MARKOVIAN SOFTWARE RELIABILITY MODEL FOR TWO TYPES OF FAILURES WITH IMPERFECT DEBUGGING RATE AND GENERATION OF ERRORS

M. Jain

Department of Mathematics, IIT, Roorkee- 247667 (India)
madhufma@iitr.ernet.in, drmadhujain@yahoo.co.in

S. C. Agrawal, P. Agarwal*

School of Basic and Applied Sciences, Shobhit University, Meerut-250001, (India)
scagrawal7@rediffmail.com, priyanka1354@gmail.com

*Corresponding Author

(Received: November 25, 2010 – Accepted in Revised Form: January 19, 2012)

doi: *10.5829/idosi.ije.2012.25.02a.07*

Abstract This investigation deals with a software reliability model based on Markov process. For formulating the model, we define a random variable representing the cumulative number of faults successfully corrected upto a specified point of time. This model is based on the assumption that there are two types of software failures. Further the concepts of imperfect debugging environment and error generation phenomenon are taken into consideration. Transient analysis based on Laplace transform and matrix approach has been done to find the solution of the system of differential difference equations. Several performance indices for software reliability assessment are derived for this model. Numerical results with the help of Runge-Kutta Method show that the proposed framework incorporating both concepts of imperfect debugging phenomenon and error generation for two types of faults has a fairly accurate prediction capability.

Keywords Software reliability; Imperfect debugging; Error generation; Markov process; Software reliability growth.

چکیده این مقاله به بررسی قابلیت یک مدل نرم افزاری بر اساس فرایند مارکوف می‌پردازد. برای فرموله کردن روش، یک متغیر تصادفی به نمایندگی از تعداد کل خطاهایی که با موفقیت در یک نقطه مشخص از زمان تصحیح شده است تعریف کنیم. این مدل بر اساس این فرض می‌باشد: دو نوع از شکست نرم افزار وجود دارد. علاوه بر این مفاهیم محیط اشکال زدایی ناقص و پدیده ایجاد خطا در نظر گرفته شده است. آنالیز گذرا بر اساس تبدیل لاپلاس و روش ماتریس برای پیدا کردن راه حل سیستم تفاضل معادلات دیفرانسیلی انجام شده است. تعدادی از شاخص های عملکردی برای ارزیابی قابلیت اطمینان نرم افزار برای این مدل مشتق شده است. نتایج عددی با کمک روش رانگه–کوتا نشان می دهد که چارچوب پیشنهادی برای ترکیب هر دو مفهوم پدیده اشکال زدایی ناقص و تولید خطا برای دو نوع خطا دارای قابلیت پیش بینی نسبتا دقیق‌تری است.

1. INTRODUCTION

Software engineers generally need a period of time to read, and analyze the collected software failure data. Software reliability models based on stochastic process have gained wide acceptance in the software industry because they are useful engineering tools to analyze and to correct the faults. Software reliability estimates are generally made by building probability models of data collected during testing. The extent to which software reliability is influenced by factors that can be controlled by the software developer is not obvious. We seek a more fundamental approach that directly relates influence factors to the stochastic model's parameters. Software reliability growth models have high validity and usefulness for software development or operation phase. Markov models have been proved to be very useful in many practical applications and there are several

related models available in the literature related to SRGMs. For the first time Jelinski and Moranda [11] introduced such a model. Later in the field of software engineering similar models were attempted by Musa [17] and Shooman [20] and many others. Tokuno and Yamada [22] discussed the modeling of markovian software availability in their exhaustive research article. Non-Homogeneous markov reward model for a multi-state system reliability assessment with different assumptions was developed by Whittaker et al. [27], Lisnianski [13], Lisnianski et al. [15], Lisnianski and Frenkel [14]. Hamlet et al. [8] studied component based software reliability theory. Gokhale and Trivedi [5] discussed analytical models for architecture-based software reliability prediction. Dohi et al. [3] developed software reliability assessment models based on cumulative Bernoulli trial processes. Prowell and Poore [18]] computed system reliability using markov chain usage models. Chung and Ortega [2] analyzed the error tolerant motion estimation. Ravishanker et al. [19] studied NHPP models with markov switching for software reliability. Jalote et al. [10] discussed post-release reliability growth in software products. Huang and Hung [9] analyzed software reliability assessment using queueing models with multiple change-points. Recently in [4] Dulz et al. gave a polyhedron approach to calculate probability distributions for markov chain usage models.

In the case of software reliability, two major issues confound to estimate software reliability, i.e. imperfect debugging and error generation. This concept in a software reliability modeling is a controversial issue. No real world software company possesses infinite resources to test and correct every software fault in the real world. However, a Markov model approach in this regard may be worthwhile and useful to deal with such a situation.

There are several reasons for using NHPP models or hazard rate models in a specific situation. Generally, models are developed for the analysis of failure data collected during the testing stage then we develop our model analytically with the help of NHPP models or hazard rate models. These group of models provides an analytical framework for describing the software failure phenomenon during testing phase and with the help of these group of models we can estimate the future behavior of the software system.

One common assumption of conventional software reliability modeling is that the detected faults are immediately removed; but this assumption may not be realistic in actual software development. In reality, most latent faults may remain uncorrected for a very long time, even after they are detected by professional testers, which increase their impact. Kapur et al. [12] obtained the transient solution of a software reliability model with imperfect debugging and error generation. Yamada et al. [28] did the software reliability measurement in imperfect debugging environment and also discussed its application. Gokhale et al. [6] developed a non-homogeneous markov software reliability model with imperfect repair. Sridharan and Jayashree [21] considered a transient solution of a software model with imperfect debugging and generation of errors by two servers. Tokuno and Yamada [23]] studied a markovian software reliability model with a decreasing perfect debugging rate. Tokuno and Yamada [24] established the imperfect debugging model with two types of hazard rates for software reliability measurement and assessment. Tokuno and Yamada [25] suggested the markovian software reliability measurement with geometrically decreasing perfect debugging rate. Tokuno and Yamada [26] proposed the relationship between software availability measurement and the number of restorations with imperfect debugging. Gupta and Singh [7] estimated software reliability by sequential testing with simulated annealing of mean field approximation. Mathematical modeling of software reliability testing with imperfect debugging was discussed by Cai et al. [1]. Meedeniya et al. [16] derived reliability deployment optimization technique for embedded systems.

Most specifically, we develop a broader class of SRGMs for two types of faults at each discrete time point with imperfect debugging and error generation behavior. This paper is organized as follows: Section 2 provides the description of Markov model along with assumptions and notations. In section 2.1 and 2.2, we construct the governing equations of the proposed model and specifically for an illustration, respectively. Solution approach including Laplace transform and matrix method is described in section 3. Section 4 is devoted to the performance measures. Numerical

results are obtained in section 5 with the help of R-K and matrix method. To explore the effects of different parameter, the sensitivity analysis is also carried out. Finally conclusions are drawn in section 6.

2. MODEL DESCRIPTION

A stochastic model is sought that represents the injection (due to the occurrence of development and debugging errors) and removal (due to successful repairs) of software. The stochastic behavior of the fault correction phenomenon with imperfect debugging is described by a Markov process. In this investigation, we assume that there exist two types of faults in the software such as

(i) Due to originally latent in the system before the testing.
(ii) Due to generated during the testing phase and the error generation phenomenon never leads the software to having infinite errors.

We develop a software model by making the following assumptions:

- The software has a finite number of two types of faults; there are m faults of type I whereas n faults of type II.
- The probability that two or more software failures occur simultaneously is negligible.
- The failure rate is proportional to the number of fault remaining in the software.
- The debugging process is performed as soon as the software failure occurs.
- When the debugging process is performed, at most one fault is corrected and the fault correction time is considered negligible.
- The maximum number of faults in the software never exceeds a finite limit, i.e., m<M, n<N.
- When a failure occurs, an instantaneous repair effort starts and the following cases arise;

(i). The fault content function is reduced by one for type I faults with probability p_0.
(ii). The fault content function remains unchanged for type I faults with probability p_1.
(iii). The fault content function is increased by one for type I faults with probability p_2.
(iv). The fault content function is reduced by one for type II faults with probability q_0.
(v). The fault content function remains unchanged for type II of faults with probability q_1.
(vi). The fault content function is increased by one for type II faults with probability q_2.

- The debugging activity is performed without distinguishing between both types of faults.

Notations

α : Failure rate per remaining software fault for type I faults.
β : Failure rate per remaining software fault for type II faults.
m : Initial fault content for I type of faults.
M : Maximum fault content for I type of faults.
n : Initial fault content for the II type of faults.
N : Maximum fault content for II type of faults.
$F_{ij}(t)$:Probability that there are i(j) faults of type I (II) in the software at time t.
where $i = 0 \le i \le M$ *and* $j = 0 \le j \le N$

2.1. Governing Equations The transient equations governing the model are constructed by considering the transition flow rates. The differential difference equations associated with the various states are as follows (for balance equations are based on transition diagram in Fig 1):

$$F'_{00}(t) = \alpha p_0 F_{10}(t) + \beta q_0 F_{01}(t) \tag{1}$$

$$F'_{i0}(t) = -(i\alpha - i\alpha p_1) F_{i0}(t) + (i-1)\,\alpha p_2 F_{(i-1)0}(t) + (i+1)\,\alpha p_0 F_{(i+1)0}(t) + \beta q_0 F_{i1}(t), \quad 2 \le i \le M-1 \tag{2}$$

$$F'_{i0}(t) = -\alpha p_0 F_{i0} - \alpha p_2 F_{i0} + 2\alpha p_0 F_{(i+1)0}(t) + \beta q_0 F_{i1}(t), \quad i = 1 \tag{3}$$

$$F'_{M0}(t) = -M\alpha p_0 F_{M0}(t) + (M-1)\alpha p_2 F_{(M-1)0}(t) \tag{4}$$

$$F'_{0j}(t) = -\beta q_0 F_{0j}(t) - \beta q_2 F_{0j}(t) + \alpha p_0 F_{1,j}(t) + 2\beta q_0 F_{0(j+1)}, \quad j = 1 \tag{5}$$

$$F'_{0j}(t) = -(j\beta - j\beta q_1) F_{0j}(t) + (j-1)\,\beta q_2 F_{0(j-1)}(t) + \alpha p_0 F_{1,j}(t) + (j+1)\,\beta q_0 F_{0(j+1)}, \quad (2 \le j \le N-1) \tag{6}$$

$$F'_{0N}(t) = -N\beta q_0 F_{0N}(t) + (N-1)\beta q_2 F_{0(N-1)}(t) \tag{7}$$

$$F'_{i1}(t) = -\left[(\beta - \beta q_1) + (i\alpha - i\alpha p_1)\right] F_{i1}(t) + \alpha p_2 F_{11}(t) + (i+1)\,\alpha p_2 F_{(i+1)1}(t) + i\beta q_0 F_{i2}(t), \quad (2 \le i = M-2) \tag{8}$$

$$F'_{i1}(t) = -(\alpha p_0 + \beta q_0 + \alpha p_2 + \beta q_2) F_{i1}(t) + 2\alpha p_2 F_{(i+1)1}(t) + 2\beta q_0 F_{i2}(t), \quad i = 1 \tag{9}$$

$$F'_{(M-1)1}(t) = -(\beta q_0 + (M-1)\alpha p_0) F_{(M-1)1}(t) + (M-2)\alpha p_2 F_{(M-2)1}(t) \tag{10}$$

$$F'_{1j}(t) = -[(\alpha - \alpha p_1) + (j\beta - j\beta q_1)] F_{1j}(t) + (j-1)\beta q_2 F_{1(j-1)}(t) + 2\alpha p_0 F_{2j}(t) + (j+1)\beta q_0 F_{1(j+1)}(t), \quad (2 \le j \le N-2) \tag{11}$$

$$F'_{1(N-1)}(t) = -(\alpha p_0 + (N-1)\beta q_0) F_{1(N-1)}(t) + (N-2)\beta q_2 F_{1(N-2)}(t) \tag{12}$$

$$F'_{ij}(t) = -[(i\alpha - i\alpha p_1) + (j\beta - j\beta q_1)] F_{ij}(t) + (i-1)\alpha p_2 F_{(i-1)j}(j) + (j-1)\beta q_2 F_{i(j-1)}(t) + (i+1)\alpha p_0 F_{(i+1)j}(t) + (j+1)\beta q_0 F_{i(j+1)}(t), \; j = 2 \tag{13}$$

$$F'_{ij}(t) = -i\alpha p_0 F_{ij}(t) - (i+1)\beta q_0 F_{ij}(t) + i\beta q_2 F_{i(j-1)}(t) + (i-1)\alpha p_2 F_{ij}(t), \; i = M-3, \; j = N-2 \tag{14}$$

$$F'_{ij}(t) = -i\,\alpha\, p_0 F_{ij}(t) - (i-1)\beta\, q_0 F_{ij}(t) + \beta\, q_2 F_{i(j-1)}(t) + (i-1)\,\alpha\, p_2 F_{ij}(t), \; i = M-2, j = N-3 \tag{15}$$

2.2. Illustration In this section, for illustration purpose we present a markov model for two types of errors where maximum number of faults of each type are five (i.e. M=N=5). The differential difference equations related to this particular case are as follows:

$$F'_{00}(t) = \alpha p_0 F_{10}(t) + \beta q_0 F_{01}(t) \tag{16}$$

$$F'_{10}(t) = -(\alpha - \alpha p_1) F_{10}(t) + 2\alpha p_0 F_{20}(t) + \beta q_0 F_{11}(t) \tag{17}$$

$$F'_{20}(t) = -(2\alpha - 2\alpha p_1) F_{20}(t) + \alpha p_2 F_{10}(t) + 3\alpha p_0 F_{30}(t) + \beta q_0 F_{21}(t) \tag{18}$$

$$F'_{30}(t) = -(3\alpha - 3\alpha p_1) F_{30}(t) + 2\alpha p_2 F_{20}(t) + 4\alpha p_0 F_{40}(t) + \beta q_0 F_{31}(t) \tag{19}$$

$$F'_{40}(t) = -(4\alpha - 4\alpha p_1) F_{40}(t) + 3\alpha p_2 F_{30}(t) + 5\alpha p_0 F_{50}(t) + \beta q_0 F_{41}(t) \tag{20}$$

$$F'_{50}(t) = -5\alpha p_0 F_{50}(t) + 4\alpha p_2 F_{40}(t) \tag{21}$$

$$F'_{01}(t) = -(\beta - \beta q_1) F_{01}(t) + 2\beta q_0 F_{02}(t) + \alpha p_0 F_{11}(t) \tag{22}$$

$$F'_{11}(t) = -[(\beta - \beta q_1) + (\alpha - \alpha p_1)] F_{11}(t) + 2\beta q_0 F_{12}(t) + 2\alpha p_0 F_{21}(t) \tag{23}$$

$$F'_{21}(t) = -[(\beta - \beta q_1) + (2\alpha - 2\alpha p_1)] F_{21}(t) + \alpha p_2 F_{11}(t) + 2\beta q_0 F_{22}(t) + 3\alpha p_0 F_{31}(t) \tag{24}$$

$$F'_{31}(t) = -[(\beta - \beta q_1) + (3\alpha - 3\alpha p_1)] F_{31}(t) + 2\alpha p_2 F_{21}(t) + 3\beta q_0 F_{32}(t) + 4\alpha p_0 F_{41}(t) \tag{25}$$

$$F'_{41}(t) = -(\beta q_0 + 4\alpha p_0) F_{41}(t) + 3\alpha p_2 F_{31}(t) \tag{26}$$

$$F'_{02}(t) = -(2\beta - 2\beta q_1) F_{02}(t) + \beta q_2 F_{01}(t) + 3\beta q_0 F_{03}(t) + \alpha p_0 F_{12}(t) \tag{27}$$

$$F'_{12}(t) = -[(2\beta - 2\beta q_1) + (\alpha - \alpha p_1)] F_{12}(t) + \beta q_2 F_{11}(t) + 3\beta q_0 F_{13}(t) + 2\alpha p_0 F_{22}(t) \tag{28}$$

$$F'_{22}(t) = -[(2\beta - 2\beta q_1) + (2\alpha - 2\alpha p_1)] F_{22}(t) + \beta q_2 F_{21}(t) + 3\beta q_0 F_{23}(t) + 3\alpha p_0 F_{32}(t) + \alpha p_2 F_{12}(t) \tag{29}$$

$$F'_{32}(t) = -(3\beta q_0 + 3\alpha p_0) F_{32}(t) + \beta q_2 F_{31}(t) + 2\alpha p_2 F_{22}(t) \tag{30}$$

$$F'_{03}(t) = -(3\beta - 3\beta q_1) F_{03}(t) + 2\beta q_2 F_{02}(t) + 4\beta q_0 F_{04}(t) + \alpha p_0 F_{13}(t) \tag{31}$$

$$F'_{13}(t) = -[(3\beta - 3\beta q_1) + (\alpha - \alpha p_1)] F_{13}(t) + 2\beta q_2 F_{12}(t) + 4\beta q_0 F_{14}(t) + 2\alpha p_0 F_{23}(t) \tag{32}$$

$$F'_{23}(t) = -(3\beta q_0 + 2\alpha p_0) F_{23}(t) + 2\beta q_2 F_{22}(t) + \alpha p_2 F_{13}(t) \tag{33}$$

$$F'_{04}(t) = -(4\beta - 4\beta q_1) F_{04}(t) + 3\beta q_2 F_{03}(t) + 5\beta q_0 F_{05}(t) + \alpha p_0 F_{14}(t) \tag{34}$$

$$F'_{14}(t) = -(4\beta q_0 + \alpha p_0) F_{14}(t) + 3\beta q_2 F_{13}(t) \tag{35}$$

$$F'_{05}(t) = -5\beta q_0 F_{05}(t) + 4\beta q_2 F_{04}(t) \tag{36}$$

3. THE SOLUTION APPROACH

We denote the Laplace transform of $f_{ij}(t)$ by $\tilde{f}_{ij}(s)$. For solving the set of equations governing the model, we take Laplace transforms of equations (16)-(36) and solve using matrix method with initial conditions $f_k(0) = 1, f_n(0) = 0 \;\; for\; n \neq k$.

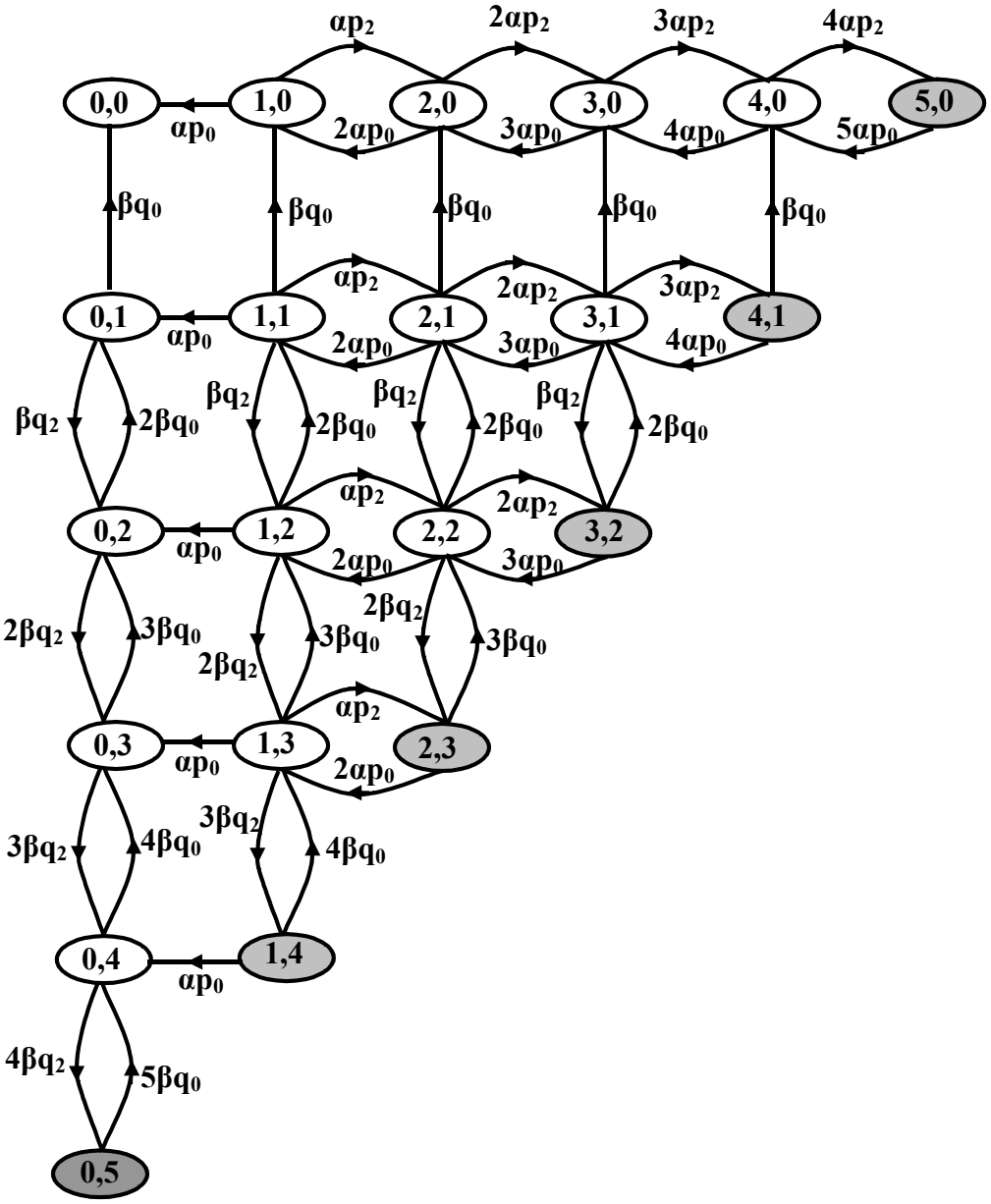

Fig 1. State transition diagram

$$-\alpha p_0\tilde{F}_{10}(s)-\beta q_0\tilde{F}_{01}(s)=1 \tag{37}$$

$$-2\alpha p_0\tilde{F}_{20}(s)-\beta q_0\tilde{F}_{11}(s)+\left[s+\left(\alpha-\alpha p_1\right)\right]\tilde{F}_{10}(s)=0 \tag{38}$$

$$-\alpha p_2\tilde{F}_{10}(s)-3\alpha p_0\tilde{F}_{30}(s)-\beta q_0\tilde{F}_{21}(s)+\left[s+\left(2\alpha-2\alpha p_1\right)\right]\tilde{F}_{20}(s)=0 \tag{39}$$

$$-2\alpha p_2\tilde{F}_{20}(s)-4\alpha p_0\tilde{F}_{40}(s)-\beta q_0\tilde{F}_{31}(s)+\left[s+\left(3\alpha-3\alpha p_1\right)\right]\tilde{F}_{30}(s)=0 \tag{40}$$

$$-3\alpha p_2\tilde{F}_{30}(s)-5\alpha p_0\tilde{F}_{50}(s)-\beta q_0\tilde{F}_{41}(s)+\left[s+\left(4\alpha-4\alpha p_1\right)\right]\tilde{F}_{40}(s)=0 \tag{41}$$

$$-4\alpha p_2\tilde{F}_{40}(s)+\left(s+5\alpha p_0\right)\tilde{F}_{50}(s)=0 \tag{42}$$

$$-2\beta q_0\tilde{F}_{02}(s)-\alpha p_0\tilde{F}_{11}(s)+\left[s+\left(\beta-\beta q_1\right)\right]\tilde{F}_{01}(s)=0 \tag{43}$$

$$-2\beta q_0\tilde{F}_{12}(s)-2\alpha p_0\tilde{F}_{21}(s)+\left[s+\left(\beta-\beta q_1\right)+\left(\alpha-\alpha p_1\right)\right]\tilde{F}_{11}(s)=0 \tag{44}$$

$$-\alpha p_2\tilde{F}_{11}(s)-2\beta q_0\tilde{F}_{22}(s)-3\alpha p_0\tilde{F}_{31}(s)+\left[s+\left(\beta-\beta q_1\right)+\left(2\alpha-2\alpha p_1\right)\right]\tilde{F}_{21}(s)=0 \tag{45}$$

$$-2\alpha p_2\tilde{F}_{21}(s)-3\beta q_0\tilde{F}_{32}(s)-4\alpha p_0\tilde{F}_{41}(s)+\left[s+\left(\beta-\beta q_1\right)+\left(3\alpha-3\alpha p_1\right)\right]\tilde{F}_{31}(s)=0 \tag{46}$$

$$-3\alpha p_2\tilde{F}_{31}(s)+\left[s+\left(\beta q_0+4\alpha p_0\right)\right]\tilde{F}_{41}(s)=0 \tag{47}$$

$$-\beta q_2\tilde{F}_{01}(s)-3\beta q_0\tilde{F}_{03}(s)-\alpha p_0\tilde{F}_{12}(s)+\left[s+\left(2\beta-2\beta q_1\right)\right]\tilde{F}_{02}(s)=0 \tag{48}$$

$$-\beta q_2\tilde{F}_{11}(s)-3\beta q_0\tilde{F}_{13}(s)-2\alpha p_0\tilde{F}_{22}(s)+\left[s+\left(2\beta-2\beta q_1\right)+\left(\alpha-\alpha p_1\right)\right]\tilde{F}_{12}(s)=0 \tag{49}$$

$$-\beta q_2\tilde{F}_{21}(s)-3\beta q_0\tilde{F}_{23}(s)-3\alpha p_0\tilde{F}_{32}(s)-\alpha p_2\tilde{F}_{12}(s)+\left[\left(2\beta-2\beta q_1\right)+\left(2\alpha-2\alpha p_1\right)\right]\tilde{F}_{22}(s)=0 \tag{50}$$

$$-\beta q_2\tilde{F}_{31}(s)-2\alpha p_2\tilde{F}_{22}(s)+\left[s+\left(3\beta q_0+3\alpha p_0\right)\right]\tilde{F}_{32}(s)=0 \tag{51}$$

$$-2\beta q_2\tilde{F}_{02}(s)-4\beta q_0\tilde{F}_{04}(s)-\alpha p_0\tilde{F}_{13}(s)+\left[s+\left(3\beta-3\beta q_1\right)\right]\tilde{F}_{03}(s)=0 \tag{52}$$

$$-2\beta q_2\tilde{F}_{12}(s)-4\beta q_0\tilde{F}_{14}(s)-2\alpha p_0\tilde{F}_{23}(s)+\left[s+\left(3\beta-3\beta q_1\right)+\left(\alpha-\alpha p_1\right)\right]\tilde{F}_{13}(s)=0 \tag{53}$$

$$-2\beta q_2\tilde{F}_{22}(s)-\alpha p_2\tilde{F}_{13}(s)+\left[s+\left(3\beta q_0+2\alpha p_0\right)\right]\tilde{F}_{23}(s)=0 \tag{54}$$

$$-3\beta q_2\tilde{F}_{03}(s)-5\beta q_0\tilde{F}_{05}(s)-\alpha p_0\tilde{F}_{14}(s)+\left[s+\left(4\beta-4\beta q_1\right)\right]\tilde{F}_{04}(s)=0 \tag{55}$$

$$-3\beta q_2\tilde{F}_{13}(s)+\left[s+\left(4\beta q_0+\alpha p_0\right)\right]\tilde{F}_{14}(s)=0 \tag{56}$$

$$-4\beta q_2\tilde{F}_{04}(s)+\left(s+5\beta q_0\right)\tilde{F}_{05}(s)=0 \tag{57}$$

For brevity, we denote the probabilities $F_{i,j}$ and Laplace transform of probabilities $\tilde{F}_{i,j}(s)$ with single suffix i.e. by F_i as defined and $\tilde{F}_i(s)$, respectively below:

$$F_{i,0}=F_{i+1},\tilde{F}_{i,0}(s)=\tilde{F}_{i+1}(s),\ 0\le i\le 5;$$

$$F_{i,1}=F_{6+i+1},\tilde{F}_{i,1}(s)=\tilde{F}_{6+i+1}(s),\ 0\le i\le 4;$$

$$F_{i,2}=F_{11+i+1},\tilde{F}_{i,2}(s)=\tilde{F}_{11+i+1}(s),\ 0\le i\le 3;$$

$$F_{i,3}=F_{15+i+1},\tilde{F}_{i,3}(s)=\tilde{F}_{15+i+1}(s),\ 0\le i\le 2;$$

$$F_{i,4}=F_{18+i+1},\tilde{F}_{i,4}(s)=\tilde{F}_{18+i+1}(s),\ 0\le i\le 1;$$

$$F_{i,5}=F_{21},\tilde{F}_{i,5}(s)=\tilde{F}_{21}(s)$$

The system of Eqs (37)-(57) reduces to matrix form as

$$\mathrm{Q}(s).\tilde{F}(s)=F(0) \tag{58}$$

where, $\tilde{F}(s)$ and $F(0)$ are the column vector i.e., $\tilde{F}(s)=\left[\tilde{F}_1(s),\tilde{F}_2(s),\ldots\tilde{F}_{21}(s)\right]^T$ and $F(0)=[1,0,0,\ldots,0]^T$ Also Q(s) is matrix given by

$$\mathrm{Q}(s)=\begin{bmatrix}A_1 & A_2 & 0\\ A_3 & A_4 & A_5\\ 0 & A_6 & A_7\end{bmatrix}_{21\times 21}$$

Here submatrices A_i (i=1,2,……9) are constructed for particular case as follows:

$$A_1=\begin{bmatrix} s & -\alpha p_0 & 0 & 0 & 0 & 0 & -\beta q_0 \\ 0 & s+(\alpha-\alpha p_1) & -2\alpha p_0 & 0 & 0 & 0 & 0 \\ 0 & -\alpha p_2 & s+(2\alpha-2\alpha p_1) & -3\alpha p_0 & 0 & 0 & 0 \\ 0 & 0 & -2\alpha p_2 & s+(3\alpha-3\alpha p_1) & -4\alpha p_0 & 0 & 0 \\ 0 & 0 & 0 & -3\alpha p_2 & s+(4\alpha-4\alpha p_1) & -5\alpha p_0 & 0 \\ 0 & 0 & 0 & 0 & -4\alpha p_2 & s+5\alpha p_0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s+(\beta-\beta q_1) \end{bmatrix}$$

$$A_4=\begin{bmatrix} s+\theta_1 & -2\alpha p_0 & 0 & 0 & 0 & -2\beta q_0 & 0 \\ -\alpha p_2 & s+\theta_2 & -3\alpha p_0 & 0 & 0 & 0 & -2\beta q_0 \\ 0 & -2\alpha p_2 & s+\theta_3 & -4\alpha p_0 & 0 & 0 & 0 \\ 0 & 0 & -3\alpha p_2 & s+(\beta q_0+4\alpha p_0) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s+(2\beta-2\beta q_1) & -\alpha p_0 & 0 \\ -\beta q_2 & 0 & 0 & 0 & 0 & s+\theta_4 & -2\alpha p_0 \\ 0 & -\beta q_2 & 0 & 0 & 0 & -\alpha p_2 & s+\theta_5 \end{bmatrix}$$

$$A_7=\begin{bmatrix} s+(3\beta q_0+3\alpha p_0) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & s+(3\beta-3\beta q_1) & -\alpha p_0 & 0 & -4\beta q_0 & 0 & 0 \\ 0 & 0 & s+\theta_6 & -2\alpha p_0 & 0 & -4\beta q_0 & 0 \\ 0 & 0 & -\alpha p_2 & s+(3\beta q_0+2\alpha p_0) & 0 & 0 & 0 \\ 0 & -3\beta q_2 & 0 & 0 & s+(4\beta-4\beta q_1) & -\alpha p_0 & -5\beta q_0 \\ 0 & 0 & -3\beta q_2 & 0 & 0 & s+(4\beta q_0+\alpha p_0) & 0 \\ 0 & 0 & 0 & 0 & -4\beta q_2 & 0 & s+5\beta q_0 \end{bmatrix}$$

$$A_2=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\beta q_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -\beta q_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\beta q_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\beta q_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\alpha p_0 & 0 & 0 & 0 & -2\beta q_0 & 0 & 0 \end{bmatrix} \qquad A_5=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3\beta q_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -3\beta q_0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -3\beta q_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -3\beta q_0 & 0 & 0 & 0 \end{bmatrix}$$

$$A_3=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\beta q_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad A_6=\begin{bmatrix} 0 & 0 & -\beta q_2 & 0 & 0 & 0 & -2\alpha p_2 \\ 0 & 0 & 0 & 0 & -2\beta q_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2\beta q_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2\beta q_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

For brebiely of notations, in sub matrices, we have used

$\theta_i = \left[(\beta - \beta q_1) + i\alpha(1-p_1)\right]$, i=1, 2, 3

$\theta_i = \left[i\beta(1-q_1) + \alpha(1-p_1)\right]$, i=4, 5, 6

Using Cramer's rule, the probabilities $\tilde{F}_k(s)$, can be obtained as

$$\tilde{F}_k(s) = \frac{|Q_{k+1}(s)|}{|Q(s)|}, \qquad 0 \le k \le L \tag{59}$$

For calculating the characteristic roots of the matrix Q(s), we note that
s = 0 is one of the roots. Let s = -d, so that we get

$$Q(-d) = (Q - dI) \tag{60}$$

Now Eq (58) becomes

$$Q(-d).\tilde{F}(s) = (Q - dI)\tilde{F}(s) = F(0) \tag{61}$$

It may be observed that the eigen values of Q are real and distinct and Q is positive definite. So, all eigen values of Q are positive. Let $\nu_k\,(1 \le k \le L)$ denotes the eigen values of Q, then we get

$$|Q(s)| = s\prod_{k=1}^{L}(s+\nu_k) \tag{62}$$

so that

$$\tilde{F}_k(s) = \frac{|Q_{l+1}(s)|}{s\prod_{k=1}^{L}(s+\nu_k)}, \qquad 1 \le k \le L \tag{63}$$

We may expand $F_k(s)$ by partial fractions, i.e., in the form

$$\tilde{F}_1(s) = \frac{a_0}{s} + \sum_{k=1}^{L}\frac{a_{0k}}{s+\nu_k} \tag{64}$$

$$\tilde{F}_k(s) = \sum_{j=1}^{L}\frac{a_{kj}}{s+\nu_j}, \qquad k = 2, 3, \ldots, L \tag{65}$$

where a_0 and a_n (n=1,2,…..L) are real numbers calculated as

$$a_0 = \frac{|Q_1(0)|}{\prod_{j=1}^{L}\nu_j} \tag{66}$$

and

$$a_{lk} = -\frac{|Q_{l+1}(-\nu_k)|}{\nu_k\prod_{\substack{j=1\\ \neq k}}^{L}(\nu_j - \nu_k)}, \qquad 1 \le l \le L,\ 2 \le k \le L \tag{67}$$

On taking inverse Laplace transform of Eqs (59) and (60), we get

$$F_1(t) = \frac{|Q_1(0)|}{\prod_{k=1}^{L}\nu_k} - \sum_{k=1}^{L}\frac{|Q_1(-\nu_k)|\exp(-\nu_k t)}{\nu_k\prod_{\substack{j=1\\ \neq k}}^{L}(\nu_j - \nu_k)} \tag{68}$$

On inverting Eq (63), we have

$$F_l(t) = -\sum_{k=1}^{L}\frac{|Q_{l+1}(-\nu_k)\exp(-\nu_k t)|}{\nu_k\prod_{\substack{j=1\\ \neq k}}^{L}(\nu_j - \nu_k)}, \quad where\ 2 \le l \le L \tag{69}$$

4. PERFORMANCE INDICES

Now we give some performance measures for the quantification of software reliability indices as follows:

- The probability of a perfect program at time t is given by $F_1(t)$.
- The mean number of faults remaining in the software at time t is given as

$$E\{D(t)\} = \sum_{i=1}^{L} iF_i(t) \tag{70}$$

- The software reliability is defined as

$$R(x/t) = \sum_{i=1}^{L} F_i(t) \times \left(\exp\left(-(\alpha+\beta)x\right)\right)^i \tag{71}$$

5. SENSITIVITY ANALYSIS

Since no live data is available so instead of estimating the parameters we have used the secondary data for validity and practical utility of our model (Ref. Kapur et al. 1992).

In this section, we perform computational experiment for the transient analysis by employing Runge-Kutta technique (RKT) of fourth order and matrix method to solve the system of differential equations. R-K method is implemented by exploiting MATLAB's 'ode45' function. A time span is considered with equal intervals. Eigen

values are evaluated by the using MATLAB'6.5 software. For illustration purpose, we choose default parameters as $\alpha = 0.49$, $\beta = 0.02$, p_0=0.6, p_1=0.3, p_2=0.1, q_0=0.5, q_1=0.3, and q_2=0.2.

From Tables 1 and 2, we notice the patterns of various performance indices namely R(t) and E{D(t)} by varying the probabilities p_0, p_1, p_2 and q_0, q_1, q_2, respectively. It is observed that there is an increasing trend in the values of R(t) and decreasing trend in E{D(t)} with the increasing values of p_0 and q_0.

Table 1. Software reliability and E{D(t)} for different values of p_0, $p_1$and p_2.

p_0	p_1	p_2	t	R(t)	E{D(t)}
0.6	0.25	0.15	0	0.787	12.000
			1	0.852	8.122
			2	0.897	5.591
			3	0.926	3.950
			4	0.946	2.893
			5	0.958	2.213
0.7	0.2	0.1	0	0.787	12.000
			1	0.852	8.122
			2	0.897	5.591
			3	0.926	3.950
			4	0.946	2.893
			5	0.958	2.213
0.8	0.13	0.07	0	0.787	12.000
			1	0.852	8.122
			2	0.897	5.591
			3	0.926	3.950
			4	0.946	2.893
			5	0.958	2.213

Table 2. Software reliability and E{D(t)} for different values of q_0, $q_1$and q_2.

q_0	q_1	q_2	t	R(t)	E{D(t)}
0.6	0.3	0.1	0	0.787	12.000
			1	0.858	7.848
			2	0.904	5.227
			3	0.933	3.596
			4	0.951	2.589
			5	0.963	1.971
0.7	0.2	0.1	0	0.787	12.000
			1	0.868	7.238
			2	0.917	4.494
			3	0.945	2.944
			4	0.961	2.078
			5	0.969	1.596
0.8	0.13	0.07	0	0.787	12.000
			1	0.880	6.542
			2	0.930	5.591
			3	0.956	3.950
			4	0.968	2.893
			5	0.974	2.213

The effects of failure rates α and β on I and II type faults, are shown in Tables 3 and 4. As expected, reliability R (t) increases with testing time whereas decreases with the increase in the failure rate α. Mean number of remaining faults decreases as testing time increases but remains same for the increasing values of failure rate α.

Table 3. Performance indices for different values of α.

	α=0.01		α =0.03		α =0.05	
t	R(t)	E{D(t)}	R(t)	E{D(t)}	R(t)	E{D(t)}
0	0.698	12.000	0.549	12.000	0.432	12.000
1	0.796	7.848	0.691	7.848	0.603	7.848
2	0.862	5.227	0.787	5.227	0.724	5.227
3	0.903	3.596	0.850	3.596	0.803	3.596
4	0.929	2.589	0.889	2.589	0.852	2.589
5	0.945	1.971	0.913	1.971	0.883	1.971
6	0.955	1.593	0.928	1.593	0.902	1.593
7	0.961	1.362	0.937	1.362	0.914	1.362

Table 4. Performance indices for different values of β.

	β=0.5		β =0.6		β =0.7	
t	R(t)	E{D(t)}	R(t)	E{D(t)}	R(t)	E{D(t)}
0	0.000	12.000	0.000	12.000	0.000	12.000
1	0.071	7.848	0.073	7.524	0.074	7.215
2	0.167	5.227	0.166	4.832	0.164	4.473
3	0.240	3.596	0.233	3.238	0.223	2.929
4	0.289	2.589	0.275	2.303	0.258	2.067
5	0.321	1.971	0.300	1.757	0.279	1.590
6	0.340	1.593	0.315	1.439	0.290	1.326
7	0.352	1.362	0.324	1.255	0.296	1.180

Figs 2a and 2b depict the trend of probability of perfect program for different parameters α and β. It is noticed that the accuracy of the software increases as testing time increases. We also notice that there is no significant change with the increasing values of α but as β increases, the probability of perfect program increases for some time, and finally becomes constant.

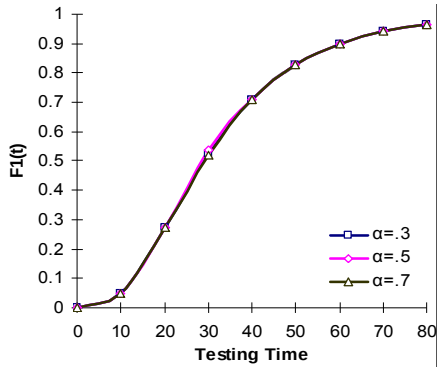

Fig 2a. Probability of perfect program at time t by varying α.

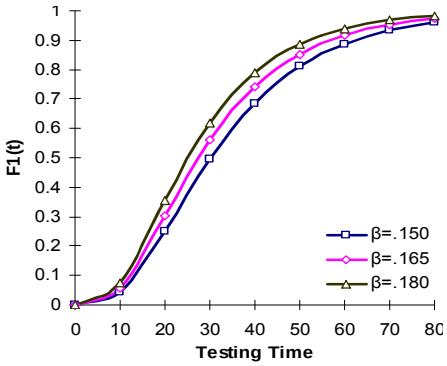

Fig 2b. Probability of perfect program at time t by varying β.

Figs 3a and 3b are plotted for the reliability by varying α and β for I and II type of faults, respectively. From Fig 3a, we notice that the reliability decreases as α increases. But in Fig 3b, initially reliability increases with the testing time and remains almost same with the increasing the value of β.

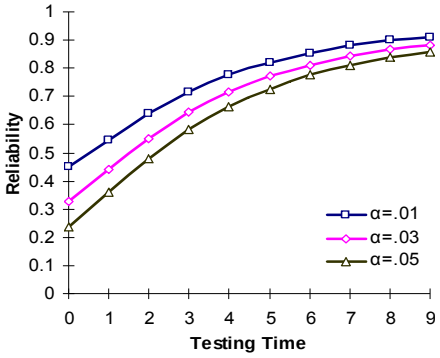

Fig 3a. Software reliability by varying α

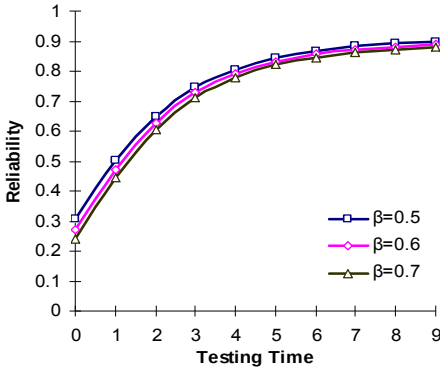

Fig 3b. Software reliability by varying β

From Figs 4a and 4b, mean number of remaining faults E{D(t)} has been examined by varying the parameters α and β. It is seen that E{D(t)} decreases as time increases but remains same for all values of α and β.

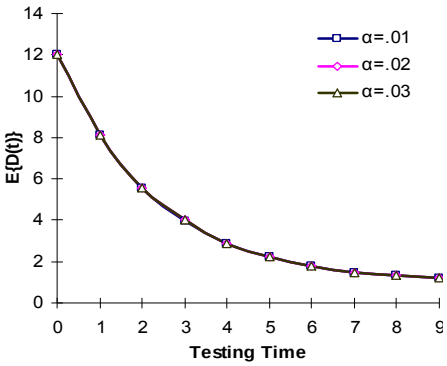

Fig. 4a. Mean number of faults remaining by varying α.

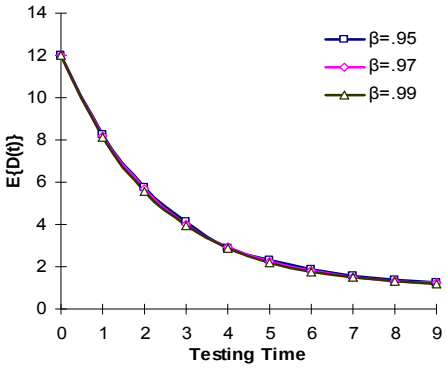

Fig. 4b. Mean number of faults remaining by varying β.

Overall, with the help of numerical results we observe that the optimal release time of the software can be determined successfully. For

example, if the initial error content function is assumed to be 12, we can notice from the Figs 4a and 4b that $E\{D(t)\} \leq 12$. This means that the software developer can decide the time to a specific software quality level with the condition that the reliability may reach at a maximum level. Similar conclusions for other performance measures are also evident from the other figures and tables.

Based on the sensitivity analysis which has been given in this paper, one can estimate the idea about the release time of the software. For example, from Tables 9.3 and 9.2, we see that as remaining faults in the software are becoming less then reliability of the software is increasing and finally it became constant that shows the real situation of testing the software. In that sense, the markovian software reliability model with imperfect debugging and generation of errors proposed in this paper are intuitively understandable and can provide to a software developer a more tractable framework for developing a real time situation assessment tools in spite of their simple structure.

6. CONCLUSION

In this paper, we have developed the markovian software reliability model by including the concept of imperfect debugging and error generation phenomenon. The suggested approach is suitable for practical application in reliability engineering. Our stochastic model provides a theoretical framework during the software development for understanding the factors that affect the software reliability. The suggested model may be helpful in measuring and assessing the software reliability, during operational phase.

7. REFERENCES

1. Cai, K.-Y., Cao, P., Dong, Z. and Liu, K. (2010): Mathematical modeling of software reliability testing with imperfect debugging, ***Computers & Mathematics with Applications***, Vol. 59, No. 10, 3245-3285.
2. Chung, H. and Ortega, A. (2005): Analysis and testing for error tolerant motion estimation, ***In Proceeding of International Symposium on Defect and Fault Tolerance in VLSI Systems***, 514-522.
3. Dohi, T., Yasui, K. and Osaki, S. (2003): Software reliability assessment models based on cumulative Bernoulli trial processes, ***Mathematical and Computer Modelling***, Vol. 38, 1177-1184.
4. Dulz, W., Holpp, S. and German, R. (2010): A Polyhedron Approach to Calculate Probability Distributions for Markov Chain Usage Models, ***Electronic Notes in Theoretical Computer Science***, Vol. 264, No. 3, 19-35.
5. Gokhale, S. S. and Trivedi, K. S. (2006): Analytical models for architecture-based software reliability prediction: a unification framework, ***IEEE Transactions on Reliability***, Vol. 55, No. 4, 578-590.
6. Gokhale, S. S., Philip, T. and Marinos, P. N. (1996): A non-homogeneous markov software reliability model with imperfect repair, ***In Proceeding of International Performance and Dependability Symposium***.
7. Gupta, N. and Singh, M. P. (2006): Estimation of software reliability by sequential testing with simulated annealing of mean field approximation, ***International Journal of Engineering, Transactions B: Applications, Iran***, Vol. 19, No. 1, 35-44.
8. Hamlet, D., Woit, D. and Mason, D. (2001): Theory of software reliability based components, ***In Proceeding of International Conference on Software Engineering***, 361-370.
9. Huang C.-Y. and Hung, T.-Y. (2010): Software reliability analysis and assessment using queueing models with multiple change-points, ***Computers & Mathematics with Applications***, Vol. 60, No. 7, 2015-2030.
10. Jalote, P., Murphy, B. and Sharma, V. S. (2008): Post-release reliability growth in software products, ***ACM Transactions on Software Engineering and Methodology***, Vol. 17, No. 4, 17.1-17.20.
11. Jelinski, Z. and Moranda, P. B. (1972): Software reliability research, ***Statistical Computer Performance Evaluation, New York***, 468-484.
12. Kapur, P. K., Sharma, K. D. and Garg, R. B. (1992): Transient solution of a software reliability model with imperfect debugging and error generation, ***Microelectronics and Reliability***, Vol. 38, No. 4, 475-478.
13. Lisnianski, A. (2007): The markov reward model for a multi-state system reliability assessment with variable demand, ***Quality Technology & Quantitative Management***, Vol. 4, No. 2, 265-278.
14. Lisnianski, A. and Frenkel, I. (2009): Non-Homogeneous markov reward model for aging multi state system under minimal repair, ***International Journal of Performability Engineering***, Vol. 5, No. 4, 303-312.
15. Lisnianski, A. and Frenkel, I., Khvatskin, L. and Ding, Y. (2007): Markov reward model for multi state system reliability assessment, ***Statistical models and Methods for Biomedical and Technical Systems***, 153-168.
16. Meedeniya, I., Buhnova, B., Aleti, A. and Grunske, L. (2011): Reliability-driven deployment optimization for embedded systems, ***Journal of Systems and Software***, Vol. 84, No. 5, 835-846.
17. Musa, J. D. (1975): Theory of Software reliability and its application, ***IEEE Transactions on Software***

Engineering, SE-1, 312-327.

18. Prowell, S. J. and Poore, J. H. (2004): Computing system reliability using markov chain usage models, ***The journal of Systems and Software***, Vol. 73, 219-225.
19. Ravishanker, N., Liu, Z. and Ray, B. K. (2008): NHPP models with markov switching for software reliability, ***Computational Statistics and Data Analysis***, Vol. 52, 3988-3999.
20. Shooman, M. L. (1977): Software reliability: analysis and prediction, ***Report AGARD AG224, Integrity in Electronic Flight Control Systems***, 1-17.
21. Sridharan, V. and Jayashree, P. R. (1998): Transient solution of a software model with imperfect debugging and generation of errors by two servers, ***Mathematical and Computing Modelling***, Vol. 27, No. 3, 103-108.
22. Tokuno, K. and Yamada, S. (1999a): A summary of markovian software availability modeling, ***In Proceeding of Fifth ISSAT International Conference Of Reliability and Quality in Design***, (Edited by Pham, H. and Lu, M.-W.), 218-222.
23. Tokuno, K. and Yamada, S. (1999b): A markovian software reliability model with a decreasing perfect debugging rate**, In proceeding First Western Pacific and Third Australia-Japan Workshop on Stochastic Models in Engineering, Technology and Management**, (Edited by Wilson, R. J., Osaki, S. and Faddy, M. J.), 528-536.
24. Tokuno, K. and Yamada, S. (2000): An imperfect debugging model with two types of hazard rates for software reliability measurement and assessment, ***Mathematical and Computing Modelling***, Vol. 31, 343-352.
25. Tokuno, K. and Yamada, S. (2003a): Markovian software reliability measurement with a geometrically decreasing perfect debugging rate, ***Mathematical and Computing Modelling***, Vol. 38, 1443-1451.
26. Tokuno, K. and Yamada, S. (2003b): Relationship between software availability measurement and the number of restorations with imperfect debugging, ***Computers and Mathematics with Applications***, Vol. 46, 1155-1163.
27. Whittaker, J. A., Rekab, K. and Thomson, M. G. (2000): A markov chain model for predicting the reliability of multi-build software, ***Information and Software Technology***, Vol. 42, 889-894.
28. Yamada, S., Tokuno, K. and Osaki, S. (1993): Software reliability measurement in imperfect debugging environment and its application, ***Reliability Engineering and System Safety***, Vol. 40, 139-147.