Volume 1, No. 2, July-August 2010

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Predictability of Software Reliability With Imperfect Debugging Based on Multiple Change Point

O. NagaRaju *
Asst. Professor
Acharya Nagarjuna University
Nagarjuna Nagar, INDIA
onrajunrt@gmail.com

Dr. R. SatyaPrasad
Associate Professor
Acharya Nagarjuna University
Nagarjuna Nagar, INDIA
profrsp@gmail.com

Prof. R. R. LKantam
Professor
Nagarjuna University
Nagarjuna Nagar, INDIA
kantam_rrl@rediffmail.com

Abstract: In this paper, we propose to investigate some techniques for predictability of software reliability with imperfect debugging. And we also incorporate the concept of multiple change points in SRGM. Some models are proposed and discussed under both ideal and imperfect debugging conditions. A numerical example with real software failure data is presented in detail and the results show that the proposed models can provide better capability to predict the software reliability.

***Keywords*:** Non homogeneous poisson process, software reliability, imperfect debugging, failure data.

I. INTRODUCTION

Several distinct faces of engineering a software are definition ,analysis, development, text and maintenance etc., A very important role is developed by software reliability measurement as a robust, high quality software product between software reliability and assured software quality, as applied by software reliability growth model (SRGM) General formulation can unify various traditional SRGMs.
Generally, at the beginning of the testing phase, inspection can discover many faults and the fault discovery efficiency depends on the fault detection rate, the fault density, the testing-effort and the inspection rate.

The change point problem in the field of soft reliability has been addressed by some papers in recent years. Different assumptions are made in different SRGMs. Therefore problem of imperfect debugging can be applied to different situations as addressed by many authors.

However, this may not be correct since the testing environment may not reproduce the typical use of the system in field operation [4, 10, and 15]. The fault detection phenomenon in the operational phase is different from that in the testing phase. Thus we have to make an adjustment for the selected SRGM that was accurate in the past. Here we will further propose a very useful approach to describe the transitions from the testing to the operational phase. That is, based on the unified theory, we can incorporate the concept of multiple change-points into software reliability modeling. Besides, the proposed models are also discussed under both ideal and imperfect debugging conditions.

II. SOFTWAREOPERATIONAL RELIABILITY MEASUREMENT

A. A General Continuous SRGM

We will discuss a general continuous NHPP model in this section. We let m(t+Δt) be equal to the quasi-arithmetic mean of m(t) and a with weights w(t, Δt) and 1-ω(t, Δt), then

$$g(m(t+\Delta t))=\omega(t,\Delta t)g(m(t))+(1-\omega(t,\Delta t))g(a), 0<\omega(t,\Delta t)<1$$

………. (2.1)

Where g is a real-valued, strictly monotonic, and differentiable function.

$$\frac{g(m(t+\Delta t))-g(m(t))}{\Delta t}=\frac{1-\omega(t,\Delta t)}{\Delta t}(g(a)-g(m(t))),$$

0<ω(t,Δt)<1.

Suppose (1- ω(t, Δt)/ Δt→b(t) as Δt→0, we get the differential equation

$$\frac{\partial}{\partial t}g(m(t))=b(t)g(a)-g(m(t)).$$ …………. (2.2)

For g(x)=x in equation, then

$$\frac{\partial}{\partial t}g(m(t))=b(t)(a-m(t)).$$ ………………. (2.3)

Here, b(t) is the fault detection rate per error. Furthermore, if b(t)=b then if would be the popular SRGM of Goel-Okumoto model(1979)in short known as G-O model.

Theorem: let

$$\frac{\partial}{\partial t}g(m(t))=b(t)(g(a)-g(m(t))),.$$ ……………. (2.4)

where g is a real-valued, strictly monotonic, and differentiable function. We have:

$$m(t) = g^{-1}(g(a) + \{g(m(0) - g(a)\}e^{(a\beta + b)t}$$

and

$$B(t) = \int_0^t a(u)du$$

<u>Corollary1:</u> Based on the weighted arithmetic mean, take g(x)=x in equation and let k=1-m(0)/a, then the mean value function, Our proposed mean value function for the model with the above specification is

$$m(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} ke^{(a\beta + b)t}}{1 + \frac{a\beta}{b_1} ke^{(a\beta + b)t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

, a>0, 0<k≤1………… (2.5)

B. Software Operational Reliability Estimation Based on Multiple Change-Point SRGMs

$$m_1(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} e^{(a\beta + b)t}}{1 + \frac{a\beta}{b_1} e^{(a\beta + b)t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

In reality, the fault detection phenomenon in the operational phase is different from that in the testing phase [7, 16, 17, 19-22]. However, this fact is not distinctly incorporated in many software reliability modeling efforts. Generally, at the beginning of the testing phase, many faults can be discovered by inspection and the fault detection rate depends on the fault discovery efficiency, the fault density, the testing-effort, and the inspection rate. In the middle stage of testing phase, the fault detection rate normally depends on other parameters such as the execution rate of CPU instruction, the code expansion factor, and the scheduled CPU hours per calendar day. Consequently, the fault detection rate can be calculated. We can use this rate to track the progress of checking activities, to evaluate the effectiveness of test planning, and to assess the checking methods we adopted. Practically, during the software development process, the fault detection rate may not be a constant or smooth, i.e., it may be changed at some time moment called change-point.

In general, a change-point is a model which has some parameters which are discontinuous in time. That is, it is the time at which the parameter changes its values. In the recent years, some papers have addressed the change-point [2, 5, 11, and 13].If we want to detect more additional faults during software development process; it is advisable to introduce new tools and techniques. That is these approaches can provide a steady improvement in software testing and productivity. Therefore, the timing of introducing new tools and techniques is a change-point. In this paper, we will show how to improve traditional SRGMs by incorporating the concepts of multiple change points.

C. Our Model with Multiple Change-Points

The G-o model mathematical expression can be expressed as

$$\frac{\partial m(t)}{\partial t} = b(a - m(t))$$

…………………… (2.6)

Solving above equations using the boundary conditions m(0)=0, we have m(t)=a(1-exp[-bt]) and

$$\frac{\partial m(t)}{\partial t} = b(t)[a - m(t)]$$

……………………….. (2.7)

On the same lines we propose our model as

$$m(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} e^{(a\beta + b)t}}{1 + \frac{a\beta}{b_1} e^{(a\beta + b)t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

……………………… (2.8)

Further the model with two change points and a given the fault deletion rate as given by

b(t) = a_1 if 0≤t≤ τ_1

b(t) = a_2 if t> τ

solving similar equations. The solutions are

, 0≤t≤ τ_1…………………..(2.9)

$$m_2(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_1 + a_2(t - \tau_1))t}}{1 + \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_{1_1} + a_2(t - \tau_1))t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

, t> τ ……….. (2.10)

In fact, the above equation can also be derived based on the unified theory, but we have to make some necessary adjustments for Theorem due to multiple change points. Firstly, when0≤t≤ τ_1, if we take g(x)=x,k₁-m(0)/a, and $B_1(t)=\int_0^\tau a_1 du$ from the corollary1, we can easily obtain the mean value function $m_1(t)=a(1-\exp[-b_1t])$ furthermore, when t>τ, if we take g(x)=x and $B_2(t)=\int_0^\tau a_2 du$, then from the corollary 1, we have

$$m_2(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_1 + a_2(t - \tau_1))t}}{1 + \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_1 + a_2(t - \tau_1))t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

If $k_2 = 1 - \frac{m_1(\tau)}{a} = e^{(a\beta + a_1\tau_1)t}$

we obtain

$$m_2(t) = \frac{1}{2}\left\{\left[\frac{1 - \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_1 + a_2(t - \tau_1))t}}{1 + \frac{a\beta}{b_1} e^{(a\beta + a_1\tau_1 + a_2(t - \tau_1))t}}\right](a + \frac{b_1}{\beta}) + a - \frac{b_1}{\beta}\right\}$$

.. (2.11)

Similarly, we can further consider our model with three change points. If the fault detection rate is given by

$$b(t)=\begin{cases} a_1 & \text{if } 0\le t\le \tau_1 \\ a_2 & \text{if } \tau_1\le t\le \tau_2 \\ \ldots\ldots & \\ a_n & \text{if } \tau_{n-1}\le t\le \tau_n \end{cases}$$

The mean value function can be obtained by following similar procedures. These are

$$m(t)=m_1(t)=\frac{1}{2}\left\{\left[\frac{1-\frac{a\beta}{b_1}e^{(a\beta+b)t}}{1+\frac{a\beta}{b_1}e^{(a\beta+b)t}}\right](a+\frac{b_1}{\beta})+a-\frac{b_1}{\beta}\right\}$$

$0\le t\le \tau_1$ .... (2.12)

$$m(t)=m_2(t)=\frac{1}{2}\left\{\left[\frac{1-\frac{a\beta}{b_1}e^{(a\beta+a_1\tau_1+a_2(t-\tau_1))t}}{1+\frac{a\beta}{b_1}e^{(a\beta+a_1\tau_1+a_2(t-\tau_1))t}}\right](a+\frac{b_1}{\beta})+a-\frac{b_1}{\beta}\right\}$$

$\tau_1\le t\le \tau_2$ .... (2.13)

$$m(t)=m_3(t)=\frac{1}{2}\left\{\left[\frac{1-\frac{a\beta}{b_1}e^{(a\beta+a_1\tau_1+a_2(\tau_2-\tau_1)+a_3(t-\tau_3))t}}{1+\frac{a\beta}{b_1}e^{(a\beta+a_1\tau_1+a_2(\tau_2-\tau_1)+a_3(t-\tau_3))t}}\right](a+\frac{b_1}{\beta})+a-\frac{b_1}{\beta}\right\}. \quad (2.14)$$

Finally if

$$b(t)=\begin{cases} a_1 & \text{if } 0\le t\le \tau_1 \\ a_2 & \text{if } \tau_1\le t\le \tau_2 \\ \ldots. & \\ a_n & \text{if } \tau_{n-1}\le t\le \tau_n \end{cases}$$

and $k_n=1-\dfrac{m_n-1(\tau_{n-1})}{a}=\sum_{k=1}^{n-1}a_k\left(\tau_k-\tau_{k-1}\right)$

we can get a solutions for descending multiple change-points of our model

$$m_n(t)=\frac{1}{2}\left\{\left[\frac{1-\frac{a\beta}{b_1}e^{\left(a\beta+\left\{a_n*(t-\tau_n-1)+\sum_{k=1}^{n-1}a_k*(\tau_k-\tau_k-1)\right\}t\right)}}{1+\frac{a\beta}{b_1}e^{\left(a\beta+\left\{a_n*(t-\tau_n-1)+\sum_{k=1}^{n-1}a_k*(\tau_k-\tau_k-1)\right\}t\right)}}\right]\left(a+\frac{b_1}{\beta}\right)+a-\frac{b_1}{\beta}\right\} \quad (2.15)$$

D. Generalization of our Model with Multiple Change Point:

We can follow similar procedure described to the generalized model with multiple change mends

$$\text{If } b_k(t)=a_k c t^{c-1}$$

We can get the mean value functions

$$m_n(t)=\frac{1}{2}\left\{\left[\frac{1-\frac{a\beta}{b_1}e^{a\beta+\left\{a_n+(t^c-\tau^c_{n-1})+\sum_{k=1}^{n-1}a_k*(\tau_k^c-\tau^c_{k-1})\right\}t^c}}{1-\frac{a\beta}{b_1}e^{a\beta+\left\{a_n+(t^c-\tau^c_{n-1})+\sum_{k=1}^{n-1}a_k*(\tau_k^c-\tau^c_{k-1})\right\}t^c}}\right]\left(a+\frac{b_1}{\beta}\right)+a-\frac{b_1}{\beta}\right\} \quad \ldots(2.16)$$

III. IMPERFECT-DEBUGGING MODELING

In general, different SRGM make different assumptions and therefore can be applied to different situations. From our studies, most SRGMs assume that each time a failure occurs, that fault that caused it is immediately removed and no new faults are introduced sometimes these assumptions can help to reduce the complaints of modeling software reliability[8,12]. As we know that software debugging is the process of identifying the cause for software detective behavior and addressing that problem, there is much this paper can address the problem of imperfect debugging [6, 18]. We plan to incorporate relaxations of some assumptions in order to make the SRGMs more realistic and practical.

A general continues SRGM we let m(t+Δt) be equal to the quasi arithmetic mean of m(t) and a with weights w(t,Δt) and 1-w(t,Δt), then:

$(m(t+\Delta t))= G(m(t,\Delta t)\ g(m(t)+(1-w(t,\Delta t))g(a),0<w(t,\Delta t)<1.$

Where g is real-valued by modifying equations (A) if we let m(t+Δt) be equal to the quasi arithmetic mean of m(t) and n(t) with weights w(t,Δt) and 1-w(t,Δt) then:

$g(m(t+\Delta t)) = w(t,\Delta t)\ g(m(t))+(1-w(t,\Delta t))g(n(t))$

Where n(t) is the fault content functions (n(0)=a) g is a real – valued; strictly monotonic and differentiable function. That is

$$\frac{g(m(t\Delta t))-g(m(t))}{\Delta t}=\frac{1-w(t,\Delta t)}{\Delta t}(g(n(t)-g(m(t))$$

……………(3.1)

Suppose $(1-w(t,\Delta t))/\Delta t \rightarrow b(t)$ as $\Delta t \rightarrow 0$, we get the differential equations.

$$\frac{\partial}{\partial t}(g(m(t))=b(t)\{g(n(t))-g(m(t))\} \quad ..\ (3.2)$$

For example if g(x) =x from (3.2) becomes

$$\frac{\partial}{\partial t}(g(m(t))=b(t)\{n(t)-m(t)\} \quad \ldots\ldots\ldots\ldots\ldots\ldots.(3.3)$$

The following theorem is true

Let $\dfrac{\partial}{\partial t}(g(m(t))=b(t)g(n(t))-g(m(t))$

Where G is a real valued, strictly monotonic, and differentiable function we have

$$m(t)=g^{-1}(g(n(t))+\{g(m(0)-g(n(t))\}\exp^{[(a\beta+b)t]})$$

and

$$B(t)=\int_0^t a(u)du$$

A. Our Model with Multiple Change Points and Imperfect Debugging

We can describe our model with single change point under imperfect debugging as the following

$$\frac{\partial m(t)}{\partial t}=b(t)*(a-m(t))$$

We can get the resulting the model is

$$\frac{\partial m(t)}{\partial t}=b(t)*(n(t)-m(t)) \quad \ldots\ldots\ldots\ldots\ldots\ (3.4)$$

and

$$b(t)=\begin{cases} b_1=a_1(t) & \text{if } 0\le t\le \tau \\ b_2=a_2(t) & \text{if } t>\tau. \end{cases}$$

Here we note that n(t) is a fault content function which is defined as the sum of the expected number of initial software faults are introduce d faults as a function f time 't'.
There are some fault content functions in (3.4), but further discussion of this is beyond the scope of this paper there we assume:

$$\frac{\partial m(t)}{\partial t} = \beta \frac{\partial m(t)}{\partial t}$$ ………………. (3.5)

Where β is the fault introductions rate and n (0) =a.
Therefore eq (2) becomes
N(t)=βm(t)+c
N(0)=βm(0)+c
A=0+c
Therefore n(t)=βm(t)+c

$$\frac{\partial m(t)}{\partial t} = b(t)[a + \beta m(t)]$$ …………….. (3.6)

Solving eq(3.6) and assuming m(0)=0 we obtain the mean value function as follows

$$\frac{\partial m(t)}{\partial t} = b(t)\left[a + \beta m(t) - m(t)\right]$$

$$\frac{\partial m(t)}{\partial t} = b(t)\left[a + m(t)(\beta - 1)\right]$$

$$m(t) = \left\{\frac{1}{2(1-\beta)}\left[\frac{1-\frac{a(1-\beta)}{2}e^{a(1-\beta)+a_1)t}}{1+\frac{a(1-\beta)}{2}e^{a(1-\beta)+a_1)t}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

$$= \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{2}e^{a(1-\beta)+a_1)\tau+a_2(1-\beta)(t-\tau))}}{1+\frac{a(1-\beta)}{2}e^{a(1-\beta)+a_1)\tau+a_2(1-\beta)(t-\tau))}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

……. (3.7)

In fact eq(3.7) we can also solve based on the unified theory for SRGM. Finally when 0≤t≤τ, if we take g(x)=x,

$$k_1 = 1-(1-\beta)\frac{m(0)}{a}$$

From theorem, we can easily obtain the mean value functions

$$m_1(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-a(1-\beta)e^{a(1-\beta)+a_1)t}}{1+a(1-\beta)e^{a(1-\beta)+a_1)t}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

……….. (3.8)

Furthermore, when t>τ,if we take g(x)=x and $B_2(t)= a_2du$ then from theorem we have

$$m_2(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-a(1-\beta)e^{(k_2+a_2(1-\beta)(t-\tau))}}{1+a(1-\beta)e^{(k_2+a_2(1-\beta)(t-\tau))}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

…….. (3.9)

Where $k_2=1-(1-\beta)m_1(\tau)=\exp\,(a(1-\beta)+b_{11}\,\tau$
Also the fault contest function is given by

$$n(t) = n_1(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{2}\beta e^{a(1-\beta)+a_1)t}}{1+\frac{a(1-\beta)}{2}\beta e^{a(1-\beta)+a_2)t}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

……….. (3.10)
When 0≤t≤τ
When τ<t
Finally if

$$k_n = 1-\frac{(1-\beta)m_{n-1}(t_{n-1})}{a} = \exp\left[(1-\beta)\sum_{k=1}^{n-1}(\tau_k - \tau_{k-1})\right]$$

We obtain a generalized solution for describing the model with multiple change points under suspected debugging.

$$m_n(t) = \frac{1}{2(1-\beta)}\left\{\begin{array}{l}\left[\frac{1-\frac{a(1-\beta)}{b_1}e^{(1-\beta)\left\{a+\{a_n*(t-\tau_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau_k-\tau_{k-1}]\right\}t}}{1+\frac{a(1-\beta)}{b_1}e^{(1-\beta)\left\{a+\{a_n*(t-\tau_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau_k-\tau_{k-1}]\right\}t}}\right] \\ \left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\end{array}\right\}$$

and

$$n_n(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{b_1}\beta e^{(1-\beta)\left\{a+\{a_n*(t-\tau_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau_k-\tau_{k-1}]\right\}t}}{1+\frac{a(1-\beta)}{b_1}\beta e^{(1-\beta)\left\{a+\{a_n*(t-\tau_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau_k-\tau_{k-1}]\right\}t}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

………. (3.11)
where τ=0andτ_{n-1}<t.

B. Generalized our Model with Multiple Change-Points Considering Imperfect Debugging

The generalized our model with multiple change-points under imperfect debugging, we have the mean value function:

$$m_n(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{b_1}e^{(1-\beta)\left\{a+\{a_n*(t^c-\tau^c{}_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau^c{}_k-\tau^c{}_{k-1}]\right\}}}{1+\frac{a(1-\beta)}{b_1}e^{(1-\beta)\left\{a+\{a_n*(t^c-\tau^c{}_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau^c{}_k-\tau^c{}_{k-1}]\right\}}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

And

$$n_n(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{b_1}\beta e^{(1-\beta)\left\{a+\{a_n*(t^c-\tau^c{}_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau^c{}_k-\tau^c{}_{k-1}]\right\}}}{1+\frac{a(1-\beta)}{b_1}\beta e^{(1-\beta)\left\{a+\{a_n*(t^c-\tau^c{}_{n-1})\}+\sum_{k=1}^{n-1}a_k[\tau^c{}_k-\tau^c{}_{k-1}]\right\}}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

…… (3.12)
Where r_0 =0 and τ_{n-1}<t.

C. Inflection S-Shaped Model with Multiple Change-Points Considering Imperfect Debugging

The Inflection S-shaped model with multiple change-points under imperfect debugging, we have the mean value function.

$$n(t) = n_2(t) = \frac{1}{2(1-\beta)}\left\{\left[\frac{1-\frac{a(1-\beta)}{2}\beta e^{a(1-\beta)+a_1)\tau+a_2(1-\beta)(t-\tau_1)}}{1+\frac{a(1-\beta)}{2}\beta e^{a(1-\beta)+a_1)\tau+a_2(1-\beta)(t-\tau_1)}}\right]\left(a+\frac{b_1}{1-\beta}\right)+a-\frac{b_1}{1-\beta}\right\}$$

$$m_n(t) = \frac{1}{2(1-\beta)}\left\{\begin{array}{l}\left[\left[\frac{1-\frac{a\beta}{b_1}\psi * e^{(a\beta+b)t+a_n(t-\tau_{n-1})t}}{1+\frac{a\beta}{b_1}\psi * e^{(a\beta+a_1\tau_1+a_2(t-\tau_1))t}}\right](a+\frac{b_1}{\beta})+a-\frac{b_1}{\beta}\right] * \\ \prod_{k=1}^{n-1}\frac{1-\frac{a\beta}{b_1}\psi * e^{(a\beta+a_k\tau_k)t+a_n(t_k-\tau_{n-1})t}}{1+\frac{a\beta}{b_1}\psi * e^{(a\beta+a_k\tau_k)t+a_n(t_k-\tau_{n-1})t}}\end{array}\right\}^{1-\beta}$$

and

$$n_n(t) = \frac{1}{2(1-\beta)}\left\{\begin{array}{l}\left[\left[\frac{1-\frac{a\beta}{b_1}(1-\beta)\psi * e^{(a\beta+b)t+a_n(t-\tau_{n-1})t}}{1+\frac{a\beta}{b_1}(1-\beta)\psi * e^{(a\beta+a_1\tau_1+a_2(t-\tau_1))t}}\right](a+\frac{b_1}{\beta})+a-\frac{b_1}{\beta}\right] \\ * \prod_{k=1}^{n-1}\frac{1-\frac{a\beta}{b_1}\psi * e^{(a\beta+a_k\tau_k)t+a_n(t_k-\tau_{n-1})t}}{1+\frac{a\beta}{b_1}\psi * e^{(a\beta+a_k\tau_k)t+a_n(t_k-\tau_{n-1})t}}\end{array}\right\}^{1-\beta}$$

… (3.13)

Where $\tau_0=0$ and $\tau_{n-1}<t$.

Parameter estimations is of primary importance in software reliability predictions. Once the analytical solutions for m (t) are known for a given model, the parameters in the solutions need to be determined. Parameter estimations is achieved by applying a technique of MLE, the most important and widely used estimation technique. The MLE technique is to estimate the unknown parameters for the software reliability models.

MLE estimates the parameters by following the likelihood functions L is

$$L = \prod_{i=1}^{n} \frac{(m(t_i)-m(t_{i-1})}{(m_i-m_i}$$

……………(3.14)

$$LnL = \sum_{i=1}^{n}(y_i - y_{i-1})\, m[m(t_i) - m(t_{i-1})] - \sum_{i=1}^{n} m\,(m_i - m_{i-1})!$$

Differencing the above equations with respect to each unknown parameter and setting the partial derivations to zero yields a set of nonlinear equations .The maximum likelihood estimations of the parameter can be obtained by solving the set of equations numerically.

The estimate of parameters a and b for specified β using the MLE method can be obtained by solving the following equations simultaneously.

$$a = \frac{n(1+\beta e^{-bt_n})}{(1-e^{bt_n})}$$

……………………..(3.15)

$$\frac{nt_n e^{-bt_n}(1+\beta)\sum_{i=1}^{n}(mi-mi-1)}{(1-e^{-bt_n})(1+\beta e^{-bt_n})} = \frac{n}{\beta} - \sum_{i=1}^{n} t_i + 2\sum_{i=1}^{n}\frac{\beta t_i e^{-bt_{i-t}}}{(1+\beta e^{-bt_i})}$$

……….. (3.16)

D. Numerical Examples

The data used was from the real software project [9].The system was Brazilian electronic switching system.TROPICOR-1500 for 1500 subscribers. Software size was about 300kb written in assembly language .during 81weeks 461 faults were removed. Actually this data has 81 data entries. Entries 31 through 42 were obtained during field trails, and entries 43 through 81 were obtained during system operation [12]. Here we assume that the 31st week and the 43rd week as the first and the second change points respectively. The data shows in table1.

Table I: Data of real software project

Validation		Field trial		Operation	
Time unit	CNF	Time unit	CNF	Time unit	CNF
1	7	31	301	43	356
2	8	32	302	44	367
3	36	33	310	45	373
4	45	34	317	46	373
5	60	35	319	47	378
6	74	36	323	48	381
7	82	37	324	49	383
8	98	38	338	50	384
9	106	39	342	51	384
10	115	40	345	52	387
11	120	41	350	53	387
12	134	42	352	54	387
13	139			55	388
14	142			56	393
15	145			57	398
16	153			58	400
17	157			59	407
18	174			60	413
19	183			61	414
20	196			62	417
21	200			63	419
22	214			64	420
23	223			65	429
24	246			66	440
25	257			67	443
26	277			68	448
27	283			69	454
28	286			70	456
29	292			71	456
30	297			72	456
				73	457
				74	458
				75	459
				76	459
				77	459
				78	460
				79	460
				80	460
				81	461

In order to show quantitative comparisons for long-term predictions, here we use mean square error (MSE) to judge the performance of the model .The MSE can be expressed as.

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

………….. (3.17)

Where $m(t_i)$ is the expected number of faults by time t_i estimated by a model ,mi is the observed number of faults by time t_i. A smaller MST indicates a smaller tilting error and better performance.

E. Performance Analysis

In this paper, the parameter of proposed models can be estimated by using the methods of maximum likelihood estimations (MLE) and least square estimations (12, 25). The estimated parameters are shown in table2 summarizing the comparisons of the results. Here let's suppose that the derivations of mean value function will take change points into considerations and to estimate software reliability growth. [24]Table 4 the MSE of the unmolded with multiple change points using 52% and 100% results of the data are less than the traditional and model using 52% and 100% of the data in table 3.

Finally it is worthwhile to note that by adding more estimate parameters in modeling the phenomenon, the estimates become more difficult as more numerical calculations one involved assume her e additional calculations can be fully automated. Actually if higher reliability is required in same crucial applications the cost of estimate computations for more accuracy is easily justified and is valuable.

Table II: parameter estimations of the our model

	a	b	Ψ
Our Model (37%)	309.89	0.02009	0.9009
(52%)	450.20	0.0100	0.3400
(100%)	400.90	0.0189	0.2900

Table III: comparison results of our model and inflection S-shaped model.

Model	MSE	
	Our model	Inflection S-shaped model
model (37%of data)	590.902	793.02
(52%of data)	390.842	385.42
(100% of data)	77.604	84.57

Table IV: Comparison results of our model with multiple change points and inflection S-shaped model.

Model	MSE	
	Our model	Inflection S-shaped model
Our model (37%)	570.402	793.00
(52%)	400.9003	404.01
(100%)	70.420	86.36

IV. CONCLUSION

Our model use with and without multiple change point gives smaller MSE in most of situations. Therefore our model as a better predictability of software reliability then the inflection s-shaped model consider by chin-yu huang at 2005 .We proposed several software operational reliability growth models with multiple change points. We showed that most existing SRGMs and NHPPs can be improved by incorporating the concepts of multiple change points. We provide a simple but useful approach to measure and access operational software reliability. Our model also discussed under imperfect debugging conditions.

V.REFERENCES

[1] A. L. Goel and K. Okumoto, "An Analysis of Recurrent Software Errors in a Real-Time Control System," *Proceedings of the 1978 Annual Conference (ACM'78)*, pp. 496-501, 1978, Washington, D.C.,USA.

[2] C. Y. Huang, "Performance Analysis of Software Reliability Growth Models with Testing-Effort and Change-Point," *Journal of Systems and Software*, Vol. 76, Issue 2, pp. 181-194, May 2005.

[3] C. Y. Huang, M. R. Lyu, and S. Y. Kuo, "A Unified Scheme of Some Non-Homogenous Poisson Process Models for Software Reliability Estimation," *IEEE Trans. on Software Engineering*, Vol. 29, No. 3, pp. 261-269, March 2003.

[4] C. Y. Huang, S. Y. Kuo, M. R. Lyu, and J. H. Lo, "Quantitative Software Reliability Modeling from Testing to Operation," *Proceedings of the 11th International Symposium on Software Reliability Engineering (ISSRE 2000)*, pp. 72-82, Oct. 2000, San Jose, California, USA.

[5] F. Z. Zou, "A Change-Point Perspective on the Software Failure Process," *Software Testing, Verification and Reliability*, Vol. 13, No. 2, pp. 85-93, June 2003.

[6] H. Pham, *Software Reliability*, Springer-Verlag, 2000.

[7] J. D. Musa, "Sensitivity of Field Failure Intensity to Operational Profile Errors," *Proceedings of the 5th International Symposium on Software Reliability Engineering (ISSRE'94)*, pp. 334-337, Nov. 1994,Monterey, California, USA.

[8] J. D. Musa, *Software Reliability Engineering: More Reliable Software,Faster Development and Testing*, McGraw-Hill, 1999.

[9] K. Kanoun and J. C. Laprie, "Software Reliability Trend Analyses from Theoretical to Practical Considerations," *IEEE Trans. onSoftware Engineering*, Vol. 20, No. 9, pp. 740-747, Sept. 1994.

[10] K. Tokuno and S. Yamada, "Markovian Availability Measurement with Two Types of Software Failures During The Operation Phase," *International Journal of Reliability, Quality and Safety Engineering*, Vol. 6, No. 1, pp. 43-56, March 1999.

[11] M. J. Kwang, "An Adaptive Failure Rate Change-Point Model for Software Reliability," *International Journal of Reliability and Applications*, Vol. 2, No. 3, pp. 99-207, 2001.

[12] M. R. Lyu, *Handbook of Software Reliability Engineering*, McGrawHill, 1996.

[13] M. Zhao, "Change-Point Problems in Software and HardwareReliability," *Communications in Statistics– Theory and Methods,* Vol. 22, No. 3, pp. 757-768, 1993.

[14] N. Karunanithi, D. Whitley, and Y. K. Malaiya, "Prediction of Software Reliability Using Connectionist Models," *IEEE Trans. on Software Engineering*, Vol. 18, No. 7, pp. 563-574, July 1992.

[15] P. K. Kapur, A. K. Bardhan, and N. L. Butani, "Modelling Failure Phenomenon of A Commercial Software in

Operational Use,” *Proceedings of the 2001 International Conference on Quality,Reliability and Control*, Chief Editor: A. K. Verma, IETE and IIT: Mumbai, R-59.

[16] Q. Kenney, “Estimating Defects in Commercial Software during Operational Use,” *IEEE Trans. on Reliability*, Vol. 42, No. 1, pp. 107-115, March 1993.

[17] R. Chillarege, R. K. Iyer, J. C. Laprie, and J. D. Musa, “Field Failures and Reliability in Operation,” *Proceedings of the 4th International Symposium on Software Reliability Engineering (ISSRE'93)*, pp.122-126, Nov. 1993, Denver, Colorado, USA.

[18] S. Yamada, K. Tokuno, and S. Osaki, “Software Reliability Measurement in Imperfect Debugging Environment and Its Application,” *Reliability Engineering & System Safety*, Vol. 40, No. 2, pp. 139-147, 1993.

[19] S. Yamada, “Software Reliability Measurement during Operation Phase and Its Application,” *Journal of Computer and Software Engineering*, Vol. 1, No. 4, pp. 389-402, 1993

[20] S. Yamada, Y. Tanio, and S. Osaki, “A Software Reliability Evaluation Method during Operation Phase,” *The Transactions of IEICE*, Vol. J72-D-I, No. 11, pp. 797-803, 1989.

[21] T. Philip, P. N. Marinos, and K. S. Trivedi, “A Multiphase Software Reliability Model: From Testing To Operational Phase,” *Technical Report (TR-96-01),* Center for Advanced Computing and Communication, Duke University, January 1996.

[22] Y. Chen, “Modeling Software Operational Reliability via Input Domain-Based Reliability Growth Model,” *Proceedings of the 28th International Symposium on Fault-Tolerant Computing (FTCS '98)*, pp. 314-323, June 1998, Munich, Germany.

[23] Y. K. Malaiya, N. Karunanithi, and P. Verma, “Predictability Measuresfor Software Reliability Models,” *Proceedings of the 14th Annual International Computer Software and Applications Conference (COMPSAC’90)*, pp. 7-12, Oct. 1990, Chicago, Illinois, USA.

[24] Y. Huang, “Reliability prediction and assessment of fielded software based on multiple change-point models,” Dependable Computing, 2005. Proceedings. 11th Pacific Rim International Symposium on, Dec 2005.

[25] Z. Wang and J. Wang, “Parameter Estimation of Some NHPP Software Reliability Models with Change-Point,” *Communications in Statistics: Simulation & Computation*, Vol. 34, No. 1, pp. 121-134, Feb, 2005.

Authors:

Dr. R. Satya Prasad Received Ph.D. degree in Computer Science in the faculty of Engineering in 2007 from Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. He have a satisfactory consistent academic track of record and received gold medal from Acharya Nagarjuna University for his outstanding performance in a first rank in Masters Degree. He is currently working as Associative Professor and Head of the Department, in the Department of Computer Science & Engineering, Acharya Nagarjuna University. He has occupied various academic responsibilities like practical examiner, project adjudicator, external member of board of examiners for various Universities and Colleges in and around in Andhra Pradesh. His current research is focused on Software Engineering, Image Processing & Database Management System. He has published several papers in National & International Journals.

O.NagaRaju received the Masters Degree in Computer Science & Engineering from Acharya Nagarjuna University, Guntur, and Andhra Pradesh, India. He is currently working as Assistant Professor in the Department of Computer Science & Engineering. He is currently pursuing Ph.D., at Department of Computer Science and Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, India. His research interests include Software Engineering, Network Computing, and Image Processing. He has published several papers in National & International Journals.

R.R.L. Kantam is a professor of statistics at Acharya Nagarjuna University, Guntur-India. He has 31 years of teaching experience in statistics at Under Graduate and Post Graduate programs. As researcher in Statistic, he has successfully guided 8 students for M.Phil in statistics and 5 students for Ph.D. in statistics. He has authored 54 research publications appeared various statistics journals published in India and other countries like US, UK, Germany, Pakistan, Srilanka and Bangladesh. He has been a referee for Journal of Applied Statistics (U.K.), METRON (Italy), Pakistan Journal of Statistics (Pakistan), IAPQR – Transactions (India), Assam Statistical Review (India) and Gujarat Statistical Review (India). He has been a special speaker in technical sessions of a number of Seminars and Conferences. His areas of research interest are Statistical Inference, Reliability Studies, Quality Control Methods and Actuarial Statistics. As a teacher his present teaching areas are Probability Theory, Reliability, and Actuarial Statistics. His earlier teaching topics include Statistical Inference, Mathematical Analysis, Operations Research, Econometrics, Statistical Quality Control, Measure theory.