Discrete Software Reliability Growth Models with Discrete Test Effort Functions

Sk.MD.Rafi [1] and Shaheda Akthar[2]

[1]Assoc.Professor, Dept.of CSE , SMITW, Affiliated to JNTU,Kakinada
Mail: mdrafi.527@gmail.com

[2]Professor, Dept.of CSE, SMCE, Affiliated to JNTU,Kakinada
Mail:shaheda.akthar@yahoo.com

Abstract

Number of software Reliability growth models has been proposed in the literature. A mathematical technique which describes the software testing phenomenon known as the software reliability growth model. Software reliability growth models are used to predict the number of faults and reliability of the software. In the view of this software reliability growth models are basically differentiated as the continuous and discrete models. There is a plenty of development in the continuous models but little towards the discrete models. In this paper we have presented a discrete reliability growth model with different discrete testing effort functions and the same time software release policy is discussed. A new imperfect debugging discrete software reliability growth model with testing effort is proposed. All calculations are done on real data. The results shows the proposed testing effort models are perfectly fit to the data.

Keywords

Software Reliability, NHPP, Testing effort, Discrete SRGM.

I.INTRODUCTION

Software plays an important role in every body's life. From the general usages to the heavy equipment needs the software. As the software usages increase every body needs a zero defect software. Software with zero defects treated as the quality and reliable software. But software is developed rather than manufactured like classic sense. During software development software is going through phases like requirement, design and code there is chance that error might propagate in to the software. Even though we have sophisticated techniques in identifying the errors ,but complexity of the software make errors to escape. Several quality control and quality improvement techniques has been developed in early phases of software life cycle intended in improving the software.

Although the efficiency, performance and reliability are external functional requirements, but they play an important role in the quality of the software. Software reliability is defined as how long a software will function well before it struck with a fault. Many companies are spending enormous amount of cost during development to achieve the good quality software product. Reliability and quality are two important factors associated with software. People used to spend much time and effort during the testing to find the errors. Software reliability growth models [5][9] which were described as mathematical formulation of complex expressions which describes the real time testing environment. These SRGM provides the mathematical relation between time span and cumulative faults which are discovered during the testing. Several reliability growth models are proposed in software literature. These reliability growth models help in understanding the real testing environment. These reliability growth models are categorized as how they analyze the software failure data. Failure data can analyzed by the count

DOI : 10.5121/ijsea.2011.2402

process models which considers the failure count data for reliability estimation and time interval process models which requires the time interval data. Generally these software reliability growth models are divided as two groups one based on calendar/execution time period and another based on the number of test cases used. Goel and Okumoto [5][6] proposed a exponential SRGM, which is characterized by the time and cumulative number of failures. Yamada [10] has proposed a NHPP S shaped software reliability growth model, called as delayed S shaped model and Inflection S shaped model. If the models considers time span for their model they are called as the continuous execution time models and where as if they consider number of test cases to characterize their data is called as Discrete time models [1][8][9].

In discrete time models [1][8][9] software fault detection period is countable. A test case is defined as single computer test run executed in an hour, intended to know its behavior. These test cases can be used in an hour, day, week or even month. Most SRGM are charactrised by the mean value function of non homogeneous Poisson process and uses the past failure data collected during testing phase to predict the quality of the software. Recent days new SRGM of non homogeneous Poisson process(NHPP) models which describes the testing strategy with effort used during testing. Testing effort during software testing better described by number of test cases used, person/months and time. Yamada proposed a SRGM with testing effort described by exponential, rayleigh and weibull curve. In the same way Huang proposed logistic curve as to define testing effort during the testing. Kapur proposed a model in which the testing phase is assumed to have two different process namely, fault isolation and fault detection process. Musa developed a scheme for classifying existing SRGM's and demonstrated the execution time is better time domain for software reliability modeling than calendar time.

Generally software testing is not always perfect in nature, testing is influenced by several factors. Due to the complexity nature of faults during testing faults are not completely removed. There is a chance that one fault may influence another one, which cause for additional faults in the software. The concept of imperfect debugging is introduced by Goel. He observed the imperfect debugging in jelinski and Moranda model. Some models observe imperfect debugging with testing effort.

In this paper we proposed perfect and imperfect debugging testing effort dependent discrete software reliability growth model. The testing effort is described by discrete exponential, discrete logistic and discrete gompertz curve. Assuming that discrete failure intensity proportional to the faults remain in the software and proportionality based on current testing effort expenditure at arbitrary test case. Also we have analyzed software release time based on cost and software intensity.

2.TESTING EFFORT FUNCTIONS

Testing effort is described as amount of testing expenditure is spend during the testing.

A) Discrete exponential curve: let W(n) be denote the expected cumulative number of faults detected up to nth testing-period. Discrete analog exponential curve as , b represents rate at which testing effort is consumed. 'a' represents the initial total test effort before the testing begins.

$$W(n+1)-W(n)=\delta\times b\times(a-W(n)) \qquad (1)$$

Above difference equation is solved using probability generating function.

$$W(n)=a\times[1-(1-\delta\times b)^{n}] \qquad (2)$$

By using the property $\lim_{x->\infty}(1+x)^{1/x}=e$ and $t=\delta\times n$ we get the above equation (2) as $a\times(1-e^{-b\times t})$ in continuous model.

B) Discrete Gompertz TEF : : let G(n) be denote the expected cumulative number of faults detected up to nth testing-period. Gompertz Curve model is one of the S- shaped growth model. Discrete analog Gompertz TEF is given by

$$G(n+1) = G(n) \times \left(\frac{G(n)}{k} \right)^{\delta \times \log(b)} \quad (3)$$

Solving the above equation with first order linear difference equation we get exact solution as

$$G(n) = k \times a^{(1+\delta \log(b))^n} \quad (4)$$

C) Discrete Logistic TEF : let L(n) be denote the expected cumulative number of faults detected up to nth testing-period. Logistic Curve model is one of the S- shaped growth model. Discrete analog Logistic TEF is given by

$$L(n+1) - L(n) = \frac{(\delta \times \alpha)}{k} \times L(n+1)(k - L(n)) \quad (5)$$

Solving the above equation we get the exact solution as

$$L(n) = \frac{k}{1 + m \times (1 - \delta \times \alpha)^n} \quad (6)$$

Above equation when applied property $\lim_{x->\infty} (1+x)^{1/x} = e$ and $t = \delta \times n$ we get $\frac{k}{1 + m \times e^{-\alpha \times t}}$ logistic curve in continuous form. The parameter m is defined as

$$L(0) = \frac{k}{1+m} \quad (7)$$

3. PARAMETER ESTIMATION OF TESTING EFFORT FUNCTIONS

A) Discrete Exponential TEF:

The parameter estimates of $\hat{a}$ and $\hat{b}$ which are estimated values of a and b can be obtained by the following procedure of method of least square. The equivalent regression equation for Eq (2) is

$$Y(n) = A + B \times W(n) \quad (8)$$

Where

$$Y(n) = W(n+1) - W(n) \quad (9)$$

$A = \delta ab$ and $B = -\delta b$

Based on the regression analysis we estimate the values of A and B and in turn we can estimate a and b values as

$$a = -\frac{A}{B} \quad and \quad b = -\frac{B}{\delta} \quad (10)$$

B) Discrete Gompertz TEF:

The parameters of Discrete Gompertz TEF such as a ,b and k is obtained as simple linear regression equation. At first the Eq.(3) is converted in to linear passion. Now take the log on both side of the Eq.(3) to get

$$\begin{aligned} &\log G(n+1) - \log G(n) = \\ &-(\delta \log b)(\log k) + (\delta \log b) \log G(n) \end{aligned} \quad (11)$$

From this above equation we get

$$Y(n) = A + B \log G(n) \quad (12)$$

Where

$$Y(n) = \log G(n+1) - \log G(n) \qquad (13)$$

C) Discrete Logistic equation:

The parameters of discrete logistic TEF such as k, m and α is obtained as simple linear equation. The equation (5) is converted as linear passion. We take the following the regression equation

$$Y(n) = A + B \times L(n+1) \qquad (14)$$

Where

$$t_n = n \times \delta \qquad (15)$$

$$Y_n = \frac{L_{n+1}}{L_n} \qquad (16)$$

Here , δ is a constant difference interval.And

$$A = \frac{1}{(1 - \delta \times \alpha)} \qquad (17)$$

$$B = \frac{-\delta \times \alpha}{k \times (1 - \delta \times \alpha)} \qquad (18)$$

Now from above

$$\hat{k} = \frac{1 - \hat{A}}{\hat{B}} \qquad (19)$$

$$\delta \times \alpha = 1 - \frac{1}{\hat{A}} \qquad (20)$$

$$\hat{m} = \frac{\sum_{n=1}^{N} (\hat{k} - L_n)}{\sum_{n=1}^{N} (L_n \times (1 - \delta \times \alpha)^n)} \qquad (21)$$

Where $\hat{A}$ and $\hat{B}$ are the estimates of parameters A and B, respectively.

4. MODELING SOFTWARE RELIABILITY GROWTH MODEL WITH DISCRETE TEF

4.1 Discrete software reliability growth model with Discrete TEF based on NHPP.

The following assumptions are made for above reliability growth model

a) Software is subjected to failure at random test runs caused by the error remaining in the software.

b) When a failure occurs, the error causing that failure is immediately removed , no new faults are introduced.

c) Testing effort expenditure are described by discrete exponential, discrete logistic and discrete gompertz curve.

d) The expected discrete failure intensity to the current testing effort expenditure is proportional to the current remaining error content.

e) The fault removal process follows the NHPP.

$$\frac{m(n+1) - m(n)}{w(n)} = b \times (a - m(n)) \qquad (22)$$

Solving the abve differential equation under the initial condition m(0)=0, we get

$$m(n) = a(1 - \prod_{i=0}^{n} (1 - b \times w(i))) \qquad (23)$$

The failure intensity at nth test case is given by

$$\lambda(n) = a \times b \times w(n) \times \prod_{i=0}^{n-1} (1 - b \times w(i)) \qquad (24)$$

4.2 Imperfect debugging discrete software reliability growth model with discrete TEF based on NHPP

The following are the assumptions for the above models

a) the fault removal process follow the NHPP.
b) The software is subject to failure random times caused by faults remaining in the system
c) The mean number of faults detected by the current testing effort is proportional to the number of remaining faults in the system.
d) The proportionality of fault detection is constant.
e) The consumption of testing effort is modeled by discrete logistic, discrete gompertz and discrete exponential curve.
f) When a fault is detected & removed new faults may be generated. When removing or fixing a detected fault , the probability of introducing another fault is a constant β.

Based on above assumption the discrete SRGM with discrete testing effort is described by

$$\frac{m(n+1) - m(n)}{w(n)} = \delta \times b \times (a + \beta \times m(n) - m(n)) \qquad (25)$$

Solving the above equation under the condition m(0)=0 we the get

$$m(n) = \frac{a}{(1-\beta)} \times (1 - \prod_{i=0}^{n-1} (1 - w(i) \times \delta \times b \times (1-\beta))) \qquad (26)$$

Next imperfect debugging model based on following

$$\frac{m(n+1) - m(n)}{w(n)} = b \times (a \times p - m(n)) \qquad (27)$$

Solve the above difference equation under the condition that m(0)=0. we get

$$m(n) = a \times p \times (1 - \prod_{i=0}^{n} (1 - b \times w(i))) \qquad (28)$$

Next imperfect debugging model based on the following equation

$$\frac{m(n+1) - m(n)}{w(n)} = b \times (a - p \times m(n)) \qquad (29)$$

Now above equation is solved under the condition m(0)=0. we get

$$m(n) = \frac{a}{p} \times (1 - \prod_{i=0}^{n} (1 - b \times p \times w(i))) \qquad (30)$$

5. PARAMETER ESTIMATION AND NUMERICAL ILLUSTRATION

MLE is method is used for the parameter estimation for proposed models. All data sets are used in the form of (n_i,x_i, y_i) where i=1,2,3 …..N. where n_i is test run number at which the x_i is the cumulative testing time used to find y_i number of cumulative faults detected during testing. The likelihood function for unknown parameters with mean value function is given by

$$L(parameters\,of\ m(n)) = \prod_{i=1}^{N} \frac{[m(n_i) - m(n_{i-1})]^{y_i - y_{i-1}}}{(y_i - y_{i-1})} \times \exp(-(m(n_i) - m(n_{i-1}))) \tag{31}$$

Taking natural logarithm of above equation we get

$$\ln L = \sum_{i=1}^{N} (y_i - y_{i-1}) \times \ln(m(n_i) - m(n_{i-1})) - \{m(n_i) - m(n_{i-1})\} - \sum_{i=1}^{N} \ln(y_i - y_{i-1})! \tag{32}$$

The MLE of the SRGM parameters can be obtained by maximizing L in the equation (32). Form the equation (25) and (32) we get

$$\ln L = \sum_{i=1}^{N} (y_i - y_{i-1}) \times [\ln(a) + \ln(b) + \ln(w(i)) + \ln(\prod_{i=0}^{N} (1 - b \times w(i)))] - a \times (1 - \prod_{i=0}^{N} (1 - b \times w(i)))] \tag{33}$$

Now

$$\frac{\partial \ln L}{\partial a} = \sum_{i=1}^{N} (y_i - y_{i-1}) \times \frac{1}{a} - (1 - \prod_{i=0}^{N} (1 - bw(i))) = 0 \tag{34}$$

$$a = \frac{y_N}{(1 - \prod_{i=0}^{N} (1 - bw(i)))} \tag{35}$$

Same for parameter b as

$$\frac{\partial \ln L}{\partial b} = \frac{y_N}{b} - [\sum_{i=1}^{N} (y_i - y_{i-1}) \times \sum_{j=0}^{i-1} \frac{w(j)}{(1 - bw(j))} - [a \times \prod_{i=1}^{N} (1 - bw(i)) \times \sum_{i=1}^{N} \frac{w(i)}{(1 - bw(i))}] \tag{36}$$

Here we have selected two data sets to present our evaluation of proposed models. Data set one taken from Ohba where a total of 47.65 CPU hours has been spend during testing to get a total of 328 faults were discovered by spending 19 weeks. The following diagram depicts the actual testing effort spend during testing to the estimated testing effort with our proposed discrete testing effort functions.

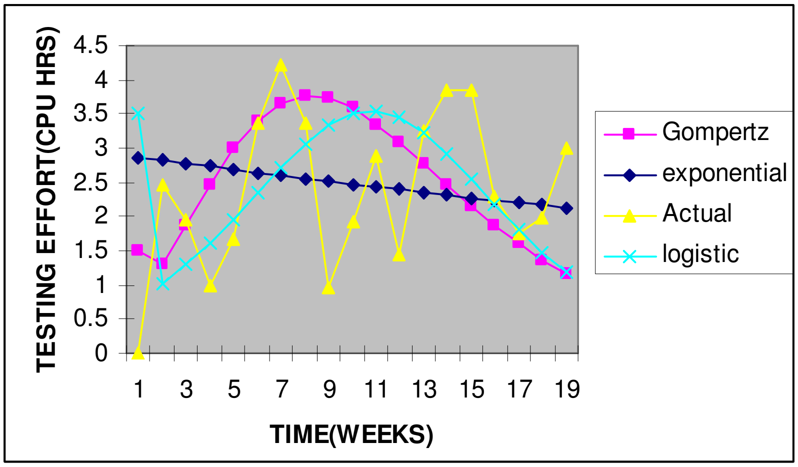


Fig1: Actual/Estimated logistic, gompertz and exponential TEF for DS1

The estimated parameters of discrete exponential TEF from equation(2) is a=175 and b=0.01252 and δ=1.3 , from equation (4) the estimated values of discrete gompertz TEF are k=54, a=0.0131 and b=0.841 and the estimated parameters for discrete logistic TEF are k=50.78 and m=17.82 from the equation (6). The following table gives the predictive value of the testing effort respect to the actual testing effort.

Table-1

Comparison result for different TEF for DS1

TEF Curve	Bias	Variation	MRE	RMS-PE
Exponential	-1.806	1.55	0.01	0.925
Gompertz	-2.098	2.83	0.007	1.89
Logistic	-0.45	1.84	0.02	1.78

The following table depicts the values of parameters of mean value functions. All parameters are calculated based on MLE from the equation 36 and 37.

Table-II

Estimated Parameter values and model comparison for DS1

Model	a	r	MSE	Noise
SRGM with Discrete Exponential TEF	470	0.0251	207	-0.324
SRGM with discrete Gompertz TEF	581	0.0171	218	0.361
SRGM with discrete Logistic TEF	606	0.0161	222	0.217
Yamada delayed S shaped model	384	0.0219	640	2.33

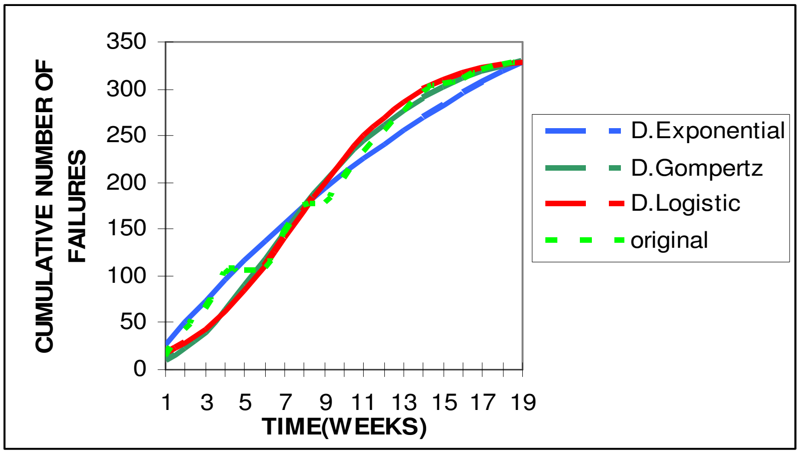


Fig 2. Cumulative number of failures for discrete Logistic, Gompertz and Exponential TEF based SRGM

6. OPTIMAL SOFTWARE RELEASE POLICY

It is very important for every software company to know how much of time needed to spend on the software and resources required to spend on the software[4][6]. The software release problem [7] has a great importance in the software literature. Several people had their own theory on software release concept. In this paper we formalize the software release concept in terms of cost spend during operational, cost spend for unit testing effort and cost during the maintenance activity. We obtained the software release time based on the cumulative number of failures, by minimizing the total cost and maximizing the reliability [2][8][10]. Mathematically we formulate this problem as

C_1= the cost of fixing error during the testing phase.
C_2= the cost of fixing error during the operational phase where $C_2>0$.
C_3= the cost per unit test expenditure.
minimize $C(n)=C_1*m(n)+C_2*(m(n_{LC})-m(n))+C_3*W(n)$. (37)
subjected to λ (n)= λ_0 from (38) we observe that

$$C(n)-C(n-1)=C_1*(m(n)-m(n-1))+ \\ C_2*(m(n)-m(n-1))+C_3*w(n) \quad (38)$$

$$=C_3*w(n)-\lambda(n)*(C_2-C_1) \quad (39)$$

$$=w(n)*[C_3-(C_2-C_1)*a*b*\prod_{i=1}^{n-1}(1-b*w(i))] \quad (40)$$

So, if $ab>\frac{C_3}{C_2-C_1}$ and there exist n=n_1 such that

$$ab\prod_{i=0}^{n_1-1}(1-bw(i))>\frac{C_3}{C_2-C_1}>ab\prod_{i=0}^{n_1}(1-bw(i))$$

C(n) is minimum at n=n_1 , while if

$$ab\prod_{i=0}^{n_1-1}(1-bw(i))>\frac{C_3}{C_2-C_1}>ab\prod_{i=0}^{n_1}(1-bw(i))$$

C(n) is minimum at n=n_1 and n_1+1

If $ab<\frac{C_3}{C_2-C_1}$, C(n) is minimum at n=0 while if

If $ab=\frac{C_3}{C_2-C_1}$ is minimum at both n=0 and 1.

Now if
$$\lambda(n)-\lambda(n-1)= \\ abw(n)\prod_{i=1}^{n-1}(1-bw(i))-abw(n-1)\prod_{i=1}^{n-2}(1-bw(i)) \quad (41)$$

From above

$$ab\prod_{i=1}^{n-2}(1-bw(i))[w(n)(1-bw(n-1))-w(n-1)] \quad (42)$$

Since $\frac{w(i)}{w(i-1)}>\frac{w(i+1)}{w(i)}$ for all i > 0, λ(n) is monotonically increasing in (0,n_x) and decreasing thereafter where n satisfies

$$\frac{1}{w(n_x-1)}-\frac{1}{w(n_x)}\geq b\,and\,\frac{1}{w(n_x)}-\frac{1}{w(n_x+1)}<b \quad (43)$$

If $\lambda(n_x)>\lambda_0$ and there exist n_2 and n_3 ($0<n_2<n_x<n_3$)such that
$\lambda(n_2)\leq\lambda_0$ and $\lambda(n_2+1)>\lambda_0$ (44)
$\lambda(n_3)\leq\lambda_0$ and $\lambda(n_3-1)>\lambda_0$ (45)

the reliability of the system is given by

$$R(n)=\exp[-(m(n+1)-m(n))] \quad (46)$$

From above mean value function the reliability is given by

$$R(n)=e^{-[abw(n)\prod_{i=0}^{n}(1-bw(i))]} \quad (47)$$

From (25) we get

$$R(n) = e^{-[\lambda(n+1)]} \tag{48}$$

Then the desired intensity is achieved for all $n \le n_2$ and $n \ge n_3$, while if $\lambda(n_x) \le \lambda_0$ the desired failure intensity is achieved.

Combining the cost and reliability requirements

(i) if $ab\prod_{i=0}^{n_1-1}(1-bw(i)) > \dfrac{C_3}{C_2-C_1} > ab\prod_{i=0}^{n_1}(1-bw(i))$ (49)

$\lambda(n_x)>\lambda_0$ and there exist n_2 and n_3 satisfying the equation (45) and (46) then
if $n_1<n_2$ or $n_1>n_3$, $n^*=n_1$, if $n_2<n_1<n_3$ then if
$C(n_2)<C(n_3)$, $n^*=n_2$.
$C(n_2)>C(n_3)$, $n^*=n_3$.
$C(n_2)=C(n_3)$, $n^*=n_3$ or $n^*=n_2$.
If $\lambda(n_x)\le\lambda_0$, $n^*=n_1$.

(ii) if $ab\prod_{i=0}^{n_1-1}(1-bw(i)) > \dfrac{C_3}{C_2-C_1} = ab\prod_{i=0}^{n_1}(1-bw(i))$ (50)

$\lambda(n_x)>\lambda_0$ and there exist n_2 and n_3 satisfying the equation (45) and (46) then
if $n_1<n_2$ or $n_1>n_3$, $n^*=n_1$, if $n_2<n_1<n_3$ then if
$C(n_2)<C(n_3)$, $n^*=n_2$.
$C(n_2)>C(n_3)$, $n^*=n_3$.
$C(n_2)=C(n_3)$, $n^*=n_3$ or $n^*=n_2$.
If there exist n_3 satisfying (46) then
If $n_1 \ge n_3-1$, $n^*=n_1+1$ else $n^*=n_3$
Else if $\lambda(n_x) \le \lambda_0$ and $n_1 < n_x$, $n^*=n_1$
Else if $\lambda(n_x) \le \lambda_0$ and $n_1 \ge n_x$, $n^*=n_1+1$.

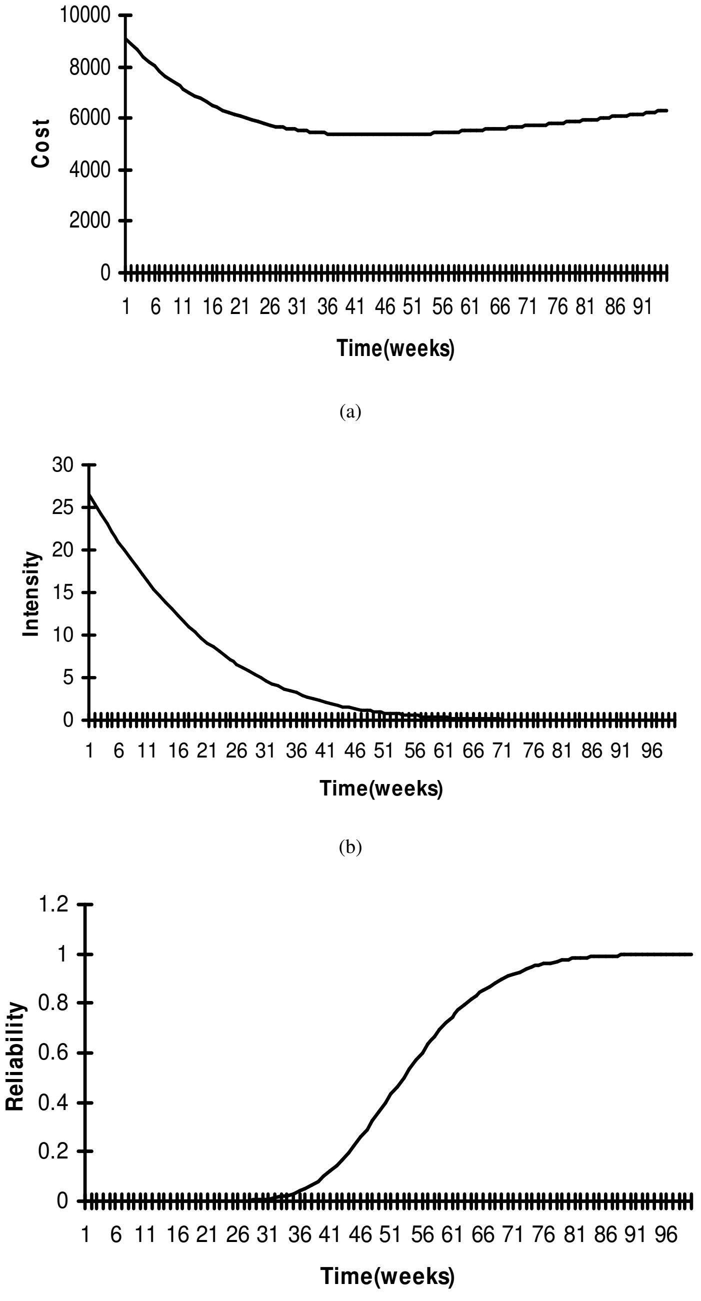
(iii) if $ab \le \dfrac{C_3}{C_2-C_1}$, $\lambda(n_x)>\lambda_0$ and there exist n_2 and n_3 satisfying (45) and (46) then

$n^*=n_2$ else if there exist n_3 satisfying (46) then $n^*=n_3$, while if $ab \le \dfrac{C_3}{C_2-C_1}$, $\lambda(n_x)\le\lambda_0$,$n^*=1$.

Numerical illustration : in this we discussed the optimal release policy based on cost , intensity and reliability. By using the first data set, the estimated parameters of discrete exponential TEF from equation(2) is $\alpha=175$ and $b=0.01252$ and $\delta=1.3$, $a=470$ and $r=0.0251$. we assume $C_1=10$, $C_2= 20$ and $C_3=4$, $n_{LC}=100$ and $\lambda_0 =0.8$. using assumed values we get $n_1=39$ and $n_3=55$, $C(n_1)=5373.84$ and $C(n_3)=5425.88$,$\lambda(n1)=2.494$ and $\lambda(n_3)=0.561$. using the theorem we get low intensity at $n^*=55$ and high reliability $R(55)=0.57$. but only cost has to be minimized the $n^*=39$ but failure intensity is high and reliability is minimum $R(39)=0.08$.

7.CONCLUSION

In this paper , we have developed a discrete SRGM with discrete exponential , discrete gompertz and discrete logistic TEF curve. At the same time we have developed discrete imperfect debugging SRGM with discrete TEF. We have developed optimal release policy based on cost, reliability and intensity requirements.



(a)

(b)

(c)

Fig 3: (a) Total cost (b) Intensity (c) Reliability

REFERENCES

[1] Brooks W.D. and Motley R.W., Analysis of discrete software reliability models, *Technical Report,*RADC-TR-80-84, *Rome Air Develop,* 1980, New York.

[2] Kapur P.K. and Garg R.B., Cost-reliability optimums release policies for a software system with testing effort, OPERATION RESEARCH. 27(2), 1990, pp 100 - 114.

[3] "An Introduction to difference equations " third edition by Saber Elaydi Springer series.

[4] Koch H.S. and Kubat P., Optimal release time for computer software, IEEE *Transaction on software Eng ,vol-SE-9,1983,pp323-327.*

[5] Musa J.D. et al, Software reliability: measurement, prediction and application, *McCraw-Hill,* 1987, New York.

[6] Okumoto K. and Goel A.L., Optimal release time for software systems based on reliability and cost criteria, *J.* Systems *and software* , 1980, 1, pp 315 -318.

[7] Ross S.M., Software reliability: the stopping rule problem, IEEE Tram *oti Software Eng,* SE – 11, 1985, pp 1472-1476.

[8] Kapur, P.K.; Min Xie; Garg, R.B.; Jha, A.K, "A discrete software reliability growth model with testing effort " IEEE conference ,1994, pp16-20.

[9] Yamada S.and Osaki S., Discrete software reliability growth *models, Applied stochastic models and data analysis, 1985, 1, pp 65-77.*

[10] Yamada **S.** and Osaki S., Optimal software release policies with simultaneous **cost** and reliability requirements, Euro J. Operation *Research 31, 1987, pp46-51.*

Sk.MD.Rafi received B.Tech (comp) from Jawaharlal Nehru Technological University, M.Tech (comp) from Acharya Nagarjuna University. Pursuing PhD from Jawaharlal Nehru Technological University. Presently working as Associate. Professor in Sri Mittapalli Institute of Technology for women, affiliated to J.N.T.U, Kakinada. My area of interest is Software Reliability, Software Architecture Recovery, Network Security, and Software Engineering. Published So many research papers in various International Journals.

Shaheda Akthar received Bachelor of computer science and Master of Computer Science from Acharya Nagarjuna University, M.S (Software Systems) from BITS, Pilani, received PhD(Comp) from Acharya Nagarjuna University. Presently working as .Professor in Sri Mittapalli College of engineering, affiliated to J.N.T.U, Kakinada. My area of interest is Software Reliability, Software Architecture Recovery, Network Security, and Software Engineering. Published So many research papers in various International Journals.