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

Sk.Md.Rafi 1, Dr. K.Nageswara Rao2, Dr.K.Pallam Sety3, Shaheda Akthar4
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
m0 (t) : Cumulative number of faults detected up to t.
m0 (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.
r1 : Constant fault detection rate in the Delayed S-shaped model with 4p-kappa TEF
r2 : 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
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

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

Where \( a > 0, \alpha > 0, \xi > 0, h > 0 \) at \( t > 0 \) where \( a \) is the total effort expenditure, \( \alpha \) controls the scale of the distribution, \( k \) and \( \xi \) are shape parameters, \( \xi \) is location parameter

\[ \frac{d}{dt} W(t) = \alpha \times \left( \frac{\xi}{1 - k \times (1 - h \times \frac{(t - \xi)}{\alpha})^\frac{1}{k}} \right) \]  

Following are the some of the special cases;

1) The generalized Pareto distribution

\[ W(t) = a \times \left( 1 - \left( 1 - k \times \frac{(t - \xi)}{\alpha} \right)^\frac{1}{k} \right) \]  

a>0,\alpha>0,\xi>0,h=1 at t>0  

2) The generalized extreme value distribution at

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

a>0,\alpha>0,\xi>0,h=0 at t>0  

3) The generalized Logistic distribution

\[ W(t) = \frac{a}{1 + \left( 1 - k \times \frac{(t - \xi)}{\alpha} \right)^\frac{1}{k}} \]  

a>0,\alpha>0,\xi>0,h=-1 at t>0  

4) The Gumbel distribution

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

a>0,\alpha>0,\xi>0,h=1 and k=1 at t>0  

5) the exponential distribution

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

a>0,\alpha>0,\xi>0,h=1 and k=0 at t>0  

6) the uniform distribution

\[ W(t) = \frac{a \times (t - \xi)}{\alpha} \]  

a>0,\alpha>0,\xi>0,h=1 and k=1 at t>0  

And also other related distributions

The testing effort reaches its maximum value at

\[ T_{max} = \frac{-\left( \ln\left( \frac{-1 + z + k}{1 + z + x \times k} \right) \times k \times \alpha - \alpha + \xi \times k \right)}{k} \]  

at t>0

3. Software Reliability growth model and testing effort functions

3.1 SRGM with 4p Kappa Testing-effort function


(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 \((t, t+\Delta t)\) 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 the following:

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

(10)

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

(11)

Substituting $W(t)$ from eq.(1), we get

$$m(t) = A \times \left(1 - \int_{0}^{t} e^{-r \cdot (t - s)} ds\right)$$

(12)

This is an NHPP model with mean value function with the GMW testing-effort expenditure.

Now failure intensity is given by

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

(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

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = r_1 \times \left[a - m(t)\right]$$

(14)

and

$$\frac{dm(t)}{dt} \times \frac{1}{w(t)} = r_2 \times \left[a - m(t)\right]$$

(15)

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

$$m_d(t) = a \times \left(1 - e^{-r_1 \times W(t)}\right)$$

(16) and

$$m_r(t) = a \times \left(1 - e^{-r_2 \times W(t)}\right)$$

(17)

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

$$m(t) = a \times \left(1 - (1 + r_x W(t) + r_x W^2(t)) \right)$$

(18)

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

$$\lambda(t) = a \times r^2 \times w(t) \times W(t) \times e^{-r \times W(t)}$$

(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

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

(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
\[ RE = \frac{(m(t_q) - q)}{q} \]  
(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]
\[ SSE = \sum_{i=1}^{n} \left( y_i - m(t_i) \right)^2 \]  
(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 \( i=1,2,\ldots,n \).

\[ PE_i = Actual(\text{observed}) - Predicted(\text{estimated}) \]  
(23)

\[ Bias = \frac{1}{n} \sum_{i=1}^{n} PE_i \]  
(24)

\[ Variation = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left( PE_i - Bias \right)^2} \]  
(25)

\[ MRE = \frac{|M_{\text{estimated}} - M_{\text{actual}}|}{M_{\text{actual}}} \]  
(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 1,317,000 lines 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 \( \alpha=65 \) (CPU hours), \( h=0.65 \), \( k=0.6214 \), \( \alpha=14.98 \), \( \xi=5.85 \). Correspondingly the estimated parameters of Logistic TEF are \( N=54.84 \) (CPU hours), \( A=13.03 \) and \( b=0.2263/\text{week} \) and Rayleigh TEF \( N=49.32 \) and \( b=0.00684/\text{week} \). 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.

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 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 the Parameters for TEF in the case of DS2 are \( \alpha=12250 \) (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-KaPpa 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 dealt 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

\[ R(t+\Delta t)/t) = \exp(-m(t+\Delta t/t)-m(t)) \]

(27)

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

\[ \ln R = -m(\Delta t) \times \exp(-r \times W(t)) \]

(28)

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

\[ C(T) = C_1 \{ m(T) + C_2 [m(T_{1C}) - m(T)] \} + C_3 \int_0^T w(t) dt \]

(30)

Where \( C_1 \) is 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 \( T_{1C} \) 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+Δt/t)≥ 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
When $T=0$ then $m(0)=0$ and when $T\to \infty$, there are several cases 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 at $T=0$.

1) if

$$\frac{\lambda(0)}{w(0)} = a \times r \leq \frac{C_3}{C_2 - C_1},$$

then

$$\frac{\lambda(T)}{w(T)} \leq \frac{C_3}{C_2 - C_1}$$

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 $T=0$.

$$\frac{\lambda(0)}{w(0)} = a \times r > \frac{C_3}{C_2 - C_1} \Rightarrow \frac{\lambda(T)}{w(T)} = a \times r \times e^{-r x a}$$

can be found a finite and unique real number $T_0$

$$T_0 \leq \frac{C_3}{C_2 - C_1} \times \frac{1-e^{-r x a}}{r x a}$$

(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 $T_0 \leq T$ 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 $R(t+\Delta t/T) \geq R_0$ 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 $C_2 < C_1 < 0$, $C_3 < 0$, $\Delta T = 0$, and $0<R_0<1$. Let $T^*$ be the optimal software release time a) If $\frac{\lambda(0)}{w(0)} > \frac{C_3}{C_2 - C_1}$ and

$$\frac{\lambda(T)}{w(T)} = a \times r \times e^{-r x a} \leq \frac{C_2}{C_2 - C_1}$$

then

$$max(T_0, T_1) \text{ for } \frac{\lambda(0)}{w(0)} \leq \frac{C_2}{C_2 - C_1} \leq R_0 < 1$$

$$T^* = \left\{ \begin{array}{ll}
T_1 \text{ for } \frac{\lambda(0)}{w(0)} \leq \frac{C_2}{C_2 - C_1} \leq R_0 < 1 \\
0 \text{ for } 0 < R_0 < \frac{\lambda(0)}{w(0)}
\end{array} \right.$$

b) if $\frac{\lambda(0)}{w(0)} \geq \frac{C_3}{C_2 - C_1}$ then $T^*$

c) if $\lambda(0) \leq \frac{C_3}{C_2 - C_1}$ then $T^*$

From the dataset one estimated values of SRGM with Kappa TEF 65(CPU hours), $h=0.65$, $k=0.6214$, $\alpha=14.98$, $\xi=5.85$, $A=492.8$ and $r=0.02532$ when $\Delta t=0.1 R_0 = 0.85$ and we let $C_1=1$, $C_2 =50$, $C_3 =100$ 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), $h=0.584$, $k=-0.3194$, $\alpha=-5.583$, $\xi=9.664$, $A=123.3$ and $r=0.0001806$ when $\Delta t=0.1 R_0 = 0.85$ and we let $C_1=1$, $C_2 =150$, $C_3 =1$ and $T_{LC} =100$ the estimated time $T_1=23.6$ 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 I** COMPARISON RESULT FOR DIFFERENT TEF APPLIED TO DS1

<table>
<thead>
<tr>
<th>TEF</th>
<th>Bias</th>
<th>Variation</th>
<th>MRE</th>
<th>RMS-PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4p-Kappa</td>
<td>-</td>
<td>0.9222</td>
<td>0.00001</td>
<td>0.922215</td>
</tr>
<tr>
<td>Logistic</td>
<td>-</td>
<td>1.306677</td>
<td>0.032246</td>
<td>1.302977</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>0.830357</td>
<td>2.169314</td>
<td>0.052676</td>
<td>2.004112</td>
</tr>
</tbody>
</table>

Table II ESTIMATED PARAMETER VALUES AND MODEL COMPARISON FOR DS1

<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>r</th>
<th>SSE</th>
<th>R^2</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM with 4p-Kappa TEF</td>
<td>492.8</td>
<td>0.02532</td>
<td>1844</td>
<td>0.9906</td>
<td>108.57</td>
</tr>
<tr>
<td>Delayed S shaped model with GMW</td>
<td>342.6</td>
<td>0.09984</td>
<td>6027</td>
<td>0.9693</td>
<td>354.56</td>
</tr>
<tr>
<td>SRGM with Logistic TEF</td>
<td>395.6</td>
<td>0.04164</td>
<td>2167</td>
<td>0.989</td>
<td>127.46</td>
</tr>
<tr>
<td>Delayed S shaped model with Logistic TEF</td>
<td>319.3</td>
<td>0.1339</td>
<td>11060</td>
<td>0.9436</td>
<td>650.25</td>
</tr>
<tr>
<td>SRGM with Rayleigh TEF</td>
<td>459.1</td>
<td>0.02734</td>
<td>5100</td>
<td>0.974</td>
<td>299.98</td>
</tr>
<tr>
<td>Delayed S shaped model with Rayleigh TEF</td>
<td>333.2</td>
<td>0.1084</td>
<td>15170</td>
<td>0.9226</td>
<td>892.2</td>
</tr>
<tr>
<td>G-O model</td>
<td>760.5</td>
<td>0.03227</td>
<td>2656</td>
<td>0.9865</td>
<td>156.2</td>
</tr>
<tr>
<td>Yamada Delayed S shaped model</td>
<td>374.1</td>
<td>0.1977</td>
<td>3205</td>
<td>0.9837</td>
<td>188.51</td>
</tr>
</tbody>
</table>

Table III 95% CONFIDENCE LIMIT FOR DIFFERENT SELECTED MODELS (DS1)

<table>
<thead>
<tr>
<th>Models</th>
<th>a Lower</th>
<th>a Upper</th>
<th>r Lower</th>
<th>r Upper</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM with 4p-Kappa TEF</td>
<td>419.1</td>
<td>366.4</td>
<td>0.01945</td>
<td>0.03118</td>
</tr>
<tr>
<td>SRGM with Logistic TEF</td>
<td>358</td>
<td>433.2</td>
<td>0.03399</td>
<td>0.04928</td>
</tr>
<tr>
<td>SRGM with Rayleigh TEF</td>
<td>348.6</td>
<td>569.6</td>
<td>0.01651</td>
<td>0.03817</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with 4p-Kappa TEF</td>
<td>313.7</td>
<td>371.6</td>
<td>0.08535</td>
<td>0.1143</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with Logistic TEF</td>
<td>291</td>
<td>347.5</td>
<td>0.1088</td>
<td>0.1589</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with Rayleigh TEF</td>
<td>288.7</td>
<td>377.7</td>
<td>0.07507</td>
<td>0.1258</td>
</tr>
<tr>
<td>G-O model</td>
<td>465.4</td>
<td>1056</td>
<td>0.01646</td>
<td>0.04808</td>
</tr>
<tr>
<td>Yamada Delayed S shaped model</td>
<td>343.7</td>
<td>404.4</td>
<td>0.1748</td>
<td>0.2205</td>
</tr>
</tbody>
</table>
Fig. 4 RE curves of selected models compared with actual failure data (DS1)

Table IV COMPARISION RESULT FOR DIFFERENT TEF APPLIED TO DS2

<table>
<thead>
<tr>
<th>TEF</th>
<th>Bias</th>
<th>Variation</th>
<th>MRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>4p-Kappa</td>
<td>0.4628</td>
<td>98.1</td>
<td>0.015</td>
</tr>
<tr>
<td>Logistic</td>
<td>-19.345</td>
<td>198.44</td>
<td>0.026</td>
</tr>
<tr>
<td>Rayleigh</td>
<td>121.61</td>
<td>322</td>
<td>0.055</td>
</tr>
</tbody>
</table>

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

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

Fig 7. Cumulative and residual error for delayed S shaped model with 4p-Kappa TEF for DS2
<table>
<thead>
<tr>
<th>Models</th>
<th>a</th>
<th>R</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRGM with 4p-Kappa TEF</td>
<td>107.8</td>
<td>0.0001356</td>
</tr>
<tr>
<td>SRGM with Logistic TEF</td>
<td>101.4</td>
<td>0.000186</td>
</tr>
<tr>
<td>SRGM with Rayleigh TEF</td>
<td>98.4</td>
<td>0.0001122</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with 4p-Kappa TEF</td>
<td>91.19</td>
<td>0.0004408</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with Logistic TEF</td>
<td>88.64</td>
<td>0.0005346</td>
</tr>
<tr>
<td>Yamada Delayed S shaped Model with Rayleigh TEF</td>
<td>88.24</td>
<td>0.0003991</td>
</tr>
</tbody>
</table>

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

\[
\frac{dm_d(t)}{dt} = r_1 \times \left[ a - m_d(t) \right]
\]

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

\[
\frac{dm_d(t)}{dt} = r_2 \times \left( m_d(t) - m_r(t) \right)
\]

\[
\frac{dm_r(t)}{dt} = r_2 \times w(t) \times m_r(t) = r_2 \times w(t) \times m_d(t)
\]

\[
IF \ of \ above \ equation \ e^{r_2 \times w(t)} \int_0^t r_2 \times w(t) \times dt
\]

\[
solving \ above \ equation \ substituting \ m_d(t)
\]

\[
m_r(t) = a \left( 1 - \left( \frac{r_1 \times e^{-r_2 \times W(t)} - r_2 \times e^{-r_1 \times W(t)}}{r_1 - r_2} \right) \right)
\]

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

\[
f(r_2) = \left( r_1 \times e^{-r_2 \times W(t)} - r_2 \times e^{-r_1 \times W(t)} \right)
\]

\[
g(r_2) = r_1 - r_2
\]

\[
\lim_{r_2 \to r_1} \frac{f(r_2)}{g(r_2)} = \lim_{r_2 \to r_1} \frac{f(r_2) - f(r_1)}{g(r_2) - g(r_1)}
\]

\[
a \times \left( W(t_n) \right)^2 \times \exp \left( -r \times W(t_n) \right) = \sum_{k=1}^{n} \left( y_k - y_{k-1} \right) \times \left\{ \left( W(t_k) \right)^2 \times \exp \left( -r \times W(t_k) \right) \right\} - \left( 1 + r \times W(t) \right) \times \exp \left( -r \times W(t_k) \right)
\]

\[f'(r_1) = -r_1 \times W(t) \times e^{\left[ -r_1 \times W(t) \right]} - e^{\left[ -r_1 \times W(t) \right]} \]

\[g'(r_1) = -1 \]

And \[\frac{f'(r_1)}{g'(r_1)} = \left( 1 + r \times W(t) \right) \times e^{\left[ -r_1 \times W(t) \right]}\]

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 yk 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 \{Pr(N(t_1) = y_1, N(t_2) = y_2, \ldots, N(t_n) = y_n)\}

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

From eq :13

\[
\frac{\partial}{\partial a} \ln(L) = 0
\]

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

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

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


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 in dept of Computer Science in Andhra University. My area of research is Software Engineering, Software Reliability, NetworkSecurity.

Shaheda Akhtar 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 Associate 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.