Information Sciences 218 (2013) 119–132

Contents lists available at SciVerse ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Quantitative effects of software testing on reliability improvement in the presence of imperfect debugging

Ping Cao [a,*], Zhao Dong [a], Ke Liu [a], Kai-Yuan Cai [b,c]

[a] MADIS and National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Sciences, CAS, Beijing 100190, China
[b] Department of Automatic Control, Beijing University of Aeronautics Astronautics, Beijing 100191, China
[c] State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing 100190, China

ARTICLE INFO

*Article history:*
Received 22 April 2009
Received in revised form 22 June 2012
Accepted 26 June 2012
Available online 7 July 2012

*Keywords:*
Markov usage model
Software reliability testing
Imperfect debugging

ABSTRACT

Software testing is essential for software reliability improvement and assurance. However, software testing is subject to imperfect debugging in the sense that new defects may be introduced into the software under test while detected defects are removed. The quantitative effects of software testing on software reliability improvement are obscure. In this paper we propose a Markov usage model to explore the quantitative relationships between software testing and software reliability in the presence of imperfect debugging. Several interesting quantities for software reliability assessment are derived and the corresponding upper and lower bounds are obtained.



## 1. Introduction

Software testing is the most popular method used for enhancing the quality of software. Relative to two other software reliability methods, automatic verification and deductive verification, software testing is simple, feasible and gives a high cost/performance ratio [11]. Thus, software testing is widely used in software industry. Since software testing is based on sampling of the software executions, it cannot guarantee to detect all the defects in a given software. Therefore, the level of software reliability achieved after testing is a problem deserving special attention. However, the quantitative effects of software testing on software reliability improvement are obscure. This motivated a considerable amount of research work in the recent years on software reliability testing, that explores the quantitative relationships between software testing and software reliability [2,3,10].

Software testing methods are traditionally divided into white- and black-box testing. White box testing can be used when the tester has access to the internal data structures and algorithms, while black box testing assumes no knowledge of internal implementation. For more details on the software testing methods, please refer to Mathur [8].

In this paper, we focus on one testing method, that is, statistical testing. Statistical testing assumes that the test cases are stochastically generated based on a probability distribution that represents a profile of actual or anticipated use of the software [1,12]. Then statistical analysis can be performed on the test history that enables the measurement of various probabilistic aspects of the testing process.

One special kind of statistical testing is Markov usage model based testing, which was proposed by Whittaker and Thomason [16]. Instead of supposing that each time the test case is chosen randomly, Markov usage model based testing assumes that the tester chooses test cases in accordance with a Markov chain to reflect the operational profile and the interactive nature of the software under test. However, Markov usage model based testing only gives a statistical method of how

* Corresponding author.
*E-mail addresses:* caoping@amss.ac.cn (P. Cao), dzhao@amt.ac.cn (Z. Dong), kliu@amss.ac.cn (K. Liu), kycai@buaa.edu.cn (K.-Y. Cai).

0020-0255/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2012.06.034

to choose the test cases. Executing test cases, detecting and removing defects, which are fundamental parts of software testing process, are not described in Whittaker and Thomason's model. Therefore, the quantitative relationship between software testing and software reliability becomes obscure in the model.

Ozekici and Soyer [10] provided a stochastic and statistical framework to model software reliability in the presence of an operational profile and discussed several issues related to software reliability and statistical inference. Using this model, several quantities of interest on software reliability can be computed. However, it works only in the operational profile in which the defect number remains unchanged. It cannot be applied in the testing process since the removal process is involved in the testing process and makes the defect number change all the time.

Considering these disadvantages of Markov usage model based testing, Cai et al. [2,3] built up a mathematical modeling framework to model the testing process, which can be used to examine the quantitative relationship between software testing and software reliability. Assuming that the execution time for each action is independent of the total current defect number in the system, Cai et al. studied the testing process through the embedded Markov chain and found that under some assumptions, the software testing processes can be treated as a linear dynamic system. One limitation of this framework is the assumption that the defect removal is perfect, which is impractical in software engineering [5,17].

Several imperfect debugging models have been proposed in the literature [6,13–15]. Gokhale et al. [6] examined various kinds of defect removal policies and analyzed the effect of these policies on the residual number of defects at the end of the testing process, using a non-homogeneous continuous time Markov chain. However, this paper only proposed a modeling framework and did not derive quantitative measures for software reliability assessment. Tokuno and Yamada [13–15] proposed a software reliability model in an imperfect debugging environment where it was assumed that there are two types of software failures: the first type is caused by the defects latent in the system before the testing and the second type is caused by the defects regenerated randomly during the testing phase. But all these papers [13–15] assumed that the residual number of defects will not increase during the defect removal process, which is not reasonable in reality. Cai et al. [4] extended their previous modeling framework [2,3] to the case of imperfect debugging, by adding the imperfect defect removal assumption proposed in [9]. It is found that the software testing process may induce a linear or nonlinear dynamic system, depending on the relationship between the probability of debugging introducing a new defect and that of debugging removing a detected defect, which is a good supplement of their previous work.

In this paper, we extend the model in [2,3] to the case of imperfect debugging in the sense that new defects may be introduced into the software under test in the defect removal process. In contrast with [4], we assume that the execution time of a testing action depends on the chosen action as well as the current number of defects remaining in the software under test, which is also adopted in Ozekici and Soyer [10]. Therefore, the mathematical tool we use is continuous Markov process rather than embedded Markov chain used in [4]. Since our model is built on the testing process, we add the assumption that the execution of a test case may be terminated without detecting any defect. This is not considered in Ozekici and Soyer's model, which is built on the operational profile. We also assume that the time required for defect removals is ignored, which is adopted in most literature [2–4,13–15]. Based on the Markovian property of our model, we obtain some interesting quantities for software reliability assessment, including dynamic behaviors of the resident defect number (RDN) and the observed failure number (OFN), distribution of the time between software failures, software reliability and distribution of the first passage time to the specified number of residual defects. From the discussion we find that the existence of Markov usage model makes the analysis of the quantitative measure cumbersome. It is hard or even impossible to obtain the analytical form of the quantities for software reliability assessment. Therefore, we discuss the upper and lower bounds of these quantities instead, which can aid to assess the software reliability.

This paper follows the style of our previous works [2,3] and discusses the imperfect debugging in the testing process from the mathematical aspect. However this by no means implies that the engineering aspect of software testing process with imperfect debugging is less important. Actually, some assumptions used in our model do not seem most reasonable in realistic software engineering, though most assumptions used have been proposed and adopted in the literature. While we focus on the mathematical analysis and the resulting insights, the corresponding empirical justifications are left to future work.

The organization of this paper is as follows: Section 2 gives the model description; in Section 3 we consider the dynamic behaviors of RDN and OFN; in Section 4 we study some distributions such as distribution of the time between software failures, software reliability and distribution of the first passage time to the specified number of residual defects; Section 5 concludes the paper.

## 2. Model

In this section we first introduce the basic assumptions of software reliability testing model, and then give its $Q$-matrix.

We assume that the given test suite of the software under test comprises $m$ classes of test case $C_1,\ C_2,\ \ldots,\ C_m$ and there are initially $N$ defects, which are assumed to be known at the beginning. If we let $A_i$ be the $(i+1)$th testing action taken since the beginning of software testing, and $A_i = j$ means that the $(i+1)$th testing action picks up a test case from $C_j$ (in this case we call that the $(i+1)$th testing chooses action $j$), then the dynamic testing action process evolves as follows: The first testing action is selected according to the probability distribution $\{p_1, p_2, \ldots, p_m\}$, that is, $\Pr\{A_0 = j\} = p_j,\ j = 1,\ 2,\ \ldots,\ m$; and next test case is immediately selected and executed after the current action is finished; the sequence $\{A_0, A_1, A_2, \ldots, A_i, A_{i+1}, \ldots\}$ forms a Markov chain with $\Pr\{A_{i+1} = l | A_i = k\} = p_{kl}, i \geqslant 0$, where $p_{kl}$ is the transition probability from action $k$ to action $l$, satisfying $p_{kl} \geqslant 0,\ \sum_{l=1}^{m} p_{kl} = 1,\ k,\ l = 1, 2, \ldots, m$.

Two independent and exclusive cases may cause the termination of the $n$-th test. One is that the test may be terminated without detecting any defect. In this case, we assume that the testing time length $T_n^1$ satisfies $\Pr\{T_n^1 > t|N_{n-1}=k, A_{n-1}=i\} = e^{-\mu t}$, i.e., the test will be terminated with rate $\mu$. The other case is that the test may be terminated by detecting a defect. In this case, we assume that the testing time length $T_n^2$ satisfies $\Pr\{T_n^2 > t|N_{n-1}=k, A_{n-1}=i\} = e^{-k\theta_i t}$, i.e., the test will be terminated with a rate which is proportional with the number of the remaining defects. Here $\theta_i$ denotes the average rate for detecting a remaining defect by action $i$, and $N_n$ denotes the number of defects remaining in the software after the $n$th test. It is easy to see that the distribution of $(T_n^1, T_n^2)$ is independent with $N_{n-1}, A_{n-1}$.

Either one of these two cases will cause the termination of the $n$th test. Therefore, the $n$-th testing time length $T_n$ can be denoted by $T_n = \min\{T_n^1, T_n^2\}$.

We also assume that the defect removal is imperfect, which means whenever a defect is detected, there are three mutually exclusive possibilities for the corresponding defect removal effort: reduction in the defect content by 1 with probability $q$, an increase in the defect content by 1 with probability $p$, no change in the defect content with probability $1-p-q$. And we assume that the defect removal is instantaneous, i.e., the defect removal process takes no time, which is also adopted in [6].

The dynamic process of software testing can be more precisely described as follows. At the beginning of the test, a testing action is chosen according to the evolution of the software testing action process described above. For the sake of clarity, we assume that action $i$ is chosen. Then a test case is selected from $C_i$ and executed subsequently. It takes a length of time $T_1$. If no defect is detected in this test, we will move to next test. If a defect is detected, we will try to remove the defect, which results in three mutually exclusive possibilities for the corresponding defect removal effort, and then move to next test. The next test will begin by choosing the testing action (say $j$) according to the evolution of the dynamic testing action process and subsequently executing a test case selected from $C_j$ and so on.

Because we assume that the defect removal is imperfect, we cannot know how many defects there are during the testing process and thus when there will be no defects in the software system. Therefore, we assume when there are no defects we still continue the testing process as before.

Next we give the $Q$-matrices of the testing process.

At time $t$, let $N_t$ be the remaining defects in the software testing system, $A_t$ be the taken action, $M_t$ be the number of the observed failures, and $H_t$ be the execution number of testing, respectively. It is not difficult to show that stochastic process $X_t = (N_t, A_t, M_t, H_t) \in \mathcal{N} \times \{1,2,\ldots,m\} \times \mathcal{N} \times \mathcal{N}$ ($\mathcal{N} = \{0,1,2,\ldots\}$ denotes the set of nonnegative integers) is a Markov process, and its $Q$-matrix is given by

$$Q_1((n,i,l,g),(m,j,l',g')) = \begin{cases} qn\theta_i p_{ij}, & m = n-1, l' = l+1, g' = g+1; \\ pn\theta_i p_{ij}, & m = n+1, l' = l+1, g' = g+1; \\ (1-p-q)n\theta_i p_{ij}, & m = n, l' = l+1, g' = g+1; \quad (n \geqslant 1) \\ \mu p_{ij}, & m = n, l' = l, g' = g+1; \\ -(n\theta_i + \mu), & m = n, j = i, l' = l, g' = g, \end{cases} \tag{1}$$

and

$$Q_1((0,i,l,g),(0,j,l',g')) = \begin{cases} \mu p_{ij}, & l' = l, g' = g+1; \\ -\mu, & j = i, l' = l, g' = g. \end{cases} \tag{2}$$

That is, if the state of the process $X_t$ is $(n,i,l,g)$ $(n \geqslant 1)$, it will move to states $(n-1,j,l+1,g+1)$, $(n+1,j,l+1,g+1)$, $(n,j,l+1,g+1)$ and $(n,j,l,g+1)$ with rates $qn\theta_i p_{ij}, pn\theta_i p_{ij}, (1-p-q)n\theta_i p_{ij}$ and $\mu p_{ij}$, respectively; If the state of the process $X_t$ is $(0,i,l,g)$, then it will move to state $(0,j,l,g+1)$ with rate $\mu p_{ij}$.

Next we give a brief explanation of the $Q$-matrix $Q_1$. When the process $X_t$ is in state $(n,i,l,g)$ $(n \geqslant 1)$, it is performing the $g$th test. It can be terminated without detecting any defect with rate $\mu$ or terminated by detecting a defect with rate $n\theta_i$. In the former case, both $N_t$ and $M_t$ are unchanged and $H_t$ is increased by 1. The $(g+1)$-th test will be started by choosing action $j$ with probability $p_{ij}$. Thus, it will move to state $(n,j,l,g+1)$ with rate $\mu p_{ij}$. In the latter case, both $M_t$ and $H_t$ are increased by 1 and the defect removal is performed. The number of defects is decreased by 1, increased by 1 and unchanged with probability $q, p$ and $1-p-q$, respectively. Then the $(g+1)$-th test will be started by choosing action $j$ with probability $p_{ij}$. Therefore, it will move to states $(n-1,j,l+1,g+1), (n+1,j,l+1,g+1)$ and $(n,j,l+1,g+1)$ with rates $qn\theta_i p_{ij}, pn\theta_i p_{ij}$ and $(1-p-q)n\theta_i p_{ij}$, respectively. The meaning of the following Q-matrices $Q_2$ and $Q_3$ can also be understood similarly and we omit the explanation of them for brevity.

The process $Y_t = (N_t, A_t) \in \mathcal{N} \times \{1,2,\ldots,m\}$ is a Markov process, with its $Q$-matrix

$$Q_2((n,i),(m,j)) = \begin{cases} qn\theta_i p_{ij}, & m = n-1; \\ pn\theta_i p_{ij}, & m = n+1; \\ ((1-p-q)n\theta_i + \mu)p_{ij}, & m = n, \quad j \neq i; \\ -(n\theta_i + \mu) + ((1-p-q)n\theta_i + \mu)p_{ii}, & m = n, \quad j = i, \end{cases} (n \geqslant 1) \tag{3}$$

and

$$Q_2((0,i),(0,j)) = \begin{cases} \mu p_{ij}, & j \neq i; \\ -\mu + \mu p_{ii}, & j = i. \end{cases} \tag{4}$$

The process $Z_t = (N_t, A_t, M_t) \in \mathcal{N} \times \{1,2,\ldots,m\} \times \mathcal{N}$ is also a Markov process with its $Q$-matrix

$$Q_3((n,i,l),(m,j,l')) = \begin{cases} qn\theta_i p_{ij}, & m = n-1, l' = l+1; \\ pn\theta_i p_{ij}, & m = n+1, l' = l+1; \\ (1-p-q)n\theta_i p_{ij}, & m = n, l' = l+1; \quad (n \geqslant 1) \\ \mu p_{ij}, & m = n, j \neq i, l' = l; \\ -(n\theta_i + \mu) + \mu p_{ii}, & m = n, j = i, l' = l, \end{cases} \tag{5}$$

and

$$Q_3((0,i,l),(0,j,l')) = \begin{cases} \mu p_{ij}, & j \neq i, \quad l' = l; \\ -\mu + \mu p_{ii}, & j = i, \quad l' = l. \end{cases} \tag{6}$$

It is not difficult to know that $(N_n, A_n)_{n=0,1,\ldots}$ is an embedded Markov chain of $Y_t$ and 0 is the absorbing state for $N_n$ when $q \neq 0$. So we have $\lim_{t\to\infty} N_t = 0$, a.s.

Since a homogenous Markov process can be fully determined by its initial distribution and its $Q$-matrix, in the next section we analyze dynamic behaviors of RDN and OFN through three $Q$-matrices.

## 3. Dynamic behaviors of RDN and OFN

The residual defect number $N_t$ in a software system directly contributes to its failure intensity, causing software unreliability, and it is also an important measure for the software developer from the point of view of planning maintenance activities [6]. Therefore, in Subsection 3.1 we study the dynamic behavior of $N_t$. But the value of $N_t$ cannot immediately be observed, since in the case of imperfect removal the cumulative number of software failures is not always the same as the number of failures that are corrected. However, the observed failure number $M_t$ can always be observed and it has a close relationship with $N_t$, so we can estimate $N_t$ through $M_t$. Thus in Subsection 3.2 we study the dynamic of $M_t$.

### 3.1. Dynamic behavior of RDN

In this subsection we derive an expression of the probability distribution of $N_t$ through the Markov processes $Y_t = (N_t, A_t)$ and calculate its expectation and variation.

Define $P_{ij}(n,t) = \Pr\{N_t = n, A_t = j | N_0 = N, A_0 = i\}$, $P(n,t) = (P_{ij}(n,t))_{m\times m}$. Noting that the $Q$-matrix of the process $Y_t = (N_t, A_t)$ is $Q_2$, by a standard Chapman–Kolmogorov argument for $n \geqslant 2$ we have

$$P'_{ij}(n,t) = P_{ij}(n,t)[-(n\theta_j + \mu) + ((1-p-q)n\theta_j + \mu)p_{jj}] + \sum_{\substack{k=1 \\ k\neq j}}^{m} P_{ik}(n,t)((1-p-q)n\theta_k + \mu)p_{kj}$$
$$+ \sum_{k=1}^{m} P_{ik}(n+1,t)q(n+1)\theta_k p_{kj} + \sum_{k=1}^{m} P_{ik}(n-1,t)p(n-1)\theta_k p_{kj},$$

which can be rewritten in matrix form for $n \geqslant 2$

$$P'(n,t) = -nP(n,t)\Theta - \mu P(n,t) + (1-p-q)nP(n,t)\Theta P + \mu P(n,t)P + q(n+1)P(n+1,t)\Theta P + p(n-1)P(n-1,t)\Theta P. \tag{7}$$

Similarly, we have

$$P'(1,t) = -P(1,t)\Theta - \mu P(1,t) + (1-p-q)P(1,t)\Theta P + \mu P(1,t)P + 2qP(2,t)\Theta P, \tag{8}$$
$$P'(0,t) = -\mu P(0,t) + \mu P(0,t)P + qP(1,t)\Theta P, \tag{9}$$

where $\Theta = diag\{\theta_1, \theta_2, \ldots, \theta_m\}$, $P = (p_{ij})_{m\times m}$ and $I = diag\{1,1,\ldots,1\}$.

Let the generating function of the matrix $P(n,t)$ be $P^*(z,t) = \sum_{n=0}^{\infty} P(n,t)z^n$. Since $\frac{\partial}{\partial t}P^*(z,t) = \sum_{n=0}^{\infty} P'(n,t)z^n$ and $\frac{\partial}{\partial z}P^*(z,t) = \sum_{n=1}^{\infty} nP(n,t)z^{n-1}$, with the help of Eqs. (7)–(9) we know that $P^*(z,t)$ satisfies the differential equation

$$\frac{\partial}{\partial t}P^*(z,t) - \frac{\partial}{\partial z}P^*(z,t)\Theta(-zI + (1-p-q)zP + qP + pz^2P) + \mu P^*(z,t)(I-P) = O \tag{10}$$

with boundary condition $P^*(z,0) = z^N I$, where $O$ is a $m \times m$ zero matrix.

The expectation of $N_t$ can be expressed by $P^*(z,t)$ as follows.

$$E[N_t|N_0=N]=\sum_{i=1}^{m}\Pr\{A_0=i\}E[N_t|N_0=N,A_0=i]=\sum_{i=1}^{m}p_i\sum_{n=0}^{\infty}\sum_{j=1}^{m}n\Pr\{N_t=n,A_t=j|N_0=N,A_0=i\}$$
$$=\sum_{i=1}^{m}p_i\sum_{n=0}^{\infty}\sum_{j=1}^{m}nP_{ij}(n,t)=\mathbf{p}_0\sum_{n=0}^{\infty}nP(n,t)\cdot\mathbf{1}^T=\mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}\cdot\mathbf{1}^T, \tag{11}$$

where $\mathbf{p}_0=(p_1,p_2,\ldots,p_m)$ and $\mathbf{1}=(1,1,\ldots,1)$.

In general, the solution $P^*(z,t)$ of (10) does not have an explicit expression, except for some special cases.

We consider a special case that all $\theta_i,\ i=1,\ 2,\ \ldots,\ m$ are equal to $\theta$, which means the change of action has no effect on the evolution of software testing process. It is the same as the case that there is only one action. Therefore, we can suppose $m=1$.

**Proposition 3.1.** *For the software testing process defined in Section 2 and in the case of $m=1$, it holds*

$$E[N_t|N_0=N]=Ne^{\theta(p-q)t},$$
$$Var[N_t|N_0=N]=\begin{cases}\frac{p+q}{p-q}Ne^{2\theta(p-q)t}-\frac{p+q}{p-q}Ne^{\theta(p-q)t}, & \text{if } p\neq q;\\ 2p\theta Nt, & \text{if } p=q.\end{cases}$$

**Proof.** Eq. (10) becomes

$$\frac{\partial}{\partial t}P^*(z,t)-\theta((-p-q)z+q+pz^2)\frac{\partial}{\partial z}P^*(z,t)=0 \tag{12}$$

with boundary condition $P^*(z,0)=z^N$.

The solution of (12) is

$$P^*(z,t)=\begin{cases}\left(\frac{1-\frac{qz-1}{p_z-\frac{q}{p}}e^{\theta(p-q)t}}{1-\frac{z-1}{z-\frac{q}{p}}e^{\theta(p-q)t}}\right)^N, & p>0,\quad p\neq q;\\ (1-(1-z)e^{-\theta qt})^N, & p=0,p\neq q;\\ \left(1-\frac{1}{\frac{1}{1-z}+\theta pt}\right)^N, & p>0,p=q,\end{cases}=\begin{cases}\left(\frac{1-\frac{qz-q}{pz-q}e^{\theta(p-q)t}}{1-\frac{pz-p}{pz-q}e^{\theta(p-q)t}}\right)^N, & p\neq q;\\ \left(1-\frac{1}{\frac{1}{1-z}+\theta pt}\right)^N, & p>0,p=q.\end{cases} \tag{13}$$

Therefore,

$$\frac{\partial}{\partial z}P^*(z,t)=\begin{cases}N\left(\frac{1-\frac{qz-q}{pz-q}e^{\theta(p-q)t}}{1-\frac{pz-p}{pz-q}e^{\theta(p-q)t}}\right)^{N-1}\frac{\frac{(p-q)^2}{(pz-q)^2}e^{\theta(p-q)t}}{\left[1-\frac{pz-p}{pz-q}e^{\theta(p-q)t}\right]^2}, & p\neq q;\\ N\left(1-\frac{1}{\frac{1}{1-z}+\theta pt}\right)^{N-1}\frac{1}{(\frac{1}{1-z}+\theta pt)^2}, & p>0,p=q.\end{cases} \tag{14}$$

So

$$E[N_t|N_0=N]=\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}=\begin{cases}Ne^{\theta(p-q)t}, & p\neq q;\\ N, & p>0,p=q,\end{cases}=Ne^{\theta(p-q)t}. \tag{15}$$

Similarly we can derive $E[N_t^2|N_0=N]$ and thus $Var[N_t|N_0=N]$.

In the above proof we have obtained the expression of $P^*(z,t)$ explicitly, but it is rather cumbersome to obtain $E[N_t]$ and $E[N_t^2]$. Next we give another relatively concise proof, which can also be used in Proposition 3.2.

**Another proof:** Differentiating with respect to $z$ on Eq. (12) and letting $z=1$, we have

$$\frac{\partial}{\partial t}\left(\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}\right)-(p-q)\theta\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}=0 \tag{16}$$

with boundary condition $\frac{\partial}{\partial z}P^*(z,0)\big|_{z=1}=N$.

Let $f(t)=\frac{\partial}{\partial z}P^*(z,t)\big|_{z=1}$. According to (11) we know that $f(t)=E[N_t|N_0=N]$. From (16), we have $f'(t)-(p-q)\theta f(t)=0$ and $f(0)=N$. Therefore, $E[N_t|N_0=N]=f(t)=Ne^{(p-q)\theta t}$.

Differentiating twice with respect to $z$ on Eq. (12) and letting $z = 1$, we have

$$\frac{\partial}{\partial t}\left(\frac{\partial^2}{\partial z^2}P^*(z,t)\Big|_{z=1}\right) - 2(p-q)\theta\frac{\partial^2}{\partial z^2}P^*(z,t)\Big|_{z=1} - 2p\theta\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1} = 0 \tag{17}$$

with boundary condition becomes $\frac{\partial^2}{\partial z^2}P^*(z,0)\Big|_{z=1} = N(N-1)$.

Therefore,

$$\frac{\partial^2}{\partial z^2}P^*(z,t)\Big|_{z=1} = \begin{cases} [N(N-1) + \frac{2pN}{p-q}(1 - e^{-\theta(p-q)t})]e^{2\theta(p-q)t}, & p \neq q; \\ N(N-1) + 2p\theta Nt, & p = q. \end{cases}$$

$$E[N_t^2|N_0 = N] = \frac{\partial^2}{\partial z^2}P^*(z,t)\Big|_{z=1} + \frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1} = \begin{cases} [N(N-1) + \frac{2pN}{p-q}]e^{2\theta(p-q)t} - \frac{p+q}{p-q}Ne^{\theta(p-q)t}, & p \neq q; \\ N^2 + 2p\theta Nt, & p = q. \end{cases}$$

Thus,

$$Var[N_t|N_0 = N] = E[N_t^2|N_0 = N] - E[N_t|N_0 = N]^2 = \begin{cases} \frac{p+q}{p-q}Ne^{2\theta(p-q)t} - \frac{p+q}{p-q}Ne^{\theta(p-q)t}, & p \neq q; \\ 2p\theta Nt, & p = q. \end{cases} \quad \square$$

Note that the variance of $N_t$ can be seen as a risk measure of the testing process [7], which is an important factor in the software development. From the above proposition we know how the variance of $N_t$ will evolve with the relationship of $p$ and $q$.

In general, it is impossible to obtain the explicit expression of $E[N_t]$ through Eq. (11). But we can obtain upper and lower bounds of $E[N_t]$. In the following argument we denote $\bar{\theta} = \max_{1\leqslant k\leqslant m}\{\theta_k\}$ and $\underline{\theta} = \min_{1\leqslant k\leqslant m}\{\theta_k\}$.

**Proposition 3.2.** *For the software testing process defined in Section 2, it holds*

$$\begin{aligned} &Ne^{(p-q)\bar{\theta}t} \leqslant E[N_t|N_0 = N] \leqslant Ne^{(p-q)\underline{\theta}t}, && \text{if } p < q; \\ &Ne^{(p-q)\underline{\theta}t} \leqslant E[N_t|N_0 = N] \leqslant Ne^{(p-q)\bar{\theta}t}, && \text{if } p > q; \\ &E[N_t|N_0 = N] = N, && \text{if } p = q. \end{aligned}$$

**Proof.** Differentiating with respect to $z$ on Eq. (10), letting z = 1, premultiplying by $\mathbf{p}_0$ and postmultiplying by $\mathbf{1}^T$, we have

$$\frac{\partial}{\partial t}\left(\mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}\cdot\mathbf{1}^T\right) - (p-q)\mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\Big|_{z=1}\Theta\cdot\mathbf{1}^T = 0. \tag{18}$$

Since $\underline{\theta}\mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\big|_{z=1}\cdot\mathbf{1}^T \leqslant \mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\big|_{z=1}\Theta\cdot\mathbf{1}^T \leqslant \bar{\theta}\mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\big|_{z=1}\cdot\mathbf{1}^T$ and $E[N_t|N_0 = N] = \mathbf{p}_0\frac{\partial}{\partial z}P^*(z,t)\big|_{z=1}\cdot\mathbf{1}^T$, the result follows from (18). $\square$

Although it is difficult to obtain the exact value of the expected residual defect number, the above proposition gives its upper and lower bounds, which is helpful for assessing the software reliability. The above proposition also tells us that in the case of $p < q$ the expected residual defect number will converge to 0 exponentially, in the case of $p = q$ the expected number will remain constant and in the case of $p > q$ the expected number will go to infinity exponentially. In realistic software testing process we always have $p < q$.

### 3.2. Dynamic behavior of $M_t$ and software failure intensity

Although $N_t$ is an important quantity for software reliability assessment, the value of $N_t$ cannot be observed immediately. However, we can estimate $N_t$ through the cumulative number of observed failures $M_t$. By a standard Chapman-Kolmogorov argument on the Markov process $Z_t = (N_t, A_t, M_t)$, we obtain

$$\begin{aligned} \frac{d}{dt}\Pr\{M_t = l, N_t = n, A_t = j\} = &-(n\theta_j + \mu)\Pr\{M_t = l, N_t = n, A_t = j\} + \sum_{i=1}^{m}\Pr\{M_t = l, N_t = n, A_t = i\}\mu p_{ij} \\ &+ \sum_{i=1}^{m}\Pr\{M_t = l-1, N_t = n, A_t = i\}(1-p-q)n\theta_i p_{ij} + \sum_{i=1}^{m}\Pr\{M_t = l-1, N_t = n-1, A_t = i\}p(n-1)\theta_i p_{ij} \\ &+ \sum_{i=1}^{m}\Pr\{M_t = l-1, N_t = n+1, A_t = i\}q(n+1)\theta_i p_{ij}. \end{aligned} \tag{19}$$

Summing over $n$, we have

$$\frac{d}{dt}\Pr\{M_t=l,A_t=j\}=-\mu\Pr\{M_t=l,A_t=j\}-\sum_{n=0}^{\infty}n\Pr\{M_t=l,N_t=n,A_t=j\}\theta_j+\mu\sum_{i=1}^{m}\Pr\{M_t=l,A_t=i\}p_{ij}$$
$$+\sum_{i=1}^{m}\sum_{n=0}^{\infty}n\Pr\{M_t=l-1,N_t=n,A_t=i\}\theta_i p_{ij}. \tag{20}$$

Therefore, $E[M_t;A_t=j]=\sum_{l=0}^{\infty}l\Pr\{M_t=l,A_t=j\}$ satisfies

$$\frac{d}{dt}E[M_t;A_t=j] \tag{21}$$

$$=-\mu E[M_t;A_t=j]-\sum_{l=0}^{\infty}\sum_{n=0}^{\infty}l n\Pr\{M_t=l,N_t=n,A_t=j\}\theta_j+\mu\sum_{i=1}^{m}E[M_t;A_t=i]p_{ij}+\sum_{l=0}^{\infty}\sum_{i=1}^{m}\sum_{n=0}^{\infty}l n\Pr\{M_t=l,N_t=n,A_t$$
$$=i\}\theta_i p_{ij}+\sum_{i=1}^{m}\sum_{n=0}^{\infty}n\Pr\{N_t=n,A_t=i\}\theta_i p_{ij}=-\mu E[M_t;A_t=j]-E[M_tN_t;A_t=j]\theta_j+\mu\sum_{i=1}^{m}E[M_t;A_t$$
$$=i]p_{ij}+\sum_{i=1}^{m}E[M_tN_t;A_t=i]\theta_i p_{ij}+\sum_{i=1}^{m}\sum_{n=0}^{\infty}E[N_t;A_t=i]\theta_i p_{ij}. \tag{22}$$

Now we let

$$\alpha_j(t)=E[M_t;A_t=j],\quad \alpha(t)=(\alpha_1(t),\alpha_2(t),\ldots,\alpha_m(t));$$
$$\beta_j(t)=E[N_t;A_t=j],\quad \beta(t)=(\beta_1(t),\beta_2(t),\ldots,\beta_m(t));$$
$$\gamma_j(t)=E[M_tN_t;A_t=j],\quad \gamma(t)=(\gamma_1(t),\gamma_2(t),\ldots,\gamma_m(t)).$$

Then (22) has a vector form:

$$\alpha'(t)=-\mu\alpha(t)-\gamma(t)\Theta+\mu\alpha(t)P+\gamma(t)\Theta P+\beta(t)\Theta P. \tag{23}$$

Similarly, if we let $\varphi_j(t)=E[N_t^2;A_t=j]$, $\varphi(t)=(\varphi_1(t),\varphi_2(t),\ldots,\varphi_m(t))$, we also have

$$\beta'(t)=-\mu\beta(t)-\varphi(t)\Theta+\mu\beta(t)P+\varphi(t)\Theta P+(p-q)\beta(t)\Theta P. \tag{24}$$

Both postmultiplying Eqs. (23) and (24) by $\mathbf{1}^T$, we have

$$\alpha'(t)\cdot\mathbf{1}^T=\beta(t)\Theta\cdot\mathbf{1}^T, \tag{25}$$

$$\beta'(t)\cdot\mathbf{1}^T=(p-q)\beta(t)\Theta\cdot\mathbf{1}^T. \tag{26}$$

Therefore, $((q-p)\alpha'(t)+\beta'(t))\cdot\mathbf{1}^T=0$, i.e., $((q-p)\alpha(t)+\beta(t))\cdot\mathbf{1}^T$ is a constant.

Note that $E[M_t]=\sum_{j=1}^{m}E[M_t;A_t=j]=\alpha(t)\cdot\mathbf{1}^T$, $E[N_t]=\sum_{j=1}^{m}E[N_t;A_t=j]=\beta(t)\cdot\mathbf{1}^T$, so $(q-p)E[M_t]+E[N_t]$ is a constant, and it is equal to $(q-p)E[M_0]+E[N_0]=N$.

Here is another explanation that $(q-p)E[M_t]+E[N_t]$ is a constant: Once a failure is observed, which means $M_t$ is increased by 1, the residual defects $N_t$ is increased by 1 with probability $p$, decreased by 1 with probability $q$ and unchanged with probability $1-p-q$. So $(q-p)E[M_t]+E[N_t]$ remains invariant.

Thus if $p\neq q$,

$$E[M_t]=\frac{1}{q-p}(N-E[N_t]). \tag{27}$$

Therefore, we can estimate the expected number of the residual defect $E[N_t]$ through the expected cumulative number of observed failures $E[M_t]$ by the above equation, which is more suitable for practical use in testing process.

The expectation of $M_t$ is difficult to obtain in general, but for some special case we can easily get the value of $E[M_t]$.

**Example 3.1.** The case that all $\theta_i,\ i=1,\ 2,\ \ldots,\ m$ are equal to $\theta$.

In this case, Eq. (23) becomes

$$\alpha'(t)=\theta\beta(t) \tag{28}$$

Since $\beta(t)=E[N_t]=Ne^{\theta(p-q)t}$ and $\alpha(0)=E[M_0]=0$, we have

$$E[M_t]=\alpha(t)=\begin{cases}\frac{N}{q-p}(1-e^{\theta(p-q)t}), & p\neq q;\\ \theta Nt, & p=q.\end{cases}$$

In general, we have lower and upper bounds of $E[M_t]$, which are stated below.

**Proposition 3.3.** *For the software testing process defined in Section 2, it holds*

$$\begin{aligned} &\tfrac{N}{q-p}(1-e^{(p-q)\underline{\theta}t}) \leqslant E[M_t|N_0=N] \leqslant \tfrac{N}{q-p}(1-e^{(p-q)\bar{\theta}t}), \quad \text{if } p\neq q;\\ &\underline{\theta}Nt \leqslant E[M_t|N_0=N] \leqslant \bar{\theta}Nt, \qquad\qquad\qquad\quad \text{if } p=q. \end{aligned}$$

**Proof.** If $p \neq q$, then from (27) and Proposition 3.2 we have

$$\frac{N}{q-p}(1-e^{(p-q)\underline{\theta}t}) \leqslant E[M_t|N_0=N] \leqslant \frac{N}{q-p}(1-e^{(p-q)\bar{\theta}t}).$$

From (25) we have

$$\underline{\theta}\beta(t)\cdot\mathbf{1}^T \leqslant \alpha'(t)\cdot\mathbf{1}^T \leqslant \bar{\theta}\beta(t)\cdot\mathbf{1}^T.$$

According to the result in Proposition 3.2 we have $\beta(t)\cdot\mathbf{1}^T = E[N_t|N_0=N] = N$. Therefore,

$$\underline{\theta}Nt \leqslant E[M_t|N_0=N] \leqslant \bar{\theta}Nt. \qquad \square$$

The above proposition says that in the case of $p < q$ the expected observed failure number will converge to a constant exponentially, in the case of $p > q$ the expected number will go to infinity exponentially and in the case of $p = q$ the expected number will increase linearly.

Software failure intensity is defined as $\lambda_t = \frac{dE[M_t]}{dt}$. Since that $\frac{dE[M_t]}{dt} = \frac{d\alpha(t)\cdot\mathbf{1}^T}{dt} = \alpha'(t)\cdot\mathbf{1}^T = \beta(t)\Theta\cdot\mathbf{1}^T$ and $E[N_t] = \beta(t)\cdot\mathbf{1}^T$, from Proposition 3.2 we have

$$\begin{aligned} &N\underline{\theta}e^{(p-q)\bar{\theta}t} \leqslant \lambda_t \leqslant N\bar{\theta}e^{(p-q)\underline{\theta}t}, \quad \textit{if } p<q;\\ &N\underline{\theta}e^{(p-q)\underline{\theta}t} \leqslant \lambda_t \leqslant N\bar{\theta}e^{(p-q)\bar{\theta}t}, \quad \textit{if } p>q;\\ &N\underline{\theta} \leqslant \lambda_t \leqslant N\bar{\theta}, \quad \textit{if } p=q. \end{aligned}$$

## 4. Some interesting distributions

In this section we study some distributions often considered in software reliability, including distribution of the time between software failures, software reliability, and distribution of the first passage time to the specified number of residual defects.

### 4.1. Software reliability

Let $R(x|t)$ denote the probability that there is no defect detected during the time interval $(t, t+x)$.

Define $S_t = \inf\{s > 0 : M_{t+s} \neq M_t\}$, which means the time length of finding first defect after time $t$, then we have

$$R(x|t) = \Pr\{S_t > x\} = \sum_{n=0}^{\infty}\Pr\{N_t=n\}\Pr\{S_t > x|N_t=n\}.$$

Let $R_n(x|t) = \Pr\{S_t > x|N_t=n\}$. $R_n(x|t)$ is the probability that no failure is observed in the time interval $(t, t+x)$ on the condition that there are $n$ defects at time $t$. Define

$$r_n(x|t) = -\frac{\frac{dR_n(x|t)}{dx}}{R_n(x|t)} = -\frac{d}{dx}\ln R_n(x|t).$$

$r_n(\cdot|t)$ is conditional interval failure rate function at time $t$.

**Proposition 4.1.** *For the software testing process defined in Section 2, it holds*

$$n\underline{\theta} \leqslant r_n(x|t) \leqslant n\bar{\theta},$$

*and*

$$E[e^{-Nt\bar{\theta}x}] \leqslant R(x|t) \leqslant E[e^{-Nt\underline{\theta}x}].$$

**Proof.** From the distribution of $T_n^2$ we know that if there are $n$ defects in software systems and the testing action taken is $i$, then the rate of detecting a defect is $n\theta i$.

$$\begin{aligned} R_n(x+dx|t) - R_n(x|t) &= -\Pr\{x < S_t \leqslant x+dx|N_t=n\} = -\Pr\{S_t > x|N_t=n\}\Pr\{S_t \leqslant x+dx|N_t=n, S_t>x\}\\ &= -R_n(x|t)nE[\theta_{i(t+x)}|N_t=n, S_t>x]dx, \end{aligned}$$

where $i(t+x)$ denotes the action taken at time instant $t+x$.

Therefore, $r_n(x|t) = nE[\theta_{i(t+x)}|N_t = n, S_t > x]$. Since $\underline{\theta} \leqslant \theta_{i(t+x)} \leqslant \bar{\theta}$, we have

$$n\underline{\theta} \leqslant r_n(x|t) \leqslant n\bar{\theta}.$$

Thus, $n\underline{\theta} \leqslant -\frac{d}{dx}ln(R_n(x|t)) \leqslant n\bar{\theta}$. That is,

$$e^{-n\bar{\theta}x} \leqslant R_n(x|t) = \Pr\{S_t > x|N_t = n\} \leqslant e^{-n\underline{\theta}x}.$$

Therefore

$$E[e^{-Nt\bar{\theta}x}] = \sum_{n=0}^{\infty}\Pr\{N_t = n\}e^{-n\bar{\theta}x} \leqslant R(x|t) \leqslant \sum_{n=0}^{\infty}\Pr\{N_t = n\}e^{-n\underline{\theta}x} = E[e^{-Nt\underline{\theta}x}]. \qquad \square$$

It is known that when $t = 0, R(x|t)$ is the traditional reliability. If one assumes that $N_0 = N$, then $R(x|t) \in [e^{-N\bar{\theta}x}, e^{-N\underline{\theta}x}]$. When $t \neq 0$, $R(x|t)$ is the reliability of the software testing processes at time $t$. Therefore, $R(x|t)$ can be viewed as a generalization of traditional reliability. Proposition 4.1 gives upper and lower bounds of $R(x|t)$. Furthermore, $\bar{\theta}$ and $\underline{\theta}$ can be obtained by statistical observation.

The result can be comprehended intuitively: If at time instant $t$ there are $n$ defects remaining in the software system, then the rate of detecting a defect is at least $n\underline{\theta}$, and at most $n\bar{\theta}$. Therefore, the result of Proposition 4.1 can be expected.

### 4.2. Distribution of the time between software failures

Let $S_i$ $(i = 1, 2, \ldots)$ be time interval between the $(i-1)$th and the $i$th software failure-occurrences and $\overline{N}_i, \overline{A}_i$ be the number of the defects remaining and the taken action after the $i$th software failure-occurrence. Then we have

$$\Pr\{\overline{N}_i - \overline{N}_{i-1} = x\} = \begin{cases} p, & \text{if} x = -1; \\ q, & \text{if} x = 1; \\ 1-p-q, & \text{if} x = 0; \\ 0, & \text{otherwise}. \end{cases} \tag{29}$$

Let $a \vee b = \max\{a, b\}$ and $B(i,n) = \sum_{0\vee(-n)\leqslant k\leqslant\frac{i-n}{2}} \frac{i!}{k!(k+n)!(i-2k-n)!}p^k q^{k+n}(1-p-q)^{i-2k-n}$, then we have an explicit expression of $\Pr\{\overline{N}_i = n\}$.

**Proposition 4.2.** *For $i \geqslant 0,\ n \in \mathcal{N}$*

$$\Pr\{\overline{N}_i = n\} = \begin{cases} B(i, n-N) - B(i, -n-N), & n \geqslant 1; \\ q\sum_{k=0}^{i-1}\Pr\{\overline{N}_k = 1\}, & n = 0. \end{cases}$$

**Proof.** First we prove that $B(i,n)$ is the probability of holding the equation $x_1 + x_2 + \cdots + x_i = n$, where random variables $x_1,\ x_2,\ \ldots,\ x_i$ are i.i.d and have the same distribution as random variable $X$, which satisfies $\Pr\{X = 1\} = p$, $\Pr\{X = -1\} = q$ and $\Pr\{X = 0\} = 1 - p - q$.

If the number of $-1$ in $\{x_1, x_2, \ldots, x_i\}$ is $k$, then in order to satisfy $x_1 + x_2 + \cdots + x_i = n$, the number of 1 in $\{x_1, x_2, \ldots, x_i\}$ must be $k + n$, and the number of 0 must be $i - 2k - n$. Noting that all these numbers should be nonnegative, i.e., $k \geqslant 0,\ k + n \geqslant 0$ and $i - 2k - n \geqslant 0$, we know that the probability of $x_1 + x_2 + \cdots + x_i = n$ is

$$\sum_{0\vee(-n)\leqslant k\leqslant\frac{i-n}{2}} \frac{i!}{k!(k+n)!(i-2k-n)!}p^k q^{k+n}(1-p-q)^{i-2k-n} = B(i,n).$$

Let $\Delta_i = \overline{N}_i - \overline{N}_{i-1}$ $(i \geqslant 1)$.

First we consider the case of $n \geqslant 1$. As $\overline{N}_0 = N$, we have $\sum_{j=1}^{i}\Delta_j = n - N$, where $\Delta_j$ is either $\pm 1$ or 0. Note that if $\overline{N}_k = 0$ for some $k$, then we have $\overline{N}_i = 0$ for $\forall i > k$ and if $\overline{N}_k > 0$ then $\Delta_{k+1}$ has the same distribution with $X$. Therefore, we have $\overline{N}_k > 1,\ 1 \leqslant k \leqslant i$, and then $\overline{N}_i = n$ is equivalent to.

- $\sum_{j=1}^{i}\Delta_j = n - N$.
- $\sum_{j=1}^{k}\Delta_j > -N,\ \forall 1 \leqslant k \leqslant i$.

By the reflect principle we know that $\Pr\{\overline{N}_i = n\} = B(i, n-N) - B(i, -n-N)$.

Now we calculate $\Pr\{\overline{N}_i = 0\}$. $\overline{N}_i = 0$ means there is some $k,\ 1 \leqslant k \leqslant i$ such that $\overline{N}_k = 0$ but $\overline{N}_{k-1} \neq 0$. Then we must have $\overline{N}_{k-1} = 1$. Therefore, we have $\Pr\{\overline{N}_i = 0\} = \sum_{k=1}^{i}\Pr\{\overline{N}_k = 0, \overline{N}_{k-1} = 1\} = \sum_{k=1}^{i}\Pr\{\overline{N}_{k-1} = 1\}\Pr\{\overline{N}_k = 0|\overline{N}_{k-1} = 1\} = q\sum_{k=0}^{i-1}\Pr\{\overline{N}_k = 1\}$. $\square$

Next we calculate the conditional distribution of $S_k$, given that $\overline{N}_{k-1} = n,\ \overline{A}_{k-1} = i$.

Let $\Phi_{n,i}(x) = \Pr\{S_k \leqslant x | \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}$ and $\Psi_{n,i}(x) = \Pr\{T_k \leqslant x | \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}$. From the distribution of $T_k^1$, $T_k^2$ and $T_k = \min\{T_k^1, T_k^2\}$, it can be calculated that $\Psi_{n,i}(x) = 1 - e^{-(n\theta_i+\mu)x}$. Then we have

$$\Phi_{n,i}(x) = \Psi_{n,i}(x) * \left( \frac{n\theta_i}{n\theta_i + \mu} + \frac{\mu}{n\theta_i + \mu} \sum_{j=1}^{m} p_{ij} \Phi_{n,j}(x) \right), \tag{30}$$

where $*$ denotes a Laplace–Stieltjes convolution.

By applying the Laplace–Stieltjes (L–S) transform to (30) and noting that the Laplace–Stieltjes (L–S) transform of $\Psi_{n,i}(x)$ is $\frac{n\theta_i+\mu}{n\theta_i+\mu+s}$, we know that the L–S transform of $\Phi_{n,i}(x)$ (denoted by $\tilde{\Phi}_{n,i}(s)$) satisfies

$$\tilde{\Phi}_{n,i}(s) = \frac{n\theta_i + \mu}{n\theta_i + \mu + s} \cdot \left( \frac{n\theta_i}{n\theta_i + \mu} + \frac{\mu}{n\theta_i + \mu} \sum_{j=1}^{m} p_{ij} \tilde{\Phi}_{n,j}(s) \right). \tag{31}$$

If we can obtain the explicit expression $\tilde{\Phi}_{n,i}(s)$ through Eq. (31), then by Laplace transform inversion we can obtain $\Phi_{n,i}(x) = \Pr\{S_k \leqslant x | \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}$. But when $m$ is quite large, it is usually impossible to find an explicit expression of $\tilde{\Phi}_{n,i}(s)$. Next we derive lower and upper bounds of $\Phi_{n,i}(x)$.

**Proposition 4.3.** *For the software testing process defined in Section 2, it holds*

$$e^{-n\bar{\theta}x} \leqslant \Pr\{S_k > x | \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\} \leqslant e^{-n\underline{\theta}x}, \quad \forall 1 \leqslant i \leqslant m, \tag{32}$$

*and*

$$e^{-n\bar{\theta}x} \leqslant \Pr\{S_k > x | \overline{N}_{k-1} = n\} \leqslant e^{-n\underline{\theta}x}. \tag{33}$$

**Proof.** Fix $x$, and select $i = i(x)$ such that $\Phi_{n,i}(x) = \max_{1\leqslant k\leqslant m}\{\Phi_{n,k}(x)\} \triangleq \Phi_n(x)$. Then from (30) we have

$$\begin{aligned}
\Phi_n(x) = \Phi_{n,i(x)}(x) &= \frac{n\theta_{i(x)}}{n\theta_{i(x)} + \mu}(1 - e^{-(n\theta_{i(x)}+\mu)x}) + \frac{\mu}{n\theta_{i(x)} + \mu} \sum_{j=1}^{m} p_{ij} \int_0^x (n\theta_{i(x)} + \mu) e^{-(n\theta_{i(x)}+\mu)(x-t)} \Phi_{n,j}(t) dt \\
&\leqslant \frac{n\theta_{i(x)}}{n\theta_{i(x)} + \mu}(1 - e^{-(n\theta_{i(x)}+\mu)x}) + \mu \int_0^x e^{-(n\theta_{i(x)}+\mu)(x-t)} \Phi_n(t) dt,
\end{aligned}$$

Therefore

$$1 - \Phi_n(x) \geqslant e^{-(n\theta_{i(x)}+\mu)x} + \mu \int_0^x e^{-(n\theta_{i(x)}+\mu)(x-t)} (1 - \Phi_n(t)) dt \geqslant e^{-(n\bar{\theta}+\mu)x} + \mu \int_0^x e^{-(n\bar{\theta}+\mu)(x-t)} (1 - \Phi_n(t)) dt.$$

Let $g(x) = (1 - \Phi_n(x)) e^{(n\bar{\theta}+\mu)x}$, then from the above inequality we obtain

$$g(x) \geqslant 1 + \mu \int_0^x g(t) dt.$$

So $g(x) \geqslant e^{\mu x}$. Therefore, $\Phi_n(x) \leqslant 1 - e^{n\bar{\theta}x}$. Further more, we have

$$\Phi_{n,i}(x) \leqslant 1 - e^{n\bar{\theta}x}$$

and

$$\begin{aligned}
\Pr\{S_k \leqslant x | \overline{N}_{k-1} = n\} &= \frac{\Pr\{S_k \leqslant x, \overline{N}_{k-1} = n\}}{\Pr\{\overline{N}_{k-1} = n\}} = \frac{\sum_{i=1}^{m} \Pr\{S_k \leqslant x, \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}}{\sum_{i=1}^{m} \Pr\{\overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}} = \frac{\sum_{i=1}^{m} \Phi_{n,i}(x) \Pr\{\overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}}{\sum_{i=1}^{m} \Pr\{\overline{N}_{k-1} = n, \overline{A}_{k-1} = i\}} \\
&\leqslant 1 - e^{n\bar{\theta}x}.
\end{aligned}$$

By a similar argument we have $\Phi_{n,i}(x) \geqslant 1 - e^{n\underline{\theta}x}$ and $\Pr\{S_k \leqslant x | \overline{N}_{k-1} = n\} \geqslant 1 - e^{-n\underline{\theta}x}$.

So

$$e^{-n\bar{\theta}x} \leqslant \Pr\{S_k > x | \overline{N}_{k-1} = n, \overline{A}_{k-1} = i\} \leqslant e^{-n\underline{\theta}x}, \ \forall 1 \leqslant i \leqslant m,$$

and

$$e^{-n\bar{\theta}x} \leqslant \Pr\{S_k > x | \overline{N}_{k-1} = n\} \leqslant e^{-n\underline{\theta}x}. \qquad \square$$

According to the above proposition we know that the reliability function for $S_i$ is

$$\Pr\{S_i > x\} = \sum_{n=0}^{\infty} \Pr\{S_i > x | \overline{N}_{i-1} = n\} \cdot \Pr\{\overline{N}_{i-1} = n\}. \tag{34}$$

Therefore, we can derive upper and lower bounds of the reliability function for $S_i$:

$$\sum_{n=0}^{\infty}\Pr\{\overline{N}_{i-1}=n\}e^{-n\bar{\theta}x}\leqslant\Pr\{S_i>x\}\leqslant\sum_{n=0}^{\infty}\Pr\{\overline{N}_{i-1}=n\}e^{-n\underline{\theta}x}.$$

**Example 4.1.** The case that all $\theta_i,\ i=1,\ 2,\ \ldots,\ m$ are equal to $\theta$.

The reliability function for *Si* is

$$\Pr\{S_i>x\}=\sum_{n=0}^{\infty}\Pr\{S_i>x|\overline{N}_{i-1}=n\}\Pr\{\overline{N}_{i-1}=n\}=\sum_{n=0}^{\infty}\Pr\{\overline{N}_{i-1}=n\}e^{-n\theta x}.$$

We define embedded failure rate as

$$r_n(x|i)=-\frac{\frac{d\Pr\{S_i>x|\overline{N}_{i-1}=n\}}{dx}}{\Pr\{S_i>x|\overline{N}_{i-1}=n\}}.$$

Similar to the proof of Proposition 4.1, we have

$$n\underline{\theta}\leqslant r_n(x|i)\leqslant n\bar{\theta}.$$

### 4.3. Distribution of the first passage time to the specified number of residual defects

Define $L_n=\inf\{t\geqslant 0: N_t=n\}$ and $G_{(k,i),n}(t)=\Pr\{L_n\leqslant t|N_0=k,A_0=i\}$, then by considering the first transition of Markov process $Y_t$ we obtain the following equation:

$$G_{(k,i),n}(t)=\Psi_{k,i}(t)*\left(\frac{k\theta_i}{k\theta_i+\mu}\sum_{j=1}^{m}p_{ij}[pG_{(k+1,j),n}(t)+(1-p-q)G_{(k,j),n}(t)+qG_{(k-1,j),n}(t)]+\frac{\mu}{k\theta i+\mu}\sum_{j=1}^{m}p_{ij}G_{(k,j),n}(t)\right). \tag{35}$$

Applying the Laplace–Stieltjes (L–S) transform to (35), we have

$$\widetilde{G}_{(k,i),n}(s)=\frac{k\theta_i+\mu}{k\theta_i+\mu+s}\cdot\left(\frac{k\theta_i}{k\theta_i+\mu}\sum_{j=1}^{m}p_{ij}[p\widetilde{G}_{(k+1,j),n}(s)+(1-p-q)\widetilde{G}_{(k,j),n}(s)+q\widetilde{G}_{(k-1,j),n}(s)]+\frac{\mu}{k\theta_i+\mu}\sum_{j=1}^{m}p_{ij}\widetilde{G}_{(k,j),n}(s)\right). \tag{36}$$

Note that $\widetilde{G}_{(k,i),n}(0)=1$ and $\widetilde{G}'_{(k,i),n}(0)=-E[L_n|N_0=k,A_0=i]$, so differentiating with respect to $s$ on Eq. (36) and letting $s=0$, we can obtain an expression of $E[L_n|N_0=k,A_0=i]$:

$$k\Theta h_n(k)+\mu h_n(k)=\mathbf{1}^T+qk\Theta Ph_n(k-1)+pk\Theta Ph_n(k+1)+(1-p-q)k\Theta Ph_n(k)+\mu Ph_n(k), \tag{37}$$

where we use $h_n(k)$ to denote $(E[L_n|N_0=k,A_0=1],E[L_n|N_0=k,A_0=2],\ldots,E[L_n|N_0=k,A_0=m])^T$.

In general, it is impossible to obtain the explicit expression of $\Pr\{L_n\leqslant t|N_0=N\}$. Next we give an explicit expression in a special case and later give upper and lower bounds in general.

**Example 4.2.** The case that all $\theta_i,\ i=1,\ 2,\ \ldots,\ m$ are equal to $\theta$, i.e. $m=1$ and $p=0$.

Let $G_{k,n}(t)$ be the counterpart of $G_{(k,i),n}(t)$, then through (36) we obtain $\widetilde{G}_{k,n}(s)=\frac{qk\theta}{qk\theta+s}\widetilde{G}_{k-1,n}(s)$. Therefore,

$$\widetilde{G}_{k,n}(s)=\prod_{i=n+1}^{k}\frac{qi\theta}{qi\theta+s}=\sum_{i=n+1}^{k}A_i\frac{qi\theta}{qi\theta+s}, \tag{38}$$

where $A_i=\prod_{\substack{j=n+1\\ j\neq i}}^{k}\frac{j}{j-i}$.

By performing the inverse L–S transform of (38) we have

$$G_{k,n}(t)=\Pr\{L_n\leqslant t|N_0=k\}=\sum_{i=n+1}^{k}A_i[1-e^{-qi\theta t}].$$

Consider an auxiliary process $Y'(t)$ of $Y_t=(A_t,N_t)$ whose parameter $\theta_i$ is replaced by $\bar{\theta}=\max_{1\leqslant k\leqslant m}\{\theta_k\}$. Then the $Q$-matrix of process $Y'(t)$ is

$$Q'_2((i,n),(j,m))=\begin{cases}qn\bar{\theta}p_{ij}, & m=n-1;\\ pn\bar{\theta}p_{ij}, & m=n+1;\\ ((1-p-q)n\bar{\theta}+\mu)p_{ij}, & m=n,j\neq i;\\ -(n\bar{\theta}+\mu)+((1-p-q)n\bar{\theta}+\mu)p_{ii}, & m=n,j=i,\end{cases}\quad(n\geqslant 1) \tag{39}$$

and

$$Q_2'((i,0),(j,0)) = \begin{cases} \mu p_{ij}, & j \neq i; \\ -\mu + \mu p_{ii}, & j = i. \end{cases} \tag{40}$$

Let $N_n'$, $\overline{N}_n'$, $A_n'$, $\overline{A}_n'$, $\tau_n'$, ... be the counterparts of $N_n$, $\overline{N}_n$, $A_n$, $\overline{A}_n$, $\tau_n$, ... in the process $Y'(t)$, respectively.

Similarly, consider another auxiliary process $Y''(t)$ whose parameter $\theta_i$ is replaced by $\underline{\theta} = \min_{1\leqslant k\leqslant m}\{\theta_k\}$ and let $N_n''$, $\overline{N}_n''$, $A_n''$, $\overline{A}_n''$, $\tau_n''$, ... be the counterparts of $N_n$, $\overline{N}_n$, $A_n$, $\overline{A}_n$, $\tau_n$, ... in the process $Y''(t)$, respectively. Then we have upper and lower bounds of $\Pr\{L_n \leqslant t | N_0 = N\}$ as the following proposition demonstrates.

**Proposition 4.4.** *For the software testing process defined in Section 2, it holds*

$$\Pr\{L_n'' \leqslant t | N_0'' = N\} \leqslant \Pr\{L_n \leqslant t | N_0 = N\} \leqslant \Pr\{L_n' \leqslant t | N_0' = N\}. \tag{41}$$

**Proof.** Let $\tau_n = \inf\{k \geqslant 0 : \overline{N}_k = n\}$, then by the definition of $L_n$ we have $L_n = \sum_{i=1}^{\tau_n} S_i$.

Obviously, we know that $(\overline{N}_n')_{n=0,1,\ldots}$ has the same distribution as $(\overline{N}_n)_{n=0,1,\ldots}$. So $\tau_n'$ and $\tau_n$ have the same distribution. Therefore, from (32) we know that

$$\begin{aligned}
\Pr\{S_1+S_2\leqslant x|\overline{N}_0=n_0,\overline{N}_1=n_1\} &= \sum_{j=1}^{m}\int_0^x \Pr\left\{S_2\leqslant x-y|\overline{N}_0=n_0,\overline{N}_1=n_1,\overline{A}_1=j\right\}\Pr\left\{\overline{A}_1=j|S_1=y,\overline{N}_0=n_0\right\}d\Pr\{S_1\leqslant y|\overline{N}_0=n_0\} \\
&\leqslant \sum_{j=1}^{m}\int_0^x \left(1-e^{-n_1\bar{\theta}(x-y)}\right)\Pr\left\{\overline{A}_1=j|S_1=y,\overline{N}_0=n_0\right\}d\Pr\{S_1\leqslant y|\overline{N}_0=n_0\} \\
&= \int_0^x \left(1-e^{-n_1\bar{\theta}(x-y)}\right)d\Pr\{S_1\leqslant y|\overline{N}_0=n_0\} = \int_0^x \Pr\{S_1\leqslant x-y|\overline{N}_0=n_0\}d\left(1-e^{-n_1\bar{\theta}y}\right) \\
&\leqslant \int_0^x e^{-n_0\bar{\theta}(x-y)}d\left(1-e^{-n_1\bar{\theta}y}\right) = \Pr\{S_1'+S_2'\leqslant x|\overline{N}_0'=n_0,\overline{N}_1'=n_1\}.
\end{aligned}$$

By similar method we have

$$\Pr\left\{\sum_{i=1}^{k}S_i\leqslant x|\overline{N}_0=n_0,\overline{N}_1=n_1,\ldots,\overline{N}_{k-1}=n_{k-1}\right\} \leqslant \Pr\left\{\sum_{i=1}^{k}S_i'\leqslant x|\overline{N}_0'=n_0,\overline{N}_1'=n_1,\ldots,\overline{N}_{k-1}'=n_{k-1}\right\}.$$

Therefore, we know

$$\begin{aligned}
\Pr\{L_n\leqslant t|N_0=N\} &= \Pr\left\{\sum_{i=1}^{\tau_n}S_i\leqslant t|N_0=N\right\} = \sum_{k=1}^{\infty}\Pr\left\{\sum_{i=1}^{k}S_i\leqslant t|N_0=N,\tau_n=k\right\}\Pr\{\tau_n=k|N_0=N\} \\
&= \sum_{k=1}^{\infty}\sum_{n_0,n_1,\ldots,n_k}\Pr\left\{\sum_{i=1}^{k}S_i\leqslant t|N_0=N,\tau_n=k,\overline{N}_0=n_0,\overline{N}_0=n_1,\ldots,\overline{N}_k=n_k\right\}\times\Pr\{\overline{N}_0=n_1,\ldots,\overline{N}_k=n_k|N_0=N,\tau_n \\
&= k\}\Pr\{\tau_n=k|N_0=N\} \\
&\leqslant \sum_{k=1}^{\infty}\sum_{n_0,n_1,\ldots,n_k}\Pr\left\{\sum_{i=1}^{k}S_i'\leqslant t|N_0'=N,\tau_n'=k,\overline{N}_0'=n_0,\overline{N}_1'=n_1,\ldots,\overline{N}_k'=n_k\right\} \\
&\quad\times\Pr\{\overline{N}_0'=n_1,\ldots,\overline{N}_k'=n_k|N_0'=N,\tau_n'=k\}\Pr\{\tau_n'=k|N_0'=N\} \\
&= \Pr\{L_n'\leqslant t|N_0'=N\}.
\end{aligned}$$

Similarly, we can obtain $\Pr\{L_n \leqslant t|N_0 = N\} \geqslant \Pr\{L_n'' \leqslant t|N_0'' = N\}$.

Therefore, we have derived upper and lower bounds of $\Pr\{L_n \leqslant t|N_0 = N\}$:

$$\Pr\{L_n''\leqslant t|N_0''=N\}\leqslant\Pr\{L_n\leqslant t|N_0=N\}\leqslant\Pr\{L_n'\leqslant t|N_0'=N\}. \quad \square$$

The above proposition implies that although usually it is hard to obtain explicit expressions of the quantities, we can derive their upper and lower bounds by considering another process with only one action. The analysis becomes much easier, and based on the analysis of one-action case we can obtain a general result.

Next we consider the expectation of $L_n$. For simplicity, we only consider the case of $n = 0$.

Let $L = L_0 = \inf\{t \geqslant 0 : N_t = 0\}$ and $h(n) = h_0(n) = (E[L|N_0 = n, A_0 = 1], E[L|N_0 = n, A_0 = 2], \ldots, E[L|N_0 = n, A_0 = m])^T$. From (37),

$$\begin{aligned}
&n\Theta h(n) + \mu h(n) = \mathbf{1}^T + qn\Theta Ph(n-1) + pn\Theta Ph(n+1) + (1-p-q)n\Theta Ph(n) + \mu Ph(n), \\
&h(0) = \mathbf{0}^T.
\end{aligned} \tag{42}$$

**Proposition 4.5.** *For the software testing process defined in Section 2, it holds*

$E[L|N_0 = N] < \infty$ if and only if $p < q$.

**Proof.** By the upper and lower bounds of $\Pr\{L \leqslant x|N_0 = N\}$ (see (41)) and $E[L|N_0 = N] = \int_0^\infty \Pr\{L > x|N_0 = N\}dx$, we know that $E[L'|N_0' = N] \leqslant E[L|N_0 = N] \leqslant E[L''|N_0'' = N]$. So we only need to consider the case of $m = 1$. In this case, $E[L|N_0 = N] = \sum_{i=1}^m \Pr\{A_0 = i\}E[L|A_0 = i, N_0 = N] = \sum_{i=1}^m p_i h(i, N) = \mathbf{p}_0 h(N) = h(N)$, and from (42),

$$ph(n+1) - (p+q)h(n) + qh(n-1) + \frac{1}{n\theta} = 0.$$

Therefore

$$h(n+1) - h(n) = \frac{q}{p}[h(n) - h(n-1)] - \frac{1}{pn\theta} = \left(\frac{q}{p}\right)^n [h(1) - h(0)] - \sum_{i=1}^n \frac{1}{pn\theta}\left(\frac{q}{p}\right)^{n-i} = \left(\frac{q}{p}\right)^n h(1) - \sum_{i=1}^n \frac{1}{pn\theta}\left(\frac{q}{p}\right)^{n-i}.$$

In the case of $p \neq q$, we have

$$h(n) = \sum_{k=1}^n [h(k) - h(k-1)] = \sum_{k=1}^n \left[\left(\frac{q}{p}\right)^{k-1} h(1) - \sum_{i=1}^{k-1} \frac{1}{pi\theta}\left(\frac{q}{p}\right)^{k-1-i}\right] = \frac{\left(\frac{q}{p}\right)^n - 1}{\frac{q}{p} - 1} h(1) - \sum_{i=1}^{n-1} \frac{1}{pi\theta} \sum_{k=0}^{n-i-1} \left(\frac{q}{p}\right)^k$$

$$= \frac{\left(\frac{q}{p}\right)^n - 1}{\frac{q}{p} - 1} h(1) - \sum_{i=1}^{n-1} \frac{1}{pi\theta} \frac{\left(\frac{q}{p}\right)^{n-i} - 1}{\frac{q}{p} - 1}. \tag{43}$$

When $p < q$, the above equation yields

$$\frac{h(n)}{\left(\frac{q}{p}\right)^n - 1} = \frac{h(1)}{\frac{q}{p} - 1} - \sum_{i=1}^{n-1} \frac{1}{pi\theta} \frac{\left(\frac{q}{p}\right)^{n-i} - 1}{\left(\frac{q}{p}\right)^n - 1} \cdot \frac{1}{\frac{q}{p} - 1}. \tag{44}$$

Since $\sum_{i=1}^{n-1} \frac{1}{pi\theta} \frac{\left(\frac{q}{p}\right)^{n-i} - 1}{\left(\frac{q}{p}\right)^n - 1}$ increases in $n$ and

$$0 \leqslant \sum_{i=1}^{n-1} \frac{1}{pi\theta} \frac{\left(\frac{q}{p}\right)^{n-i} - 1}{\left(\frac{q}{p}\right)^n - 1} \leqslant \sum_{i=1}^{n-1} \frac{\left(\frac{p}{q}\right)^i}{pi\theta} \leqslant \frac{1}{p\theta} \sum_{i=1}^{n-1} \left(\frac{p}{q}\right)^i \leqslant \frac{1}{p\theta} \times \frac{1}{1 - \frac{p}{q}},$$

we know that $\sum_{i=1}^{n-1} \frac{1}{pi\theta} \frac{\left(\frac{q}{p}\right)^{n-i} - 1}{\left(\frac{q}{p}\right)^n - 1}$ has a limit, denoted by $C$.

Using the result in Subsection 3.1 we obtain

$$h(n) = E[L|N_0 = n] = \int_{t>0} \Pr\{L > t|N_0 = n\}dt = \int_{t>0} \Pr\{N_t > 0|N_0 = n\}dt = \int_{t>0} \Pr\{N_t \geqslant 1|N_0 = n\}dt$$

$$\leqslant \int_{t>0} E[N_t|N_0 = n]dt = \int_{t>0} ne^{(p-q)\theta t}dt = \frac{n}{(q-p)\theta}.$$

So letting $n \to \infty$ in Eq. (44) and using the above result we have $h(1) = C < \infty$. By substituting it into (43) we obtain the expression of $h(n)$ in the case of $p < q$ and know $h(N) < \infty$ in this case.

When $p > q$, letting $n \to \infty$ in Eq. (44) we can conclude that $h(1) = \infty$ and then $h(N) = \infty$.

In the case of $p = q$ we have

$$h(n) = \sum_{k=1}^n [h(k) - h(k-1)] = \sum_{k=1}^n \left[h(1) - \sum_{i=1}^{k-1} \frac{1}{pi\theta}\right] = nh(1) - \sum_{i=1}^{n-1} \frac{n-i}{pi\theta}. \tag{45}$$

That is

$$\frac{h(n)}{n} = h(1) - \sum_{i=1}^{n-1} \frac{n-i}{pi\theta n} \tag{46}$$

Let $n \to \infty$ in Eq. (44) we also have $h(1) = \infty$ and then $h(N) = \infty$.

Therefore, we have proven that $h(N) = E[L|N_0 = N] < \infty$ if and only if $p < q$ in the case of $m = 1$. Thus, in view of Proposition 4.4 we know that it holds for $\forall\ m$. □

The above result tells us although $\lim_{t\to\infty} N_t = 0, a.s$, the expected value of the first time that the software has no defects is finite, which seems inconsistent with our intuition. We can give a short explanation of it: although the limit of $N_t$ is zero, it is not bounded. $N_t$ can become extremely large as $t$ goes to infinity with a quite small possibility, and thus the first time that

the software has no defects will get an extremely large value as $t$ goes to infinity with a quite small possibility. Therefore, its expectation may be far away from 0. In fact, the expected value of the first time that the software has no defects is infinite in the case of $p \geqslant q$.

## 5. Conclusions

In this paper, we have developed a Markov usage model to explore the quantitative effects of software testing on software reliability improvement in the presence of imperfect debugging. We have derived several quantities for software reliability measurement and their upper and lower bounds, such as dynamic behaviors of the residual defect number and the observed failure number, distribution of the time between software failures, software reliability and distribution of the first passage time to the specified number of residual defects. These quantities and the relevant upper and lower bounds will be helpful to assess the software reliability of the software under test.

There are several issues deserving further research. One is incorporating the control of the Markov chain. In our model, we assume that the underlying Markov chain is uncontrolled and we cannot effect the evolution of the software testing process. We can relax this assumption and take the control of the Markov chain into account. Another issue worth discussing is considering the estimation of the parameters such as the initial defect number $N$, defect failure rate $\theta_i$, $p$ and $q$. We assume in our model that these parameter are known beforehand, which is quite unrealistic. We should develop statistical inference method to estimate these parameter, which is very meaningful in practice. We can also consider the case that the parameters $p$ and $q$ are time inhomogeneous. The learning curve can account for this. With time elapsing, the tester becomes more and more familiar with the software under test so that $q$ will be increasing in $t$ and $p$ will be decreasing in $t$. It is worth discussing the effect of the learning curve on the testing process.

## Acknowledgments

Ping Cao and Ke Liu were partially supported by the National Natural Science Foundation of China (Grant Nos. 70731003, 70971125) and 973 Project (Grant No. 2010CB731400). Zhao Dong was partially supported by the National Natural Science Foundation of China (Grant Nos. 10721101, 10928103, 11071008) and 973 Project (Grant No. 2011CB808000). Kai-Yuan Cai was partially supported by the National Natural Science Foundation of China (Grant No. 60973006) and the Beijing Natural Science Foundation (Grant No. 4112033). The authors thank the editors and reviewers for the constructive comments and suggestions, which helped to improve this paper.

## References

[1] A. Arcuri, M.Z. Iqbal, L. Briand, Random testing: theoretical results and practical implications, IEEE Transactions on Software Engineering 38 (2012) 258–277.
[2] K.Y. Cai, Z. Dong, K. Liu, C.G. Bai, A mathematical modeling framework for software reliability testing, International Journal of General Systems 36 (2007) 399–463.
[3] K.Y. Cai, Z. Dong, K. Liu, Software testing processes as a linear dynamic system, Information Sciences 178 (2008) 1558–1597.
[4] K.Y. Cai, P. Cao, Z. Dong, K. Liu, Mathematical modeling of software reliability testing with imperfect debugging, Computers and Mathematics with Applications 59 (2010) 3245–3285.
[5] M. Defamie, P. Jacobs, J. Thollembeck, Software reliability: Assumptions, realities and data, in:Proceeding of International Conference on Software Maintenance, September 1999.
[6] S.S. Gokhale, M.R. Lyu, K.S. Trivedi, Analysis of software fault removal policies using a non-homogeneous continuous time Markov chain, Software Quality Journal 12 (2004) 211–230.
[7] J.P. Li, M.L. Li, D.S. Wua, H. Song, An integrated risk measurement and optimization model for trustworthy software process management, Information Sciences 191 (2012) 47–60.
[8] A.P. Mathur, Foundations of Software Testing, Addison-Wesley, 2008.
[9] W. Kremer, Birth and death bug counting, IEEE Transactions on Reliability 32 (1983) 33–47.
[10] S. Ozekici, R. Soyer, Reliability of software with an operational profile, European Journal of Operational Research 149 (2003) 459–474.
[11] D.A. Peled, Software Reliability Methods, Springer, 2001.
[12] S. Poulding, J.A. Clark, Efficient software verification: statistical testing using automated search, IEEE Transactions on Software Engineering 36 (2010) 763–777.
[13] K. Tokuno, S. Yamada, An imperfect debugging model with two types of hazard rates for software reliability measurement and assessment, Mathematical and Computer Modelling 31 (2000) 343–352.
[14] K. Tokuno, S. Yamada, Markovian software reliability measurement with a geometrically decreasing perfect debugging rate, Mathematical and Computer Modelling 38 (2003) 1443–1451.
[15] K. Tokuno, S. Yamada, Relationship between software availability measurement and the number of restorations with imperfect debugging, Computers and Mathematics with Applications 46 (2003) 1155–1163.
[16] J.A. Whittaker, M.G. Thomason, A Markov chain model for statistical software testing, IEEE Transactions on Software Engineering 20 (1994) 812–824.
[17] A. Wood, Software reliability growth models: assumptions vs. reality, in: Proceeding of 8th International Symposium on Software Reliability Engineering, Albuquerque, NM, November 1997, pp. 136–141.