Analysis of 4p-Kappa TEF in to Software Reliability Growth Model and Optimal Software Release Policy

Sk.Md.Rafi [1] , Dr. K.Nageswara Rao[2] , Dr.K.Pallam Sety[3], Shaheda Akthar[4]
[1].*Department of Computer science and Engineering, Sri Mittapalli college of Engineering ,A.P,INDIA.*
[2].*Department of Computer science and Engineering, PVPSIT college of Engineering ,A.P. INDIA.*
[3].*Department of Computer science and Engineering,, Andhra university , Vishakhapatnam, A.P.INDIA*
[4].*Department of Computer science and Engineering, Sri Mittapalli college of Engineering, A.P. INDIA.*

Abstract
This paper investigates a SRGM (Software reliability growth model) based on NHPP (non-homogeneous Poisson process) which incorporates the 4p-Kappa testing effort function. Testing Software reliability is generally a key factor in software quality. Reliability is an essential ingredient in customer satisfaction. In software development process reliability conveys the information to managers to access the amount of testing effort and time at which software release into the market. Many papers are published in this context. Performance application of proposed model is demonstrated through real datasets. The experimental results shown that the model gives an excellent performance compared to other models. We also discuss the optimal release time based on reliability requirement and cost criteria.
Keywords
Non-homogeneous Poisson process, Mean value function, Optimal software release time, Software reliability growth model, Testing-effort function.

ACRONYMS
NHPP : Non Homogeneous Poisson Process
SRGM : Software Reliability Growth Model
MVF : Mean Value Function
MLE : Maximum Likelihood Estimation
TEF : Testing Effort Function
LOC : Lines of Code
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_d (t) : Cumulative number of faults detected up to t.
m_r (t) : Cumulative number of faults isolated up to t.
W (t) : Cumulative testing effort consumption at time t.
W*(t) : W (t)-W (0)
A : Expected number of initial faults
r (t) : Failure detection rate function
r : Constant fault detection rate function.
r_1 : Constant fault detection rate in the Delayed S-shaped model with 4p-kappa TEF
r_2 : Constant fault isolated rate in the Delayed S-shaped model with 4p-kappa TEF

1. Introduction

Software plays an important role in every body's life. The role of software is increasing rapidly in the fields that may be engineering, medical or business industries. Correctness and required performance are key factors for the software to be successful. Reliability is one of the key factors in accessing the quality of the software. In past many papers are published in accessing the software quality through reliability. The main objective of software industry is to prepare software which is much reliable and satisfy the customer needs. The testing phase is an important and expensive part during the software development process. Software reliability represents a customer oriented view of software quality. 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 [Hoang Pham]. 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. Many papers are published based on TEF in NHPP models by [Yamada 1986, Bokhari 2006, Kapur 1994 and Haung 1997]. All of them describe the tracking phenomenon with test expenditure. We have investigated several datasets observed that no testing effort fully fits for all the datasets. For that we used a 4p-Kappa testing effort that incorporated several testing efforts in it.
This paper describes the time dependent behavior of testing–effort by a 4p-Kappa curve. Assuming that the error detection rate in software testing is proportional to

Manuscript received February 5, 2010
Manuscript revised February 20, 2010

the current error content and the proportionality depends on the current test effort, flexible software.
Reliability growth model based on non-homogeneous Poisson process is developed and its applications are presented. Further an optimal release time is calculated based on reliability and cost. Section-2 proposed the test-effort function described by 4p Kappa curve. In Section – 3 a software reliability growth model with 4p Kappa testing- effort function is discussed. Section -4 contains a model evaluation criterion. Section -5 includes model performance analysis. Section-6 presents the prediction of optimal release time based on the application of the model to software reliability management.

4p-Kappa curve TEF

The 4p Kappa distribution was introduced by Hosking 1994 is a very general distribution which includes a variety of distributions as special cases. .The Kappa distribution with shape and scale parameters had a great flexibility in accommodating all the forms of the hazard rate function, can be used in a variety of problems for modeling software failure data. Another important characteristic of the distribution is that it contains a special sub-models, the Pareto (Kumar, Ahmad and Quadri 2005), Generalized logistic (Huang and Lyu)

2. Current cumulative Testing effort

$$W(t)=a\times\left(1-h\times\left(1-k\times\frac{(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}} \qquad (1)$$

Where a>0,α>0,ξ>0 at t>0 where a is the total effort expenditure , α controls the scale of the distribution, k and h are shape parameters, ξ is location parameter

$$\frac{d}{dt}W(t)=-\frac{a\left(1-h\left(-\frac{-\alpha+kt-k\xi}{\alpha}\right)^{\frac{1}{k}}\right)^{-\frac{-1+h}{h}}\left(-\frac{-\alpha+kt-k\xi}{\alpha}\right)^{\frac{1}{k}}}{-\alpha+kt-k\xi} \qquad (2)$$

Following are the some of the special cases;

1) The generalized Pareto distribution

$$W(t)=a\left(1-\left(1-\frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)$$

a>0,α>0,ξ>0,h=1 at t>0 (3)

2) the generalized extreme value distribution at

$$W(t)=e^{-\left(1+k\times\frac{(t-\xi)}{\alpha}\right)^{-\frac{1}{k}}}$$

a>0,α>0,ξ>0,h=0 at t>0 (4)

3) The generalized Logistic distribution

$$W(t)=\frac{a}{1+\left(1-\frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}}}$$

a>0,α>0,ξ>0,h=-1 at t>0 (5)

4) The Gumbel distribution

$$W(t)=a\times\left(1-e^{-e^{\left(\frac{(t-\xi)}{\alpha}\right)}}\right)$$

a>0,α>0,ξ>0,h=1 and k=1 at t>0 (6)

5) the exponential distribution

$$W(t)=a\times\left(1-e^{-\left(\frac{(x-\xi)}{\alpha}\right)}\right)$$

a>0,α>0,ξ>0,h=1 and k=0 at t>0 (7)

6) the uniform distribution

$$W(t)=\frac{a\,(t-\xi)}{\alpha}$$

a>0,α>0,ξ>0,h=1 and k=1 at t>0 (8)

And also other related distributions
The testing effort reaches its maximum value at

$$T_{\max}=\frac{-\left(e^{\ln\left(\frac{-1+k}{-1+k\times h}\right)\times k}\times\alpha-\alpha-k\times\xi\right)}{k} \quad \text{at } t>0$$

(9)

3. Software Reliability growth model and testing effort functions

3.1 SRGM with 4p Kappa Testing-effort function

The following assumptions are made for software reliability growth modeling (Yamada and Osaki 1985 Yamada 1986, 1993, Kapur 1999, Kuo 2001 Haung and Kuo 2002, Haung 2005)

(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) Consumption curve of testing effort is modeled by a 4p Kappa TEF.
(vi) Each time a failure occurs, the fault that caused it is immediately removed and no new faults are introduced.
(vii)We can describe the mathematical expression of a testing-effort based on following

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = r \times (A - m(t)) \qquad (10)$$

$$m(t) = A \times \left(1 - e^{-r \times (W(t) - W(0))}\right) \qquad (11)$$

substituting W(t) from eq.(1), we get

$$m(t) = A \times \left(1 - e^{-r \times \left(a \times \left(1 - h \times \left(1 - k \times \frac{(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}} - a\left(1 - h\left(1 + \frac{k\xi}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}}\right)}\right) \qquad (12)$$

This is an NHPP model with mean value function with the GMW testing-effort expenditure.
Now failure intensity is given by

$$\lambda(t) = \frac{dm(t)}{dt} = A \times r \times w(t) \times e^{-r \times W*(t)}$$

$$\frac{d}{dt} m(t) = \left(A\, r\, a \left(1 - h\left(1 - \frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}} \left(1 - \frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}} e^{-r\left(a\left(1 - h\left(1 - \frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}} - a\left(1 - h\left(1 + \frac{k\xi}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}}\right)}\right) \Bigg/ \left(\alpha\left(1 - \frac{k\,(t-\xi)}{\alpha}\right)\left(1 - h\left(1 - \frac{k\,(t-\xi)}{\alpha}\right)^{\frac{1}{k}}\right)\right) \qquad (13)$$

3.2 Yamada Delayed S-shaped model with 4p Kappa testing-effort function

The delayed 'S' shaped model originally proposed by Yamada [Yamada] and it is different from NHPP by considering that software testing is not only for error detection but error isolation. And the cumulative errors detected follow the S-shaped curve. This behavior is indeed initial phase testers are familiar with type of errors and residual faults become more difficult to uncover [Goel 1985, M.Ohba 1984, M.R.Lyu 1996].
From the above steps described section 3.1, we will get a relationship between m(t) and w(t).
The extended S-shaped model [Yamada 1983] is modeled by

$$\frac{dm_d(t)}{dt} \times \frac{1}{w(t)} = r_1 \times \left[a - m_d(t)\right] \qquad (14)$$

and

$$\frac{dm_r(t)}{dt} \times \frac{1}{w(t)} = r_2 \times \left[a - m_r(t)\right] \qquad (15)$$

We assume $r_2 \neq r_1$ by solving 14 and 15 boundry conditions $m_d(t)=0$, we have

$$m_d(t) = a \times \left(1 - e^{\left[-r_1 \times W*(t)\right]}\right) \qquad (16)$$

and

$$m_r(t) = a \times \left[1 - \frac{\left(r_1 \times e^{\left[-r_2 \times W*(t)\right]} - r_2 \times e^{\left[-r_1 \times W*(t)\right]}\right)}{r1 - r2}\right] \qquad (17)$$

At this stage we assume $r_2 \approx r_1 \approx r$, then using 'L' Hospitals rule the Delayed S-shaped model with TEF is given by

$$m(t) \cong m_r(t) = a \times \left(1 - (1 + r \times W*(t)) \times e^{\left[-r \times W*(t)\right]}\right) \qquad (18)$$

The failure intensity function for Delayed S-shaped model with TEF is given by

$$\lambda(t) = a \times r^2 \times w(t) \times W*(t) \times e^{[-r \times W*(t)]} \qquad (19)$$

4. Evaluation Criteria

a) The goodness of fit technique

Here we used MSE [M.Xie 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

$$MSE = \sum_{i=1}^{k} \frac{\left[m(t_i) - m_i\right]^2}{k} \qquad (20)$$

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

b) Coefficient of multiple determinations (R^2) which measures the percentage of total variation about 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 provies the best fit to the data. The best model is that which proves higher R^2. that is closer to 1.

c) The predictive Validity Criterion

The capability of the model to predict failure behavior from present & past failure behavior is called predictive validity. This approach, which was proposed by (J.Dmusa 1987], can be represented by computing RE for a data set

$$RE = \frac{(m(t_q) - q)}{q} \quad (21)$$

In order to check the performance of the Generalized Modified Weibull testing _effort and make a comparison criteria for our evaluations [M.Shepperd and C.Schofield 1997,K. Srinivasan and D.Fisher1995].

d) SSE criteria: SSE can be calculated as :[Hoang Pham 2000]

$$SSE = \sum_{i=1}^{n} [y_i - m(t_i)]^2 \quad (22)$$

Where y_i is 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 for i=1,2,…..,n.

$$PE_i = Actual(observed)_i - Predicted(estimated)_i \quad (23)$$

$$Bias = \sum_{i=1}^{n} \frac{PE_i}{n} \quad (24)$$

$$Variation = \sqrt{\sum_{i=1}^{n} \frac{(PE_i - Bias)^2}{n-1}} \quad (25)$$

$$MRE = \frac{|M_{estimated} - M_{actual}|}{M_{actual}} \quad (26)$$

5. Model Performance Analysis

DS1: the first set of actual data is from the study by Ohba(1984).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. Fitting the model to the actual data means by estimating the model parameter from actual failure data. Here we used the LSE (non-linear least square estimation) Mat-Lab program to estimate the parameters. Calculations are given in appendix A. All parameters of other distribution are estimated through LSE. The unknown parameters of 4p-Kappa are a=65(CPU hours), h=0.65, k=0.6214, α=14.98, ξ=5.85. Correspondingly the estimated parameters of Logistic TEF are N=54.84(CPU hours), A=13.03 and b=0.2263/week and Rayleigh TEF N=49.32 and b=0.00684/week. Fig.1 plots the comparison between observed failure data and the data estimated by 4p-Kappa TEF, Logistic TEF and Rayleigh TEF. The PE, Bias, Variation, MRE and RMS-PE for 4p-Kappa, Logistic and Rayleigh are listed in Table I. From the Table I we can see that 4p-Kappa has lower PE, Bias, Variation, MRE and RMS-PE than Logistic and Rayleigh TEF. We can say that our proposed model fits better than the other one. In the table II we have listed estimated values of SRGM with different testing-efforts. We have also given the values of SSE, R^2, and MSE. We observed that our proposed model has smallest MSE and SSE value when compared with other models. The 95% confidence limits for the all models are given in the Table III. All the calculations can found in the appendix. Fig .4 shows the RE curves for the different selected models.

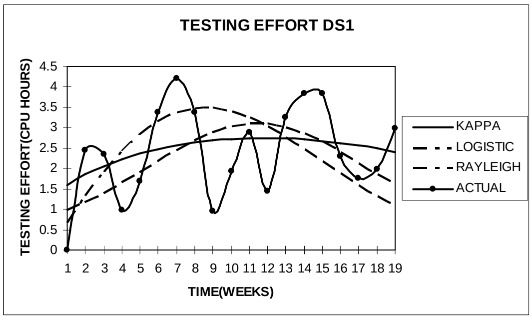


Fig 1. Observed/estimated 4p-Kappa, Logistic and Rayleigh TEF for DS1

.

DS2: the dataset used here presented by wood from a subset of products for four separate software releases at Tandem Computer Company. Wood Reported that the specific products & releases are not identified and the test data has been suitably transformed in order to avoid

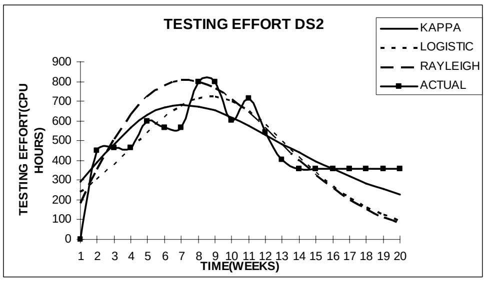


Fig 5. Observed/estimated 4p-Kappa, Logistic and Rayleigh TEF for DS2.

Confidentiality issue. Here we use release 1 for illustrations. Over the course of 20 weeks, 10000 CPU hours are consumed and 100 software faults are removed. Similarly we used a non-linear least square Mat-Lab program to estimate of the Parameters for TEF in the case of DS2 are a=12250(CPU hours), h=-0.584, k=-0.3194, α=5.583, ξ=9.664. Correspondingly the estimated parameters of Logistic TEF are N=9974(CPU hours), A=13.22 and b=0.2881/week and Rayleigh TEF N=9669 and b=0.009472/week. The computed Bias, Variation, MRE, and RMS-PE for 4p-KaPppa TEF, Logistic TEF and Rayleigh TEF are listed in the table IV ,fig 5 graphically illustrate the comparisons between the observed failure data, and the data estimated by the 4p-Kappa TEF, Logistic TEF and Rayleigh TEF. From the figure 5 we can observe the 4p-Kappa curve covers the maximum points like other TEFs. Now from the table V we can conclude that our TEF is better fit than other. Their 95% confidence bounds are given in the table VI. From the above we can see that SRGM with 4p- Kappa TEF have less MSE than other models.

6. Optimal Software Release Policy

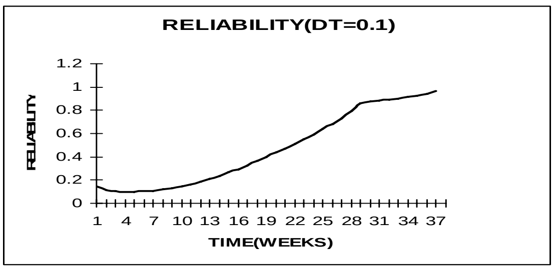
6.1 Software Release-Time Based on Reliability Criteria

Generally software release problem associated with the reliability of a software system. Here in this first we discuss the optimal time based on reliability criterion. If we know software has reached its maximum reliability for a particular time. By that we can decide right time for the software to be delivered out. Goel and Okumoto first dealed with the software release problem considering the software cost-benefit. The conditional reliability function after the last failure occurs at time t is obtained by

$$R(t+\Delta t/t)=\exp(-[m(t+\Delta t/t)-m(t)])$$
$$= \exp\left(-m(\Delta t) \times \exp(-r \times W*(t))\right) \quad (27)$$

Taking the logarithm on both sides of the above equation and rearrange the above equation we obtain

$$\ln R = -m(\Delta t) \times \exp(-r \times W*(t)) \quad (28)$$



thus

Fig 9 plot for reliability of first dataset at Δt=0.1

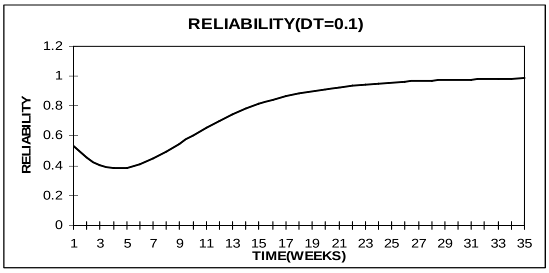


Fig 10 plot for reliability of second dataset at Δt=0.1

$$W*(t) = \frac{1}{r}\left[\ln m(\Delta t) - \ln\ln\left(\frac{1}{R}\right)\right] \quad (29)$$

By solving the eq (28) and eq(12) we can calculate that the testing time needed to reach the desired reliability level. For example for the first dataset the values are a =65(CPU hours), h=0.65, k=0.6214, α=14.98, ξ=5.85, A=492.8 and r=0.02532 this software has been run for operational time until it reaches its reliability level 0.85(Δt=0.1) is t=28.8 weeks. To reach the reliability level at 0.90 is t=32.9 weeks. In the way for the dataset2 a=12250(CPU hours), h=-0.584, k=-0.3194, α=5.583, ξ=9.664, A=123.3 and r=0.0001806, software has been run for operational time until it reaches its reliability level 0.85(Δt=0.1) is t=16.7, its reliability level 0.92(Δt=0.1) is t=20.8, its reliability level 0.960(Δt=0.1) is t=26.1.

6.2 Optimal release time based on cost-reliability criterion

This section deals with the release policy based on the cost-reliability criterion. Using the total software cost evaluated by cost criterion, the cost of testing-effort expenditures during software testing/development phase and the cost of fixing errors before and after release are

$$C(T) = C_1\, m(T) + C_2\left[m(T_{LC}) - m(T)\right] + C_3\left(\int_0^T w(x)\,dx\right) \quad (30)$$

Where C_1 the cost of correcting an error during testing, C_2 is the cost of correcting an error during the operation, $C_2 > C_1$, C_3 is the cost of testing per unit testing effort expenditure and T_{LC} is the software life-cycle length.
from reliability criteria, we can obtain the required testing time needed to reach the reliability objective R_0. Our aim is to determine the optimal software release time that minimizes the total software cost to achieve the desired software reliability. Therefore, the optimal software release policy for the proposed software reliability can be formulated as Minimize C(T) subjected to $R(t+\Delta t/t) \geq R_0$ for $C_2 > C_1$, $C_3 > 0$, $\Delta t > 0$, $0 < R_0 < 1$. Differentiate the equation (30) with respect to T and setting it to zero, we obtain

$$\frac{d}{dT}C(T)=C_1\left(\frac{d}{dT}m(T)\right)+C_2\left[\frac{\partial}{\partial T}m(T_{LC})-\left(\frac{d}{dT}m(T)\right)\right]+C_3\,w(T) \tag{31}$$

$$\frac{\partial}{\partial T}m(T_{LC})=0$$

$$\frac{d}{dT}C(T)=C_1\left(\frac{d}{dT}m(T)\right)+C_2\left[-\left(\frac{d}{dT}m(T)\right)\right]+C_3\,w(T)=0 \tag{32}$$

$$\frac{d}{dT}m(T)=\lambda(T)$$

$$\frac{\lambda(T)}{w(T)}=\frac{c3}{c2-c1}$$

$$a\,r\,e^{-r\,W(t)}=r\,(a-m(T)) \tag{33}$$

b)

When T=0 then m(0)=0 and when T->∞, there are several cases are existing depending on the h and k values and analyze the minimum value of C(T) from eq (31) is used to define the two cases $\frac{\lambda(T)}{w(T)}$ at T=0.

1) if $\frac{\lambda(0)}{w(0)}=a\times r\le\frac{C3}{C2-C1}$, then $\frac{\lambda(T)}{w(T)}\le\frac{C3}{C2-C1}$ for $0<T<T_{LC}$ it can be obtained at dC(T)/dT>0 for $0<T<T_{LC}$ and the minimal value can found at C(T) can be found at T=0.

if $\frac{\lambda(0)}{w(0)}=a\times r>\frac{C3}{C2-C1}>\frac{\lambda(T)}{w(T)}=a\times r\times e^{-r\times\alpha}$ there can be found a finite and unique real number T_0

$$T_0=\frac{-\left(-\frac{\left(-\frac{-r\,a\left(1-h\left(\frac{\alpha+k\,\xi}{\alpha}\right)^{\frac{1}{k}}\right)^{\frac{1}{h}}+\ln\left(-\frac{C3}{A\,r\,(-C2+C1)}\right)}{r\,a}\right)^{h}-1}{h}\right)^{k}\alpha+\alpha+k\,\xi}{k} \tag{34}$$

because dC(T)/dT<0 for $0<T<T_0$ and dC(T)/dT>0 for $T>T_0$, the minimum of C(T) is at $T=T_0$ for $T_0\le T$ we can easily get the required testing time needed to reach the reliability objective R_0 . Here our goal is to minimize the total software cost under desired software reliability and then the optimal software release time is obtained. That is can minimize the C(T) subjected to $R(t+\Delta t/t)\ge R_0$ where $0<R_0<1$[Yamada 1985,Huang 1999] T^* =optimal software release time or total testing time $=\max\{T_0,T_1\}$.Where T_0 =finite and unique solution T satisfying eq.(30) T_1 =finite and unique T satisfying $R(t+\Delta t/t)=R_0$ By combining the above analysis and combining the cost and reliability requirements we have the following theorem.

Theorem 1: assume C2 <C1<0, C3<0, Δt>0, and $0<R_0<1$. Let T* be the optimal software release time

a) If $\frac{\lambda(0)}{w(0)}>\frac{C3}{C2-C1}$ and $\frac{\lambda(T)}{w(T)}=a\times r\times e^{-r\times\alpha}\le\frac{C3}{C2-C1}$

then $T^*=\begin{cases}\max(T_0,T_1)\ \textbf{for}\ R\left(\frac{\Delta t}{0}\right)<R_0<1\\ T_0\ \textbf{for}\ 0<R_0<\left(\frac{\Delta t}{0}\right)\end{cases}$

b) **if** $\frac{\lambda(0)}{w(0)}\ge\frac{C3}{C2-C1}$ **then** $T^*\ge\begin{cases}T_1\ \ \textbf{for}\ R\left(\frac{\Delta t}{0}\right)<R_0<1\\ 0\ \textbf{for}\ 0<R_0<R\left(\frac{\Delta t}{0}\right)\end{cases}$

c) **if** $\frac{\lambda(0)}{w(0)}\le\frac{C3}{C2-C1}$ **then** $T^*=\begin{cases}T_1\ \ \textbf{for}\ R\left(\frac{\Delta t}{0}\right)<R_0<1\\ 0\ \textbf{for}\ 0<R_0<R\left(\frac{\Delta t}{0}\right)\end{cases}$

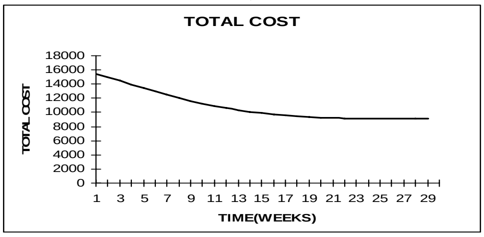


Fig 10 Total software cost for the first dataset vs Time

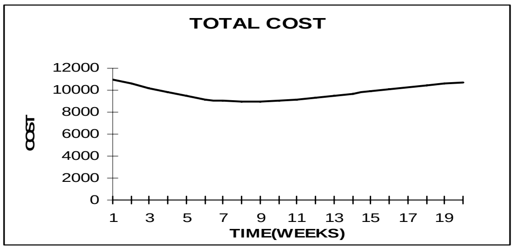


Fig 11 Total software cost for the second dataset vs Time

From the dataset one estimated values of SRGM with Kappa TEF 65(CPU hours), h=0.65, k=0.6214, α=14.98, ξ=5.85, A=492.8 and r=0.02532 when Δt=0.1 R_0 =0.85 and we let C_1=1, C2 =50, C_3 =100 and T_{LC} =100 the estimated time T_1=23.6 weeks and release time from eq 30 T_0 =12.35 weeks. Now optimal Release Time max (12.35, 23.6) is T*=23.6 weeks. Fig 10 shows the change in software cost during the time span. Now total cost of the software at optimal time 5713.

From the dataset two estimated values of SRGM with Kappa TEF a=12250(CPU hours), h=-0.584, k=-0.3194, α=5.583, ξ=9.664, A=123.3 and r=0.0001806 when Δt=0.1 R_0=0.85 and we let C_1=1, C2=150, C_3 =1 and T_{LC} =100 the estimated time T_1=16.7 weeks and release time from eq 30 T_0 =8.06 weeks. Now optimal Release Time max (8.06, 16.7) is T*=16.7 weeks. Fig 11 shows the change in software cost during the time span. Now total cost of the software at optimal time 10069.

TABLE ICOMPARISION RESULT FOR DIFFERENT TEF APPLIED TO DS1

TEF	Bias	Variation	MRE	RMS-PE
4p-Kappa	-0.005641	0.9222	0.00001	0.922215
Logistic	-0.098262	1.306677	0.022246	1.302977
Rayleigh	0.830337	2.169314	0.052676	2.004112

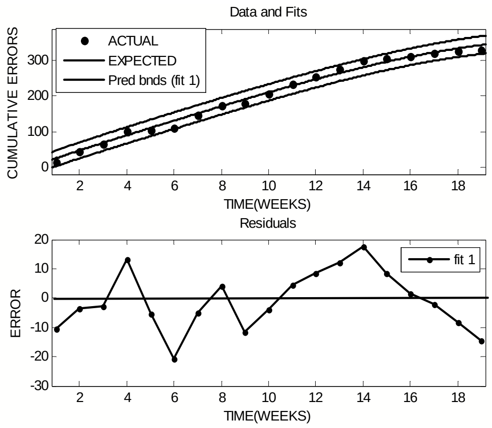


Fig 2. Cumulative and residual error for SRGM with 4p-Kappa TEF for DS1

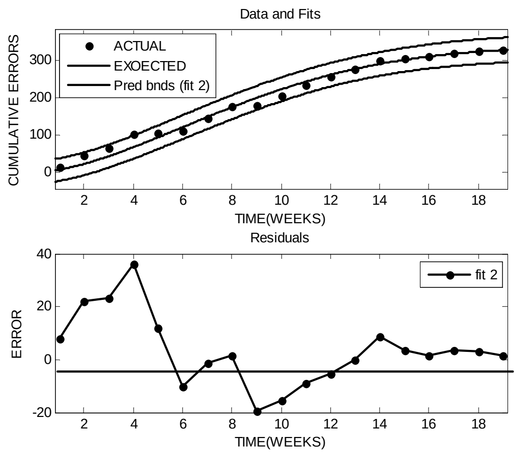


Fig 3. Cumulative and residual error for delayed S shaped model with 4p-Kappa TEF for DS1

Table II ESTIMATED PARAMETER VALUES AND MODEL COMPARISION FOR DS1

Models	a	r	SSE	R^2	MSE
SRGM with 4p-Kappa TEF	492.8	0.02532	1844	0.9906	108.57
Delayed S shaped model with GMW	342.6	0.09984	6027	0.9693	354.56
SRGM with Logistic TEF	395.6	0.04164	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 III 95% CONFIDENCE LIMIT FOR DIFFERENT SELECTED MODELS (DS1)

Models	a		r	
	Lower	Upper	Lower	Upper
SRGM with 4p-Kappa TEF	419.1	566.4	0.01945	0.03118
SRGM with Logistic TEF	358	433.2	0.03399	0.04928
SRGM with Rayleigh TEF	348.6	569.6	0.01651	0.03817
Yamada Delayed S shaped Model with 4p-Kappa TEF	313.7	371.6	0.08535	0.1143
Yamada Delayed S shaped Model with Logistic TEF	291	347.5	0.1088	0.1589
Yamada Delayed S shaped Model with Rayleigh TEF	288.7	377.7	0.07507	0.1258
G-O model	465.4	1056	0.01646	0.04808
Yamada Delayed S shaped model	343.7	404.4	0.1748	0.2205

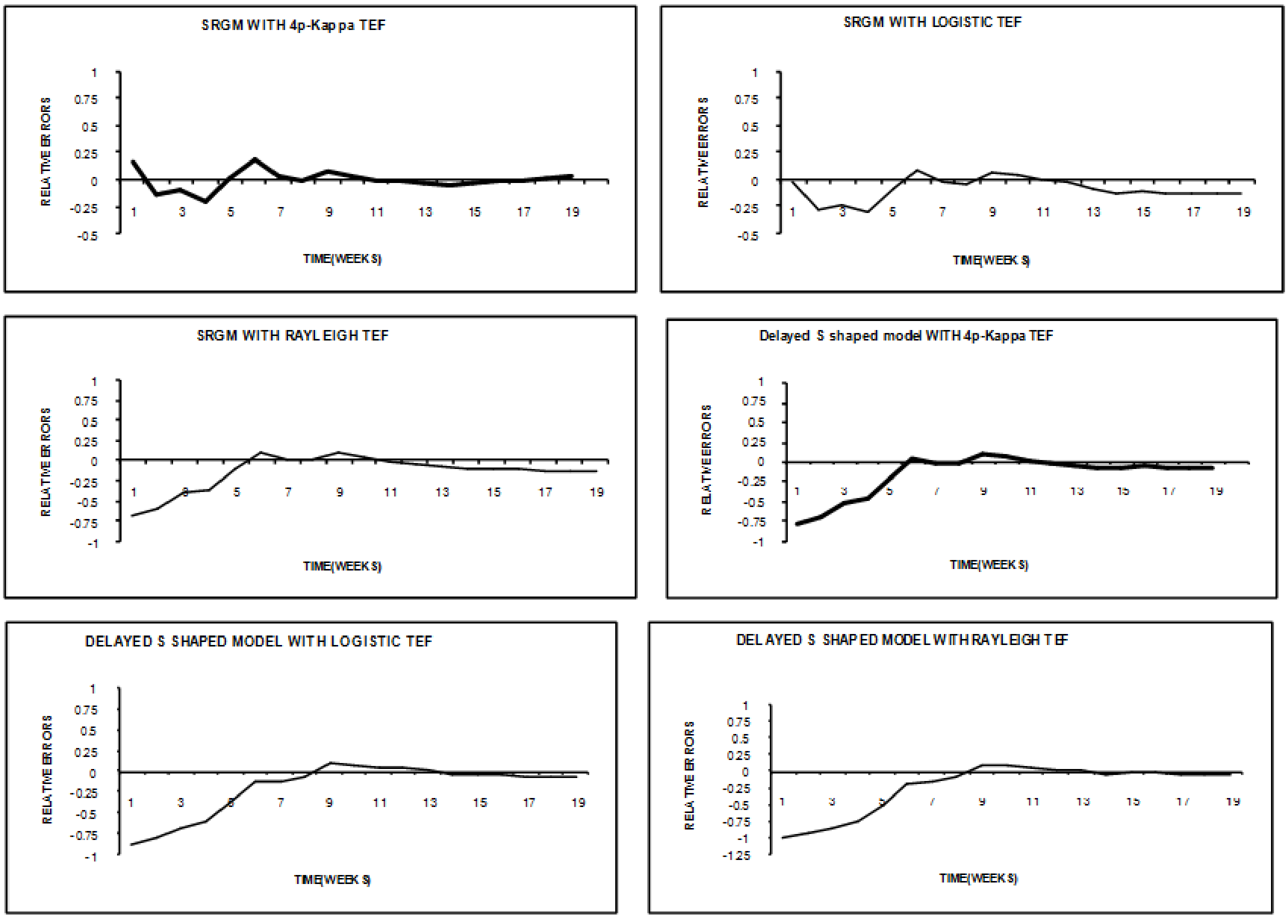


Fig.4 RE curves of selected models compared with actual failure data (DS1)

Table IV COMPARISION RESULT FOR DIFFERENT TEF APPLIED TO DS2

TEF	Bias	Variation	MRE
4p-Kappa	0.4628	98.1	0.015
Logistic	-19.345	198.44	0.026
Rayleigh	121.61	322	0.055

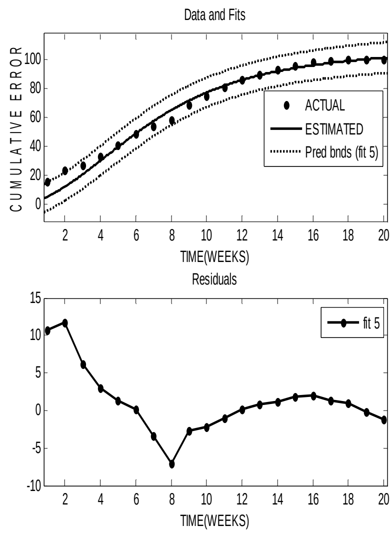


Fig 6. Cumulative and residual error for SRGM with 4p-Kappa TEF for DS2

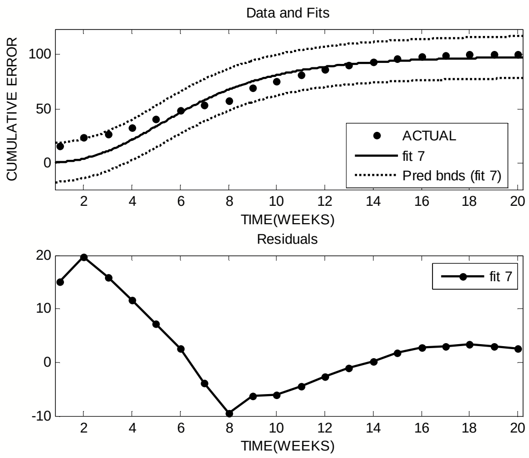


Fig 7. Cumulative and residual error for delayed S shaped model with 4p-Kappa TEF for DS2

Table V
ESTIMATED PARAMETER VALUES AND MODEL COMPARISION FOR DS2

Models	a	r	SSE	R^2	MSE
SRGM with 4p-Kappa TEF	123.3	0.0001806	388	0.9761	21.55
Delayed S shaped model with 4p-Kappa	100.6	0.0005562	1319	0.9189	73.96
SRGM with Logistic TEF	112.3	0.0002399	433.1	0.9734	24.06
Delayed S shaped model with Logistic TEF	96.88	0.0006853	1577	0.903	87.61
SRGM with Rayleigh TEF	120.9	0.0001791	792.5	0.9513	44.03
Delayed S shaped model with Rayleigh TEF	99.4	0.0005434	1930	0.8813	107.1

Table V ESTIMATED PARAMETER VALUES AND MODEL COMPARISION FOR DS2

Models	a	r	SSE	R^2	MSE
SRGM with 4p-Kappa TEF	123.3	0.0001806	388	0.9761	21.55
Delayed S shaped model with 4p-Kappa	100.6	0.0005562	1319	0.9189	73.96
SRGM with Logistic TEF	112.3	0.0002399	433.1	0.9734	24.06
Delayed S shaped model with Logistic TEF	96.88	0.0006853	1577	0.903	87.61
SRGM with Rayleigh TEF	120.9	0.0001791	792.5	0.9513	44.03
Delayed S shaped model with Rayleigh TEF	99.4	0.0005434	1930	0.8813	107.1

Table VI 95% CONFIDENCE LIMIT FOR DIFFERENT SELECTED MODELS (DS2)

Models	a		R	
	Lower	Upper	Lower	Upper
SRGM with 4p-Kappa TEF	107.8	138.8	0.0001356	0.0002256
SRGM with Logistic TEF	101.4	123.1	0.000186	0.0002938
SRGM with Rayleigh TEF	98.4	143	0.0001122	0.0002461
Yamada Delayed S shaped Model with 4p-Kappa TEF	91.19	110	0.0004408	0.0006715
Yamada Delayed S shaped Model with Logistic TEF	88.64	105.1	0.0005346	0.0008359
Yamada Delayed S shaped Model with Rayleigh TEF	88.24	110.6	0.0003991	0.0006877

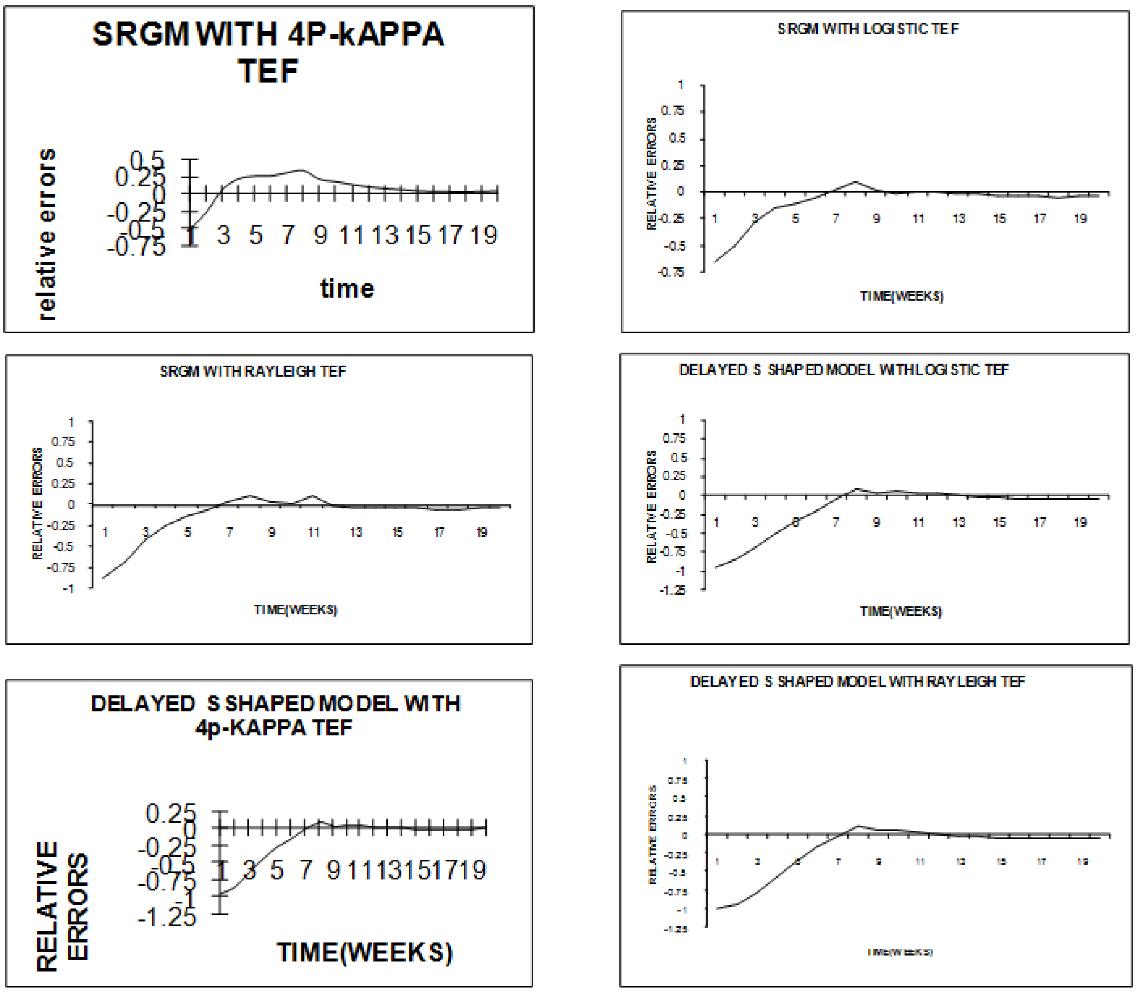


Fig.8 RE curves of selected models compared with actual failure data (DS2)

Appendix -A

$$\frac{dm_d(t)}{dt} \times \frac{1}{w(t)} = r1 \times \left[a - m_d(t)\right]$$

$$m_d(t) = a \times \left(1 - e^{-r \times W(t)}\right)$$

$$\frac{dm_r(t)}{dt} \times \frac{1}{w(t)} = r2 \times \left(m_d(t) - m_r(t)\right)$$

$$\frac{dm_r(t)}{dt} + r2 \times w(t) \times m_r(t) = r2 \times w(t) \times m_d(t)$$

$$I.F\ of\ above\ equation\ e^{\int_0^t r2 \times w(t)\, dt}$$

$$m_r(t) \times e^{r2 \times W(t)} = \int_0^t r2 \times w(t) \times m_d(t) \times e^{r2 \times W(t)}\, dt$$

solving above equation substituting $m_d(t)$

$$m_r(t) = a\left(1 - \left(\frac{\left(r1 \times e^{-r2 \times W(t)} - r2 \times e^{-r1 \times W(t)}\right)}{r1 - r2}\right)\right)$$

Above equation approaches to infinity so we apply the L' Hospitals Rule by letting

$$f(r2) = \left(r1 \times e^{-r2 \times W(t)} - r2 \times e^{-r1 \times W(t)}\right)$$
$$g(r2) = r1 - r2$$

$$\lim_{r2 \to r1} \frac{f(r2)}{g(r2)} = \lim_{r2 \to r1} \frac{(f(r2) - f(r1))}{(g(r2) - g(r1))}$$

$$f'(r1) = -r1 \times W(t) \times e^{[-r1 \times W(t)]} - e^{[-r1 \times W(t)]}$$
$$g'(r1) = -1$$

And $\frac{f'(r1)}{g'(r1)} = (1 + r1 \times W(t)) \times e^{[-r1 \times W(t)]}$

Appendix -B

Using the estimated parameters a, h, k, α, and ξ above, we estimate the reliability growth parameters A and r in (12). Suppose that the data on the cumulative number of detected errors y_k in a given time interval (0, t_k] (k = 1, 2,..., n) are observed. Then, the joint probability mass function, i.e. the likelihood function for the observed data, is given by

$$L \equiv \left\{Pr\left(N(t_1) = y_1, N(t_2) = y_2 \ldots\ldots N(t_n)\right) = y_n\right\}$$

$$\prod_{k=1}^{n} \frac{\left[m(t_k) - m(t_{k-1})\right]^{y_k - y_{k-1}}}{\left(y_k - y_{k-1}\right)} \times \exp\left(-m(t_n)\right)$$

From eq :13 $\frac{\partial}{\partial a}\ln(L) = 0$

$$0 = \sum_{k=1}^{n} \frac{y_k - y_{k-1}}{a} - 1 + \left(1 + r\, t_n\right) e^{-r\, W(t_n)}$$

$$a = \frac{y_n}{1 - \left(1 + r\, W(t_n)\right) e^{-r\, W(t_n)}}$$

$$\frac{\partial}{\partial r}\ln(L) = 0$$

$$a \times \left(W(t_n)\right)^2 \times \exp\left(-r \times W(t_n)\right) = \sum_{k=1}^{n} \left(y_k - y_{k-1}\right) \times \left\{\left(\left(W(t_k)\right)^2 \times \exp\left(-r \times W(t_k)\right) - \left(W(t_{k-1})\right)^2 \times \exp\left(-r \times W(t_{k-1})\right)\right) \Big/ \left(\left(1 + r \times W(t_{k-1})\right) \times \exp\left(-r \times W(t_{k-1})\right) - \left(1 + r \times W(t_k)\right) \times \exp\left(-r \times W(t_k)\right)\right)\right\}$$

Conclusions

In this paper, we proposed a SRGM incorporating the 4p-Kappa testing effort function. We observed that most of software failure data is time dependent. By incorporating 4p- Kappa testing effort in to SRGM we can make realistic assumptions about the software failure. The experimental results indicate that our proposed model fits fairly well compared to other models.

REFERENCES

[1] L. Goel and K. Okumoto, A time dependent error detection rate model for a large scale software system, Proc. 3rd USA-Japan Computer Conference, pp. 3540, San Francisco, CA (1978).

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

[3] Bokhari, M.U. and Ahmad, N. (2005), "Software reliability growth modeling for exponentiated Weibull functions with actual software failures data", in Proceedings of 3rd International Conference on Innovative Applications of Information Technology for Developing World (AACC'2005), Nepal.

[4] Bokhari, M.U. and Ahmad, N. (2006), "Analysis of a software reliability growth models: the case of log-logistic test-effort function", in Proceedings of the 17th International Conference on Modelling and Simulation (MS'2006), Montreal, Canada, pp. 540-5.
[5] C.-Y. Huang, S.-Y. Kuo, J.Y. Chen, Analysis of a software reliability growth model with logistic testing effort function proceeding of Eighth International Symposium on Software Reliability Engineering, 1997, pp. 378–388.
[6] Goel, A.L., "Software reliability models: Assumptions, limitations, and applicability", IEEE Transactions on Software Engineering SE-11 (1985) 1411-1423.
[7] Huang, C.Y. and Kuo, S.Y. (2002), "Analysis of incorporating logistic testing-effort function into software reliability modeling", IEEE Transactions on Reliability, Vol. 51 No. 3, pp. 261-70.
[8] Huang, C.Y., Kuo, S.Y. and Chen, I.Y. (1997), "Analysis of software reliability growth model with logistic testing-effort function", in Proceeding of 8th International Symposium on Software Reliability Engineering (ISSRE'1997), Albuquerque, New Mexico, pp. 378-88.
[9] Huang, C.Y., Kuo, S.Y. and Lyu, M.R. (1999), "Optimal software release policy based on cost, reliability and testing efficiency", in Proceedings of the 23rd IEEE Annual International
[10] Huang, C.Y., Kuo, S.Y. and Lyu, M.R. (2000), "Effort-index based software reliability growth models and performance assessment", in Proceedings of the 24th IEEE Annual International Computer Software and Applications Conference (COMPSAC'2000), pp. 454-9.
[11] Huang, Lyu and Kuo "An Assesment of testing effort dependent software reliability Growth model". IEEE transactions on Reliability Vol 56, No: 2, June 2007
[12] 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.
[13] Hoang Pham and Chin-Diew Lai "On Recent Generalizations of the Weibull Distribution" IEEE TRANSACTIONS ON RELIABILITY, VOL. 56, NO. 3, SEPTEMBER 2007.
[14] Kapur, P.K. and Younes, S. (1994), "Modeling an imperfect debugging phenomenon with testing effort", in Proceedings of 5th International Symposium on Software Reliability Engineering (ISSRE'1994), pp. 178-83.
[15] K. Pillai and V. S. Sukumaran Nair, "A model for software development effort and cost estimation," IEEE Trans. Software Engineering, vol. 23, no. 8, August 1997.
[16] K. Srinivasan and D. Fisher, "Machine learning approaches to estimating software development effort," IEEE Trans. Software Engineering, vol. 21, no. 2, pp. 126–136, 1995.
[17] Lai, Min Xie , D. N. P. Murthy "A Modified Weibull Distribution "IEEE TRANSACTIONS ON RELIABILITY, VOL. 52, NO. 1, MARCH 2003.
[18] M. Ohba, Software reliability analysis models, IBM J. Res. Dev. 28 (1984) 428–443.
[19] M.R. Lyu, Handbook of Software Reliability Engineering, Mcgraw Hill, 1996.
[20] Mudholkar, G.S. and Srivastava, D.K. (1993), "Exponentiated Weibull family for analyzing bathtub failure-rate data", IEEE Transactions on Reliability, Vol. 42, pp. 299-302.
[21] Pham, H. (2000), Software Reliability, Springer-Verlag, New York, NY.
[22] P. K. Kapur and S. Younes, "Software reliability growth model with error dependency," Microelectronics and Reliability, vol. 35, no. 2, pp. 273–278, 1995.
[23] Quadri, S.M.K., Ahmad, N., Peer, M.A. and Kumar, M. (2006), "Nonhomogeneous Poisson process software reliability growth model with generalized exponential testing effort function", RAU Journal of Research, Vol. 16 Nos 1-2, pp. 159-63.
[24] Rameshwar D. Gupta and Debasis Kundu "generalized exponential distribution: different method of estimations" j. statist. comput. simul., 2000, vol. 00, pp. 1 – 22 14 november 2000.
[25] S. Yamada, H. Ohtera and R. Narihisa, "Software Reliability Growth Models with Testing-Effort," IEEE Trans. Reliability, Vol. R-35, pp. 19-23 (1986).
[26] S. Yamada, H. Ohtera, Software reliability growth model for testing effort control, Eur. J. Oper. Res. 46 (1990) 343–349.
[27] S.Yarnada, S.Osalci, "Software reliability growth modeling: models and applications", IEEE Trans. Software Engineering, vol.l I, no.12, p.1431-1437, December 1985.
[28] Xie, M. (1991), Software Reliability Modeling, World Scientific Publication, Singapore.
[29] Yamada, S., Ohba, M., Osaki, S., 1983. S-shaped reliability growth modeling for software error detection. IEEE Trans. Reliab. 12, 475–484.
[30] Yamada, S. and Osaki, S. (1985b), "Cost-reliability optimal release policies for software systems", IEEE Transactions on Reliability, Vol. R-34 No. 5, pp. 422-4.

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.

Dr.K.NageswaraRao received B.Tech(electronics) from Karnataka University, M.Tech(comp) from Andhra University and PhD from Andhra University. Presently working as Professor& H.O.D in P.V.P.S.I.T, Vijayawada affiliated to J.N.T.U, Kakinada. My area of interest is Robotics, Software Engineering, Algorithms, and Software Reliability.

Dr.K.Pallam Sety received M.Tech(Comp) from Andhra University, PhD from Andhra University. Presently working as Professor in dept of Computer Science in Andhra University. My area of research is Software Engineering, Software Reliability,, NetworkSecurity.

Shaheda Akthar received Bachelor of computer science from Acharya Nagarjuna University, M.S (Software Systems) from BITS, Pilani, pursuing PhD from Acharya Nagarjuna University. Presently working as Asscociate .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.