \qquad (7)$$

Here, $m_r(t) = \dfrac{a}{(1-\alpha)}$, When t tends to ∞ and $\alpha_1 = \alpha_2 = \alpha$ which implies that if testing is carried out for an infinite time more faults are removed as compared to the initial fault content because there are some error added to the software due to error generation.

4. Parameter Estimation

The success of mathematical modeling approach to reliability evaluation depends heavily upon quality of failure data collected. The parameters of the SRGMs are estimated based upon these data. The model discussed in this paper is a non-linear model and it is difficult to find solution for nonlinear models using Least Square method and require numerical algorithms to solve it.

Statistical software packages such as SPSS help to overcome this problem. SPSS is a statistical package for Social Sciences. SPSS Regression Models enables the user to apply more sophisticated models to the data using its wide range of nonlinear regression models. For the estimation of the parameters of the proposed model method of Least Square has been used. Non-linear regression is a method of finding a nonlinear model of the relationship between the dependent variable and a set of independent variables.

5. Model Validation and Comparison Criteria

5.1. Model Validation

To assess the performance of the proposed SRGM incorporating change point with imperfect debugging and fault generation using logistic form of b(t) to indicate learning of the testing team, we have carried out the parameter estimation on two real software failure datasets.

Data set 1(DS-1)

The first data set (DS-1) had been collected during 35 months of testing a radar system of size 124 KLOC and 1301 faults were detected during testing. This data is cited from Brooks and Motley [2].The position of the change point can be judged from the graph of the actual failure data. In DS-1, the change point (τ) is taken to be 25.

Data set 2(DS-2)

The second data set (DS-2) had been collected during 19 weeks of testing a real time command and control system and 328 faults were detected during testing. This data is cited from Ohba et. al. [13].The position of the change point can be judged from the graph of the actual failure data. In DS-1, the change point (τ) is taken to be 13.

5.2. Comparison Criteria for SRGMs

The performance of SRGMs are judged by their ability to fit the past software fault data (goodness of fit) and predicting the future behavior of the fault data knowing the past and present software fault data (Predictive Validity Criterion).

5.2.1. Goodness of Fit criteria

The term goodness of fit is used in two different contexts. In one context, it denotes the question if a sample of data came from a population with a specific distribution. In another context, it denotes the question of "How good does a mathematical model (for example a linear regression model) fit to the data"?

a. The Mean Square Fitting Error (MSE):

The model under comparison is used to simulate the fault data, the difference between the expected values, $\hat{m}(t_i)$ and the observed data y_i is measured by MSE as follows.

$$MSE = \sum_{i=1}^{k} \frac{(\hat{m}(t_i) - y_i)^2}{k} \tag{8}$$

where k is the number of observations. The lower MSE indicates less fitting error, thus better goodness of fit[12].

b. Coefficient of Multiple Determination (R^2):

We define this coefficient as the ratio of the sum of squares resulting from the trend model to that from constant model subtracted from 1.

i.e. $$R^2 = 1 - \frac{\text{residual SS}}{\text{corrected SS}}. \tag{9}$$

R^2 measures the percentage of the total variation about the mean accounted for the fitted curve. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well. The larger R^2, the better the model explains the variation in the data[12].

c. Bias:

The difference between the observation and prediction of number of failures at any instant of time i is known as PE_i (prediction error). The average of PEs is known as bias. Lower the value of Bias better is the goodness of fit [15].

d. Variation:

The standard deviation of prediction error is known as variation.

$$Variation = \sqrt{\left(\frac{1}{N-1}\right)\sum (PE_i - Bias)^2} \tag{10}$$

Lower the value of Variation better is the goodness of fit [15].

e. Root Mean Square Prediction Error:

It is a measure of closeness with which a model predicts the observation.

$$RMSPE = \sqrt{\left(Bias^2 + Variation^2\right)} \tag{11}$$

Lower the value of Root Mean Square Prediction Error better is the goodness of fit [15].

f. The Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov test (K-S test) [3] is a non-parametric test. It tries to determine if two datasets differ significantly. The Kolmogorov-Smirnov (K-S) test is based on the empirical distribution function (ECDF). Since it is non-parametric, it treats individual observations directly and is applicable even in the case of very small sample size, which is usually the case with SRGM validation.

Given *N ordered* data points $\boldsymbol{Y_1}, \boldsymbol{Y_2}, ..., \boldsymbol{Y_N}$, the ECDF is defined as

$$E_N = n(i)/N \tag{12}$$

where $\boldsymbol{n(i)}$ is the number of points less than $\boldsymbol{Y_i}$ and the $\boldsymbol{Y_i}$ are ordered from smallest to largest value. This is a step function that increases by $1/N$ at the value of each ordered data point.

The Kolmogorov-Smirnov test statistic is defined as

$$D = \max_{1\le i\le N}\left(F(Y_i) - \frac{i-1}{N}, \frac{i}{N} - F(Y_i) \right) \tag{13}$$

where $\boldsymbol{F}$ is the theoretical cumulative distribution of the distribution being tested which must be a continuous distribution (i.e., no discrete distributions such as the binomial or Poisson), and it must be fully specified (i.e., the location, scale, and shape parameters cannot be estimated from the data). Lower the value of Kolmogorov-Smirnov test better is the goodness of fit.

5.2.2. Predictive Validity Criterion

Predictive validity is defined as the ability of the model to determine the future failure behavior from present and past failure behavior. Suppose t_k be the testing time, x_k is number of faults detected during the interval $(0, t_k]$, and $\hat{m}(t_k)$ 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 testing time $t_e (0 < t_e \le t_k)$, in which (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 $\hat{m}(t_k)$ and the reported value x_k measures the prediction fault. The ratio $\{(\hat{m}(t_k) - x_k)/x_k\}$ 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 close to zero for RPE indicates more accurate prediction, thus more confidence in the model and better predictive validity. The value of RPE is said to be acceptable if it is within $\pm(10\%)$ (Kapur et al. [12]).

6. Data Analyses and Model Comparison

Here we assume that location of change point τ can be judged from the data curve and hence need not be estimated. Using Non-linear regression technique remaining parameters of proposed SRGM(7) are estimated. To judge the predictive power and accuracy of the proposed SRGM, we have compared the results to the those of existing SRGMs like Ohba-Chou[14], P-N-Z[16], Z-T-P[19], two models by Kapur et. al.[9] and used MSE and R^2 , bias, variation, RMSPE, Kolmogorov-Smirnov test, predictive validity criterion as the performance measures.

6.1. For DS-1

The estimated values of parameters of mean value function *m*(*t*) given by (7) are worked out. For this purpose, we assume that change point τ can be judged from the data and need not be estimated. To identify the location of change point, the graph of cumulated number of faults is evaluated and wherever a sudden change in detection rate is observed, the corresponding time point is termed as change point. For DS- 1, the change is observed at around 25th value. So here we take τ=25. The estimation results are provided in Table-1 while the comparison criteria results are shown in Table-2.

Models under Comparisons	Parameter Estimation							
	a	b_1	b_2	α_1	α_2	β	p	c
Obha-Chou Model	7889	.006	-	.230	-	-	-	-
P-N-Z Model	1305	.198	-	.001	-	19	-	-
Z-T-P Model	1519	.003		.137	-	.57	.41	.0002
Kapur et. al. Model 1	1331	.201	-	.001	-	20	.99	-
Kapur et. al. Model 2	1330	.234	-	.001	-	20	.86	-
Proposed SRGM	1319	.2499	.2	.001	.0025	25	.8535	-

Table-1: Model Parameter Estimation Results

If we carefully observe the estimation results for DS-1, we observe that p=.8535 for proposed SRGM which represents a large fraction(85.35%) of the detected faults are removed perfectly while only a very small fraction faults remain even after being detected. The learning is high with β=25. Here we observe that b_2 is less than b_1. It shows the slow down in detection rate after the change point. Here, α1=.001 which is very low. Also, α2 is slightly greater than α1 which indicates that fault generation is increased a bit after the change point.

Models under Comparisons	Comparison Criteria					
	MSE	BIAS	VARIATION	R^2	RMSPE	K-S statistic
Obha-Chou Model	8924	18.58	93.974	.95810	95.79	.995
P-N-Z Model	236	-0.57	15.571	.99889	15.58	.484
Z-T-P Model	7760	16.96	87.704	.96356	89.33	.991
Kapur et. al. Model 1	206	-2.148	14.407	.99896	14.57	.453
Kapur et. al. Model 2	204	-2.148	14.327	.99904	14.49	.452
Proposed SRGM	150	-0.006	12.419	.99930	12.42	.426

Table-2: Model Comparison Results

From Table-2, we can observe that the proposed SRGM gives the best goodness-of-fit when compared to other existing models.

DS-1 is truncated into different proportions and used to estimate the parameters of the proposed model. For each truncation, one relative value is obtained. It is observed that the predictive validity of the model varies from one truncation to another and provides better predictive power to forecast the future fault behavior of the software data. The RPE estimation results are shown in Table-3 for proposed SRGM. For DS-1, we observe that even 75 % of data is sufficient to predict the future failure behavior well with RPE as low as 1.628% for proposed SRGM.

Model	(t_e/t_k)	$m(t_k)$	RPE	RPE(%)
Proposed Model	100%	1301.68	0.00052	0.052
	95%	1311.32	0.00793	0.793
	90%	1306.83	0.00448	0.448
	85%	1305.06	0.00311	0.311
	80%	1313.09	0.00929	0.929
	75%	1322.18	0.01628	1.628

Table-3: RPE Estimation Results

The fitting of the proposed model with change point to DS-1 is graphically illustrated in Figure 1. It clearly shows that the model fits the data excellently. Figure 2 illustrate graphically the retrodictive and predictive ability of the proposed model. In each case the DS-1 is truncated at t_e (75% approx.) to estimate the model parameters. The model is then used to estimate the whole DS. The points below t_e (marked by the intersection of the horizontal line with the curve) demonstrate the retrodictive ability while the points above t_e demonstrate the predictive ability of the model.

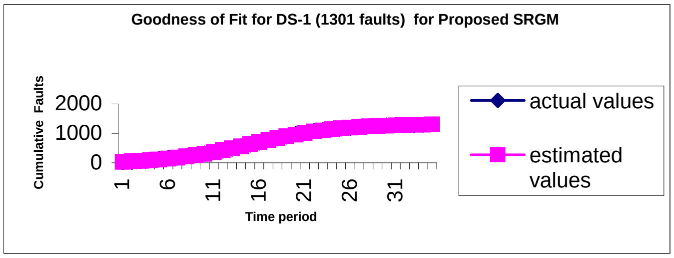

Figure 1: Goodness-of-Fit curve

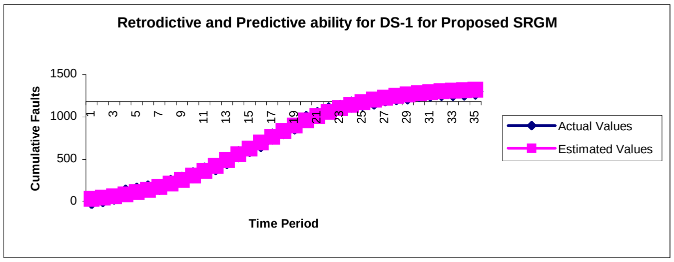

Figure 2: Retrodictive and Predictive ability

6.2. For DS-2

For DS-2, the change is observed at around 13th value. So here we take τ=13. The estimation results are provided in Table-4 while the comparison criteria results are shown in Table-5.

Models under Comparisons	Parameter Estimation							
	a	b_1	b_2	α_1	α_2	β	p	c
Obha-Chou Model	668	.04	-	.122	-	-	-	-
P-N-Z Model	305	.21		.013	-	2.78	-	-
Z-T-P Model	402	.003	-	.081	-	.515	.8	.0004
Kapur et. al. Model 1	386	.186	-	.001	-	2.65	.9	-
Kapur et. al. Model 2	382	.201	-	.001	-	2.88	.8	-
Proposed SRGM	358	.211	.2	.0001	.058	3.73	.99	-

Table-4: Model Parameter Estimation Results

If we carefully observe the estimation results for DS-2, we observe that p = .999 for proposed SRGM which represents a large fraction(99%) of the detected faults are removed perfectly while only a very small fraction faults remain even after being detected. The learning is quite fine with β=3.73. Here we observe that *b2* is less than *b1* in the proposed SRGM. It shows the slow down in detection rate after the change point. Here, α1=.0001 which is very low. Also, α2 is more than α1 in case of proposed SRGM which indicates that fault generation is increased after the change point.

Models under Comparisons	Comparison Criteria					
	MSE	BIAS	VARIATION	R^2	RMSPE	K-S statistic
Obha-Chou Model	90.85	-1.03	9.735	.98645	9.79	.998
P-N-Z Model	139.83	1.16	12.090	.99120	12.1457	.969
Z-T-P Model	116.65	0.43	11.088	.98870	11.096	.994
Kapur et. al. Model 1	83.84	-0.51	9.393	.99188	9.4065	.866
Kapur et. al. Model 2	82.70	-0.52	9.328	.99199	9.3423	.864
Proposed SRGM	77.60	0.06	9.050	.99248	9.0507	.862

Table-5: Model Comparison Results

The RPE estimation results are shown in Table-6 for proposed. For DS-2, we observe that even 85 % of data is sufficient to predict the future failure behavior well with RPE as low 9.737% for proposed SRGM.

Model	(t_e/t_k)	$m(t_k)$	RPE	RPE(%)
Proposed Model	100%	337.08	0.02768	2.768
	95%	339.69	0.03564	3.564
	90%	347.038	0.05804	5.804
	85%	359.94	0.09737	9.737
	80%	390.55	0.19071	19.071
	75%	407.06	0.24102	24.102

Table-6: RPE Estimation Results

The fitting of the proposed model with change point to DS-2 is graphically illustrated in Figure 3. Figure 4 illustrate graphically the retrodictive and predictive ability of the proposed model. It clearly shows that the model fits the data excellently.

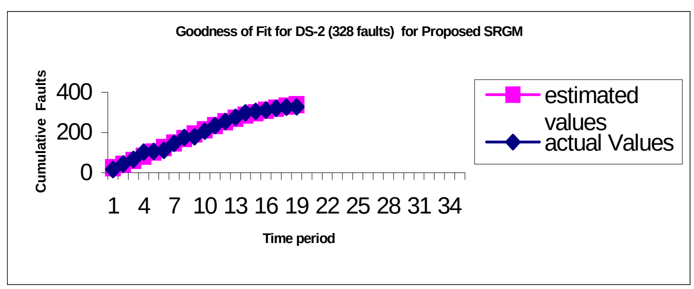

Figure 3: Goodness-of-Fit curve

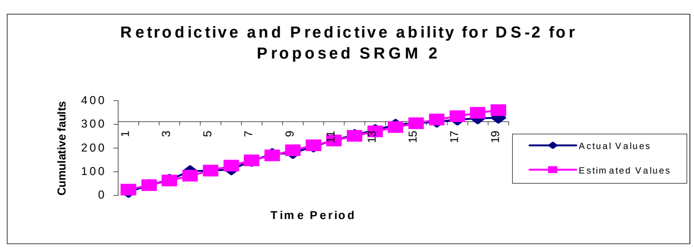

Figure 4: Retrodictive and Predictive ability

Continued on Page No.2 68

Continued from Page No.2 74

From Figures 1, 2, 3 and 4, it can be observed that the proposed SRGMs not only fits the past well but also predicts the future reasonably well.

7. Conclusion

In this paper, new SRGMs with two types of imperfect debugging has been presented. The first type, less damaging, is the case of imperfect debugging where all detected errors are not removed completely resulting in the same fault content of the software. The second type, known as error generation, describes the situation when each error removal attempt increases the fault content of the software. The concept of learning has been incorporated in the FDR to show the gain in experience of the testing team as the testing grows. As the testing progresses, the testing team gains experience and with the employment of new tools and techniques, the fault detection rate gets changed. Hence, the concept of change point has been introduced. In a nut shell, SRGMs incorporating change point with imperfect debugging and fault generation using learning function for FDR have been developed. The results show that the proposed models provides improved goodness of fit and predictive validity for software failure occurrence / fault removal data due to their applicability and flexibility. The study of change point is not limited to the area of Software Reliability but it can be extended to other application-oriented fields like Distributed Software systems, Hardware Reliability or Marketing areas.

8. Future Scope

In this paper, we have used only one type of fault generation function and a single change point. If the software being developed is big and testing is supposed to continue for considerably large interval of time then it is quite possible that change in FDR is observed more than once. The frequent changes in testing team, test cases or the management can alter the overall testing growth resulting in changes in removal rate.

So, further new SRGMs can be modeled using different fault generation functions existing in software development literature and introducing multiple change point concept.

References

1. Bittanti, S., P. Blonzera, E. Pedrotti, M. Pozzi and A. Scattolini, *A Flexible Modeling Approach in Software Reliability Growth*, Goos G. & Hartmanis (Editors), Software Reliability Modeling & Identification, Springer-Verlag, Berlin, pp. 101-140, **1988**.

2. Brooks, W. D. and R.W. Motley, *Analysis of Discrete Software Reliability Models*-Technical Report (RADC-TR-80-84), Rome Air Development Center, New York, **1980**.

3. Chakravart; Laha, *Roy Handbook of Methods of Applied Statistics*, Volume I, John Wiley and Sons, pp. 392-394,**1976**.

4. Chang, Y.P., *Estimation of Parameters for Nonhomogenous Poisson Process Software Reliability with Change-Point Model,* Communications in Statistics- Simulation and Computation, 30, pp. 623-635, **2001**.

5. Goel, A.L., *Software Reliability Models: Assumptions, Limitations and Applicability,* IEEE Transactions on Software Engineering; SE-11, pp. 1411-1423, **1985**.

6. Goel, A.L. and K. Okumoto, *Time Dependent Error Detection Rate Model for Software Reliability and other Performance Measures*, IEEE Transactions on Reliability, R-28(3), pp. 206-211, **1979**.

7. Jelinski, Z. and P. B. Moranda, *Software Reliability Research,* Freiberger W., *(Ed.)* Statistical Computer Performance Evaluation, New York, Academic Press, pp. 465-497, **1972**.

8. Kapur P.K, Anu Gupta, Omar Shatnawi and V.S.S.Yadavalli, *Testing Effort Control using Flexible Software Reliability Growth Model with Change Point*, International Journal of Performability Engineering, Vol. 2(3), pp. 245-262, **2006**.

9. Kapur P. K., Kumar D., Gupta A., Jha P. C., *On How To Model software Reliability Growth in the Presence Of Imperfect Debugging and Fault Generation*, Proceedings of 2nd International Conference on Reliability and safety Engineering, pp. 515-523, **2006**.

10. Kapur, P. K. and R. B. Garg, *A Software Reliability Growth Model for an Error Removal Phenomenon,* Software Engineering Journal, Vol. 7, pp. 291-294, **1992**.

11. Kapur, P. K., and R. B. Garg, *Optimal Software Release Policies for Software Reliability Growth Model under Imperfect Debugging*, RAIRO, Vol. 24, pp. 295-305, **1990**.

12. Kapur, P. K., R. B. Garg and S. Kumar, *Contributions to Hardware and Software Reliability,* Singapore, World Scientific Publishing Co. Ltd., **1999**.

13. Ohba, M., *Software Reliability Analysis Models,* IBM Journal of Research and Development, Vol. 28, pp. 428-443, **1984**.

14. Ohba M. and Chou, X.M., "*Does Imperfect Debugging Effect Software Reliability Growth*", proceedings of 11th International Conference of Software Engineering, pp 237-244, **1989.**

15. Pillai K. and V.S.S. Nair., *A Model for Software Development effort and Cost Estimation*, IEEE Transactions on Software Engineering; vol. 23(8), pp. 485-497, **1997**.

16. Pham, H., Nordmann, L. and Zang, X., "*A General Imperfect Software Debugging Model with S-Shaped Fault Detection Rate*", *IEEE Trans. Rel*., Vol. 48, pp 169-175, **1999**.

17. Shyur, H. J., *A Stochastic Software Reliability Model with Imperfect-Debugging and Change-Point,* Journal of Systems and Software, 66, pp. 135-141, **2003**.

18. Wang Z., J. Wang, *Parameter Estimation of Some NHPP Software Reliability Models with Change-Point,* Communications in Statistics- Simulation and Computation, 34, pp. 121-134, **2005**.

19. Zang, X., Teng, X. and Pham, H., "*Considering Fault Removal Efficiency in Software Reliability Assessment*", IEEE Trans. on Systems, Man and Cybernetics-Part A: Systems and Humans, Vol. 33, (1), 114-120, **2003**.

20. Zhao, M., *Change-Point Problems in Software and Hardware Reliability,* Communications in Statistics-Theory and Methods, 22(3), pp. 757-768, **1993**.