Open Source Software Reliability Growth Model by Considering Change – Point

V. B. Singh[1], **P. K. Kapur**[2] and **Mashaallah Basirzadeh**[3]

Submitted in April 2010; Accepted in November 2011

Abstract – The modeling technique for Software Reliability is reaching its prosperity. Software reliability growth models have been used extensively for closed source software. The design and development of open source software (OSS) is different from closed source software. We observed some basic characteristics for open source software like (i) more instructions execution and code coverage taking place with respect to time, (ii) release early, release often (iii) frequent addition of patches (iv) heterogeneity in fault density and effort expenditure (v) Frequent release activities seem to have changed the bug dynamics significantly (vi) Bug reporting on bug tracking system drastically increases and decreases. Due to this reason bug reported on bug tracking system keeps an irregular state and fluctuations. Therefore, fault detection/removal process can not be smooth and may be changed at some time point called change-point. In this paper, an instructions executed dependent software reliability growth model has been developed by considering change-point in order to cater diverse and huge user profile, irregular state of bug tracking system and heterogeneity in fault distribution. We have analyzed actual software failure count data to show numerical examples of software reliability assessment for the OSS. We also compare our model with the conventional in terms of goodness-of-fit for actual data. We have shown that the proposed model can assist improvement of quality for OSS systems developed under the open source project.

Index Terms - Open source software, reliability assessment, software reliability growth model, bug tracking system, change-point

1. INTRODUCTION

The advancement in the information technology has changed the dynamics of life and society as well as software development. It has added new dimensions like e-learning, e-conferencing, e- commerce, e-meeting e-governance…, and the list is now becoming endless. Since the mid 1990s, there has been a surge of interest among academics and practitioner in open source software. The design and development of open source software is significantly different from that of proprietary software. Open source software is developed by community for community.The development of OSS is of interdisciplinary nature and needs knowledge and expertise from many scientific disciplines such as computer science, management and organization, social sciences, law, economics and psychology. In this paper, we measure the reliability growth of OSS quantitatively by measuring the remaining number of bugs in the software. The rest of the paper is organized as follows. Section A and B of introduction deals with literature review of OSS and change point problem in software reliability. In section 2I, we discuss modeling framework comprising notations, assumption and model development. Section 3 deals with model validation, numerical illustration and goodness of fit curves. Finally, section 4 deals with concluding remarks and future direction.

[1]*Delhi College of Arts & Commerce, University of Delhi, Delhi*
[2]*Department of Operational Research, University of Delhi, Delhi*
[3] *Department of Operational Research, University of Delhi, Delhi*
E-mail: [1]singh_vb@rediffmail.com, [2]pkkapur1@gmail.com and [3]m_basir31@yahoo.com

1.1 Open Source Software with Reliability

The use of open source software is increasing rapidly and its role is becoming high in different domains ranging from commercial, educational, to research. According to Gartner's report, about 80 percent of all commercial software will include elements of open source technology 2012 [24]. Open source was first evolved during 1970s. Richard Stallman, an American software developer, who believes that sharing source code and ideas is fundamental to freedom of speech, developed a free version of the widely used Unix operating system under GNU [5 and 25].The spirit of open source software is the free right of using, reproducing, distributing and modifying the software, which creates an efficient economical, productive software development model: establishing commercial projects through the concept of open source, implementing collaborative development through the open source community based on the network, allocating resources optimizedly, increasing the transparency of projects, and reducing the risk of development [6]. Eric Raymond, the main proponent and co-founder of the open source project, is generally credited with establishing the movement of OS through his seminal paper "The cathedral and the Bazar" [7] and attributed the open-source software development approach as:

"Given enough eyeballs, all bugs are shallow." (p. 41)

A classification of users and developers and their role as shown in figure 1 has been discussed in [17].

The author in [18] discussed many Claims and counterclaims for open source software on the basis of number of factors including cost advantage, source code availability, maturity, vendor lock-in and external support. In the available literature, many papers address the issue of reliability for open source software qualitatively. Paper [19] proposes a number of hypotheses and tries to analyze the relationship between openness and reliability. A study has also been carried out on bug report data of open source project and it has been concluded that traditional software reliability growth models can not apply to assess the reliability growth of open source software because the design and development of open source is different from that of closed source [1].

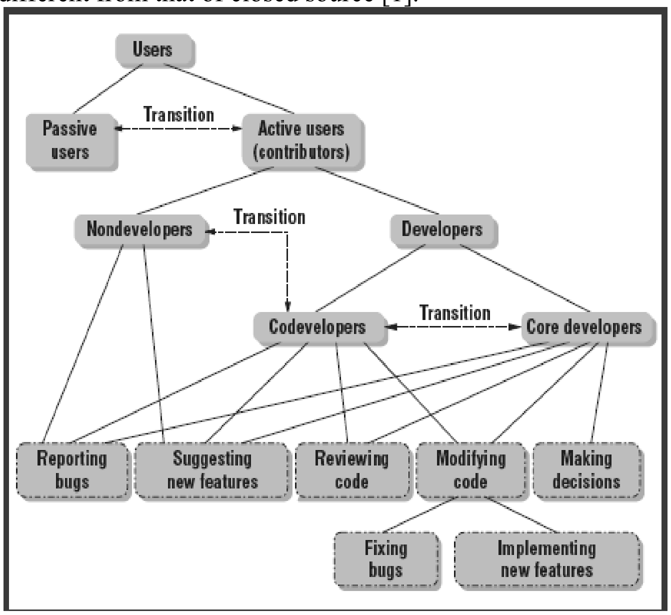


Figure 1[17]

A software reliability assessment method concerned with the software development environment of OSS has been discussed in [4]. It proposes software reliability assessment and optimization analysis method for OSS paradigm. Reliability growth models have been presented by considering user growth for open source software [23]. The paper also reveals that reliability growth curve of open source software is similar with that of closed source software by studying bug reported data from bug tracking system of software projects developed under open source environment.

1.2 Change Point Problem in Software Reliability

The fault detection rate may not be smooth and can be changed at some time moment τ due to changes in defect density, testing strategies etc. called change-point. Many researchers have incorporated change point in software reliability growth modeling. Many researchers have incorporated change point in software reliability growth modeling for closed source software. Firstly, Zaho [20] incorporated change-point in software and hardware reliability. Huang et al. [14] used change-point in software reliability growth modeling with testing effort functions. The change-point problem in OSS has been introduced by Singh et al. [15]. Kapur et al. [2, 13] introduced various testing effort functions and testing effort control incorporating change-point in software reliability growth modeling. Kapur et al. [10,11] proposed a software reliability growth model for errors of different severity using change-point. The multiple change-points in software reliability growth modeling for a fielded has been proposed by Kapur et al. [9]. Later on SRGM based on stochastic differential equations incorporating change-point concept has been proposed by Kapur et al. [12].

2. MODEL DESCRIPTION

During middle and operational stage fault detection rate normally depends on other parameters such as execution rate of CPU instruction, code expansion or code coverage [21]. The success of OSS projects has been mostly attributed to the speed of development, reliability, portability and scalability of the resulting software. Recently, instructions executed dependent models have been proposed to measure reliability growth of open source software [23]. In this paper, we are considering number of instructions executed dependent software reliability growth model by considering change point for measuring the reliability growth for open source software..

(i) Notations

$m, m(t)$: Expected number of faults identified in the time interval $(0,t]$

$e, e(t)$: Expected number of instructions executed on the software in the time interval $(0,t]$

a: Constant, representing the number of faults lying dormant in the software.

k, β : Constants

$b(t)$: Fault removal rate as a function of testing time

(ii) Assumptions

Mathematical model, which can capture various types of growth patterns as the testing/debugging progresses, are proposed in this paper.

The proposed model is based upon the following basic assumptions.

1. Software failure phenomenon can be described by the Non-homogeneous Poisson Process (NHPP).
2. The number of failures during testing is dependent upon the number of instructions executed.
3. The number of instructions executed is a power function of testing time.
4. The fault detection rate may be change at some time moment (called change-point).

(iii) Modeling Framework

Using the above assumptions, the failure phenomenon can be described with respect to time as follows in [8]

$$\frac{dm(t)}{dt} = \frac{dm(t)}{de(t)}\frac{de(t)}{dt} \tag{1}$$

Let us consider the case when the rate at which failures occur depends not only upon the number of faults remaining in the software but also on the proportion of faults already detected. Based on this assumption the differential equation for fault identification / removal can be written as:

$$\frac{dm(t)}{de(t)} = \left[k_1 + k_2\frac{m(t)}{a}\right]\left(a - m(t)\right) \tag{2}$$

Here k_1 is the rate at which residual faults cause failure. It is a constant as each one of these faults has an equal probability of causing failure. k_2 is the rate at which additional faults are identified without their causing any failure.
Let the second component of expression (1) be defined as a power function of testing time i.e.

$$\frac{de(t)}{dt} = k_3 t^K \qquad (3)$$

Substituting (2) and (3) in (1) we have:

$$\frac{dm(t)}{dt} = k_3 t^k \left(k_1 + k_2 \frac{m(t)}{a}\right)\left(a - m(t)\right) \qquad (4)$$

It is a first order differential equation. Solving it with the initial condition $m(0) = 0$ we get:

$$m(t) = a\left(\frac{1-\exp\left(-\frac{1}{k+1} b t^{k+1}\right)}{1+\beta\exp\left(-\frac{1}{k+1} b t^{k+1}\right)}\right) \qquad (5)$$

Here $b = k_3(k_1 + k_2)$ and $\beta = (k_2 / k_1)$

If we take $k = 0$ in equation(5), above model reduces to [3].

(iv) Proposed Model by Considering Change-Point

We can write differential equation for fault detection process i.e.

$$\frac{dm(t)}{dt} = b(t)\left(a - m(t)\right) \qquad (6)$$

and if we take $b(t) = \dfrac{bt^k}{1+\beta\exp\left(-\frac{1}{k+1} bt^{k+1}\right)}$ i.e a power logistic function as follows in [26] and solving for equation (6), we get same solution as given in equation (5)

Now by considering change in fault detection rate at change point τ, we can write

$$b(t) = \begin{cases} \dfrac{b_1 t^k}{1+\beta\exp\left(-\frac{1}{k+1} b_1 t^{k+1}\right)} & for\ t \le \tau \\ \dfrac{b_2 t^k}{1+\beta\exp\left(-\frac{1}{k+1} b_2 t^{k+1}\right)} & for\ t > \tau \end{cases} \qquad (7)$$

Here, b_1 and b_2 are fault detection rates before and after change point.

The fault detection equation can be written as

$$\frac{dm(t)}{dt} = \begin{cases} \dfrac{b_1 t^k}{1+\beta\exp\left(-\frac{1}{k+1} b_1 t^{k+1}\right)}\left(a - m(t)\right) & for\ t \le \tau \\ \dfrac{b_2 t^k}{1+\beta\exp\left(-\frac{1}{k+1} b_2 t^{k+1}\right)}\left(a - m(t)\right) & for\ t > \tau \end{cases} \qquad (8)$$

After solving equation (8), we get

$$m(t) = \begin{cases} a\left(1 - \left(\dfrac{1+\beta}{1+\beta\exp\left(-\frac{1}{k+1} b_1 t^{k+1}\right)}\right)\exp\left(-\frac{1}{k+1} b_1 t^{k+1}\right)\right) & for\ t \le \tau \\ a\left(1 - \left(\dfrac{1+\beta}{1+\beta\exp\left(-\frac{1}{k+1} b_1 \tau^{k+1}\right)}\right)\left(\dfrac{1+\beta\exp\left(-\frac{1}{k+1} b_2 \tau^{k+1}\right)}{1+\beta\exp\left(-\frac{1}{k+1} b_2 t^{k+1}\right)}\right) \times \exp\left(-\frac{1}{k+1}\left(b_1 \tau^{k+1} + b_2\left(t^{k+1} - \tau^{k+1}\right)\right)\right)\right) & for\ t > \tau \end{cases}$$

(9)

if we take k= 0, and β =0 , model reduces to [13] and [2] respectively

3. MODEL VALIDATION

To illustrate the estimation procedure and application of the SRGM (existing as well as proposed) we have carried out the data analysis of real software data set.

3.1 Description of Datasets

Data set 1(DS-1)

We collected all failure data of Keepass software developed under open source environment (www.sourceforge.net) from 19-Dec-03 to 27-Feb-07, 458 failures were observed. Keepass software is a password database utility. Users can keep their passwords securely encrypted on their computers. A single Safe Combination unlocks them all. From graphical view of data, we identify 19th month as change-point.

Data set 2(DS-II)

This data is cited from Fedora Core Linux (http://fedora.redhat.com/ and [4]), which is one of the operating system developed under an open source project. We have taken data up to release 3 for model validation. During the course of 57 days 164 failure were observed. From graphical view of data, we identify 17th month as change-point.

3.2 Comparison Criteria

The performance of SRGM are judged by their ability to fit the past software fault data (goodness of fit) and predicting the future behavior of the fault.

The Mean Square -Error (MSE)

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

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

where k is the number of observations. The lower MSE indicates less fitting error, thus better goodness of fit [16].

Coefficient of Multiple Determinations (R^2)

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

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

R^2 measures the percentage of the total variation about the mean accounted for the fitted curve. It ranges in value from 0 to 1. Small values indicate that the model does not fit the data well. The larger R^2, the better the model explains the variation in the data [16].

Bias

The difference between the observation and prediction of number of failures at any instant of time i is known as PEi.(prediction error). The average of PEs is known as bias. Lower the value of Bias better is the goodness of fit [22].

Variation

The standard deviation of prediction error is known as variation.

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

Lower the value of Variation better is the goodness of fit [22].

Root Mean Square Prediction Error

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

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

Lower the value of Root Mean Square Prediction Error better is the goodness of fit [22].

5. NUMERICAL RESULTS AND ANALYSIS

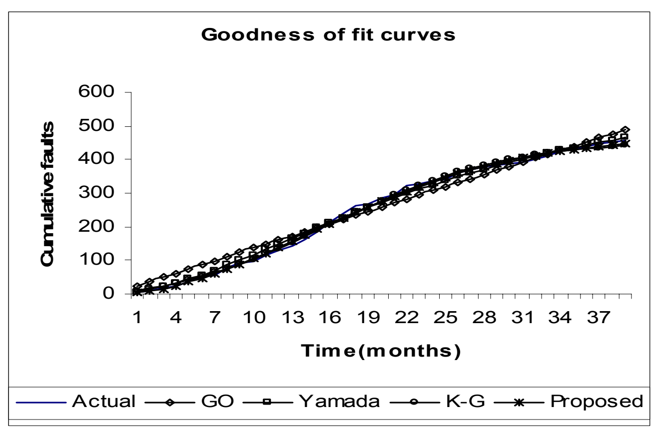
The parameter estimation and comparison criteria results for DS-1and DS-II of all the models under consideration can be viewed through Table I(a-b) and Table II(a-b) respectively. It is clear from the table that proposed model (equation 9) provides better goodness of fit for DS-1 and DS-II.The proposed model gives total failure latent in software i.e. 467 against observed i.e. 458 failures, means 9 bugs are still remaining in software and 181 against observed i.e. 164 failures means 17 bugs are still remaining in software (a fairly reasonable estimate)For DS-I and DS-II. It has been also observed that GO model overestimates the value of parameter “a”.

Models	Parameter Estimates				
	A	b_1 /b	b_2	K	β
GO [27]	12515	.001	--	--	--
Yamada[28]	577	.075	--	--	--
KG[3]	457	.150			13
Proposed model Equation[9]	467	.0054	.0055	.8958	.0619

Models	Comparison Results				
	R^2	MSE	Bias	Variation	RMSPE
GO [27]	.99047	24.32	-0.95	5.25	5.333
Yamada[28]	.99353	16.51	-0.76	4.04	4.111
KG[3]	.99669	8.43	-0.35	2.92	2.938
Proposed model Equation[9]	.99685	8.04	0	2.86	2.861

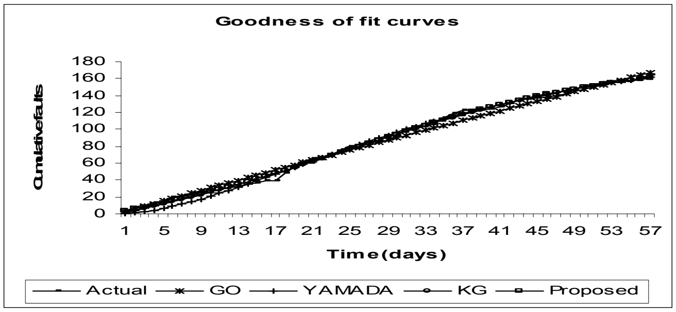
Table I (a-b): Model Parameter Estimation and comparison Results DS-1)

For DS-I



Goodness of fit Curves

For DS-II



Models	Parameter Estimates				
	A	b_1 /b	b_2	k	β
GO [27]	1548	.002	-	-	-
Yamada[28]	196	.055			
KG[3]	185	.064	-	-	4.637
Proposed model Equation[9]	181	.072	.071	0	5.656

Models	Comparison Results				
	R^2	MSE	Bias	Variation	RMSPE
GO [27]	.97167	668.18	6.05	25.449	26.16
Yamada[28]	.99452	129.26	1.88	11.357	11.51
KG[3]	.99650	82.34	0.99	9.1334	9.188
Proposed model Equation[9]	.99719	66.30	-0.46	8.233	8.246

:

Table II (a-b): Model Parameter Estimation and comparison Results DS-2)

Bug Frequencies of OSS Projects (non cumulative bugs)

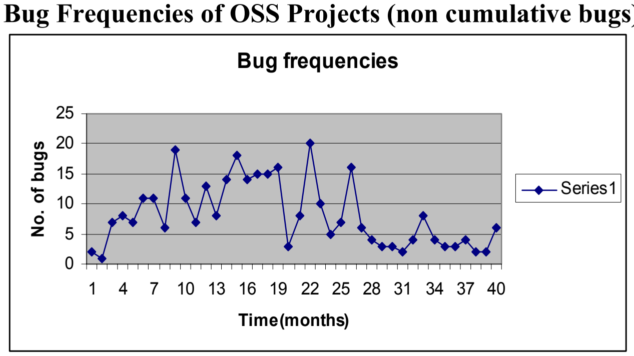


Figure2: Data set 1(DS1)

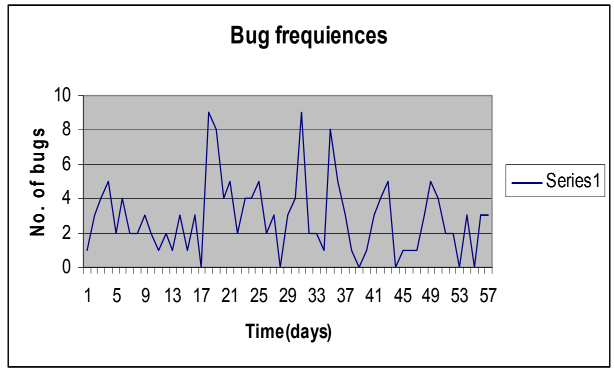


Figure3: Data set 1(DS-2) For DS-*2*

6. CONCLUSION

In this paper, we have proposed a software reliability model by considering huge user growth in case of open source software. The user growth is expressed in terms of number of instructions executed. The proposed model also incorporates change in fault detection rate due to drastic changes in reported bug on bug tracking system. Parameter estimates, comparison criteria results and goodness of fit curve has been also presented in comparison with conventional models. But, there is a need to present the model in a form that is friendly to the software developers.

In future, we will try to develop a general framework to measure reliability growth of open source software by considering detection and correction process (bug reporting and bug fixing).

ACKNOWLEDGEMENT:

First author acknowledges with thanks the financial support provided by University Grants Commission, India under the project No.F.8-1(77)2010(MRP/NRCB).

REFERENCES

[1]. Fengzhong Zou, Joseph Davis "Analyzing and Modeling Open Source Software Bug Report Data" 19th Australian Conference on Software, IEEE computer Society, DOI 10.1109/ASWEC.2008.13.

[2]. P.K.Kapur, V.B.Singh, Sameer Anand and V.S.S. Yadavalli, "Software Reliability Growth Model with Change-Point and Effort Control Using a Power Function of Testing Time" International Journal of Production Research, Vol. 46 Issue no. 3 pp.771-787 , 2008.

[3]. Kapur P.K. and Garg R.B. "A software reliability growth model for an error removal phenomenon", Software Engineering Journal, Vol. 7, pp. 291-294, 1992

[4]. Tamura Y. and Yamada S., "Optimization analysis for Reliability Assessment based on stochastic differential equation modeling for Open Source Software" International Journal of Systems Science, Vol. 40, No.4 ,pp 429-438, 2009.

[5]. www.parliamnet.uk/parliamentry_ofice/post/pubs2005.cfm
[6]. Xu Hao, Zhang Zhengang, Lin Chunpei and Ding Zhuo "The Study on Innovation Mechnasim of Open Source Softawre Community", 4th InternationalConference on Wireless Communications, Networking and Mobile Computing, pp. 1-5, 2008.
[7]. Eric S. Raymond, "The Cathedral and the Bazaar: Musings on Open Source by an Accidental Revolutionary.O'Reilly& Associates,1999,http://www.oreilly.com/catalog
[8]. Kapur,PK ,Yadavalli,VSS,Gupta, Anu" Software reliability growth modeling using power function of testing time" International Journal of Operations and Quantitative Management"Vol.12 No. 2.pp.127-140, June 2006.
[9]. P.K.Kapur, V.B.Singh and Sameer Anand, "Software Reliability Growth Model of Fielded Software Based on Multiple Change-Point Concept Using a Power Function of Testing Time", Quality Reliability and Infocom Technology, (Eds. P.K.Kapur and A.K.Verma), MacMillan India Ltd., pp.171-178,2007
[10]. Goswami D.N., Khatri Sunil K, .and Kapur Reecha "Discrete Software Reliability Growth Modeling for Errors of Different Severity incorporating Change-Point Concept" International Journal of Automation and Computing Vol. 4(4) pp.396-405,2007
[11]. Kapur P.K. Kumar Archana ,Yadav Kalpana and Khatri Sunil "Software Reliability Growth Modelling for Errors of Different Severity using Change Point" International Journal of Quality ,Reliability and Safety Engineering ,Vol.14,No.4, pp 311-326,2007
[12]. P.K.Kapur, V.B.Singh, and Sameer Anand "Effect of Change-Point on Software Reliability Growth Models Using Stochastic Differential Equations" published in the proceedings of 3rd International Conference on Reliability and Safety Engineering, (Eds. R.B.Misra, V.N.A. Naikan, S.K.Chaturvedi, and N.K.Goyal), (INCRESE-2007), Udaipur, pp. 320-333,2007
[13]. Kapur P.K, Gupta A., Shatnawi O, and Yadavalli V.S.S **"**Testing Effort Control Using Flexible Software Reliability Growth Model with Change Point" International Journal of Performability Engineering- Special issue on Dependability of Software/ Computing Systems, Vol. 2, No. 3, pp 245-263,2006
[14]. Huang Yu Chin "Performance Analysis of Software Reliability Growth Models with Testing Effort and Change-Point ", Journal of Systems and Software Vol. 76, Issue 2, pp 181-194,2005
[15]. Singh V.B., Kapur P.K.and Kumar Ravi " Developing S-shaped Reliability Growth Model for Open Source Software by Considering Change-Point." Published in the proceedings of IASTED International Conference in Software Engineering, held during Feb. 16-18,2010, Innsbruck, Austria (Acta press).
[16]. Kapur, P.K., Garg, R.B. and Kumar, S., Contributions to hardware and software reliability, World Scientific, Singapore, 1999
[17]. Cristina Gacek and Budi Arief "The Many meanings of Open Source", IEEE Software, Vol. 21 issue 1, pp.34-40, 2004.
[18]. Kris Ven, Jan Verelst and Herwing Mannaert "Should you adopt open source software?", IEEE Software, Vol. 25, issue 3,pp.54-59,2008
[19]. Ruben van Wendel de Joode and Mark de Bruijne "The organization of open source communities: Towards a Framework to Analyze the relationship between openness and reliability", Proceedings of 39th Hawaii International Conference on System Sciences, pp.1-6, 2006.
[20]. Zhao, M. "Change-Point Problems In Software And Hardware Reliability", Communications in Statistics-Theory and Methods, 22(3), pp. 757-768, 1993.
[21]. Y. Huang, S. Y. Kuo, M. R. Lyu, and J. H. Lo, "Quantitative Software Reliability Modelingfrom Testing to Operation," Proceedings of the 11th IEEE International Symposium onSoftware Reliability Engineering (ISSRE 2000), pp. 72-82, Oct. 2000, San Jose, California.
[22]. K. Pillai and V.S.S. Nair, "A Model for Software Development effort and Cost Estimation", IEEE Transactions on Software Engineering; vol. 23(8), pp. 485-497, 1997.
[23]. Singh V.B. and Kapur P.K."Measuring Reliability Growth of Open Source Software," poster presented in IBM-Indian Research Laboratory Collaborative Academia Research Exchange, held during October 26, 2009 at IBM India Research, New Delhi, India.
[24]. http://arstechnica.com/news.ars/post/20080205-gartner-80-percent-of-commercial-software-programs-will-inlude-open-source-by-2012.html
[25]. Stallman, Richard, The GNU Project, http://www.gnu.org/gnu/thegnuproject.html, 1998.
[26]. Kapur P.K "Flexible Software Reliability Growth Models" ,Reliability Safety and Hazard, Narosa Publication ,pp.155-165,2006.
[27]. Goel AL, Okumoto K. "Time Dependent Error Detection Rate Model For Software Reliability and Other Performance Measures" IEEE Transactions on Reliability, R-28(3): pp. 206-211, 1979.
[28]. Yamada ,S., Ohba M. and Osaki S. " S-shaped Software Reliability Growth Models and their Applications", IEEE Transactions on Reliability R-33,PP. 169-175, 1984