On The Development of Successive Release of Software Using Stochastic Differential Equation –A Theoretical Framework

Kuldeep Chaudhary¹, Prerna Manik² and P.C.Jha³
Department of Operational Research, University of Delhi, Delhi
¹chaudharyiitr33@gmail.com, ²premanamanik@gmail.com and ³jhapc@yahoo.com,

ABSTRACT
Since number of faults detected and removed through each debugging becomes sufficiently small as compared to the initial fault content at the beginning of the testing of large scale software, we use a stochastic process to describe the behavior of fault detection process. However large software systems are not designed in isolation rather a software developer develops to market successive releases of the software. The latest release of software retains some code of the previous release of the software, some proportion of which can also be retained in the subsequent releases in future. New code is also added to the software to further enhance the functioning of the latest release. Therefore the failure phenomenon of successive release of the software shouldn’t be modeled independently for each release rather there is some interdependence in the failure phenomenon of these releases, which should be analyzed. In this paper, we present a theoretical framework for modeling the failure phenomenon of multiple releases of the software using stochastic differential equation.

KEYWORDS
Software Reliability, stochastic differential equation

1. INTRODUCTION
Several SRGMs have been proposed and validated in the literature by many authors under different set of assumptions. Most of the SRGMs describe the failure phenomenon by an exponential (G-O Model [2]) or a S-shaped (Yamada delayed S-shaped [11] model) curve. While some of the SRGMs are flexible, in the sense that they can depict, depending upon parameter values, both exponential and S-shaped models (Ohba [7], Bittanti et al [1], and Kapur and Garg [3], etc.). Similar SRGMs that describe the failure phenomenon with respect to testing effort are developed in the literature [3,7,9,11] describing the fault occurrence and/or removal by exponential or a S-shaped curve or flexible in nature. Models incorporating some realistic factors affecting the testing process [3,4,5,7,10,12,13,17] such as imperfect debugging and fault generation, learning phenomenon of software testing team, categorization of faults, faults of different severity, faults removal as two and three stage process (failure, identification and removal) etc have also been developed for both testing and operational phases.

The SRGMs developed so far in the literature describe the failure/removal phenomenon of isolated software. We see various softwares in the market named as Windows 98, Windows 2000, Windows XP, SPSS 10, SPSS 10.1, SPSS 11 etc. First name denotes the basic software built for some specific application whereas the second name/number denotes the version of the software. In software developing organizations, softwares are not designed in isolation rather there is some interdependence between the development of successive releases of software. A software release can be called as a new version of the software. The interdependence between their developments exists in many ways, which also affects their reliability. In this paper we develop an SRGM which can be applied to describe the failure/removal phenomenon of multiple releases of software in operational phase. Wherein the expected number of failures and consequently the faults removed in operation phase is modeled by the joint effect of failure/removal phenomenon of successive releases.

Many Software Reliability Growth Models (SRGMs) are Non-Homogeneous Poisson Process (NHPP) models. As the size of software system is large and the number of faults detected during the testing phase becomes large, so the change of number of faults that are detected and removed through each debugging becomes sufficiently small as compared to the initial fault content at the beginning of testing phase. In such case, we can model the software fault detection process as a stochastic process with continuous-state space. Some continuous-state space SRGMs based on stochastic differential equations (SDE) of Itô type to assess software reliability for software systems have been proposed so far [6, 14, 16]. Modeling of time evolution of a process with stochastic differential equation is not new and has been successfully used in Financial Engineering, Physic and Life Science to name a few [8].

The paper is organized as follows: In Section 2, we describe the stochastic differential equation based model development for one time release of the software and then we explain model development for successive releases of software using SDE. Finally conclusions are drawn in section 3.

2. STOCHASTIC DIFFERENTIAL EQUATION BASED MODEL DEVELOPMENT

Copy Right © INDIACom-2011 ISSN 0973-7529 ISBN 978-93-80544-00-7
When software is executed in the field and a failure is encountered, it is reported to the developer for correction. At the developers end, the testing and debugging team locate and remove the corresponding fault and return the corrected code to the user. The testing team may also check this fault in all (some) other releases and if the corresponding fault is present in any of the release it is may be removed, however this is done according to the developer’s policies, type of contract with the users and the type of software.

Notations

\( m(t) \) : the number of faults removed in the time \( t \) and is a random variable by time \( t \).

\( a_i \) : Initial number of errors in the \( i^{th} \) release of software.

\( a_{ij} \) : Expected number of faults remaining in \( i^{th} \) release of software at time \( t_{j+1} \).

\( m_i(t-t_j) \) : Expected number of faults removed from \( i^{th} \) release in time interval \((t_j, t_{j+1}] \) due to failures reported by their own users

\( m_i(t) \) : Expected number of faults removed in the time interval \((0,t)\) from \( i^{th} \) release.

\( E(m(t)) \) : Expected number of faults detected in the time interval \((0,t)\) during testing phase.

\( b_i \) : Failure observation/detection, fault removal/correction rate.

\( \gamma(t) \) Standardized Gaussian White Noise.

\( \beta_{ij} \) constant parameter representing a learning phenomenon in the fault removal rate function from \( i^{th} \) release in time interval \((t_j, t_{j+1}] \)

2.1 STOCHASTIC FORMULATION OF SINGLE RELEASE OF SOFTWARE MODEL

The SRGM of an isolated software release with stochastic differential equation can be formulated under the following assumptions.

Assumptions

1. The Non-homogeneous Poisson Process (NHPP) can describe software failure phenomenon. Software Reliability Growth Model is the mean value function of NHPP.

2. Software is subject to failure during execution caused by faults remaining in the software.

3. Fault removal rate is assumed to be non-decreasing inflection S-shaped logistic function to describe the learning effect of the fault removal team.

4. The software fault-detection process is modeled as a stochastic process with a continuous state space.

5. The number of faults remaining in the software system gradually decreases as the testing progresses.

6. During the fault removal, no new fault is introduced into the system and the faults are debugged perfectly.

So, in order to describe the stochastic behavior of the fault detection process, we can use a Stochastic Model with continuous state space. Since the latent faults in the software system are detected and eliminated during the testing phase, the number of faults remaining in the software system gradually decreases as the testing progresses. Therefore, it is reasonable to assume the following differential equation:

\[
\frac{dm(t)}{dt} = r(t)[a - m(t)]
\]

Where \( r(t) \) is a fault detection rate per remaining fault at testing time \( t \). However, the behavior of \( r(t) \) is not completely known since it is subject to random effects such as the testing effort expenditure, skill level of the testers, testing tools and so on and thus might have irregular fluctuations. Thus, we have:

\[ r(t) = b(t) + \text{noise} \]

Let \( \gamma(t) \) be a standard Gaussian white noise and \( \sigma \) a positive constant representing magnitude of the irregular fluctuations. So equation (2) can be written as:

\[ r(t) = b(t) + \sigma \gamma(t) \]

Hence, equation (1) becomes

\[
\frac{dm(t)}{dt} = [b(t) + \sigma \gamma(t)][a - m(t)]
\]

Equation (4) can be extended to the following stochastic differential equation of an Ito type [15]

\[
dm(t) = [b(t) - \frac{1}{2} \sigma^2] [a - m(t)] dt \sigma[a - m(t)] dW(t)
\]

Where \( W(t) \) is a one-dimensional Wiener process which is formally defined as an integration of the white noise \( \gamma(t) \) with respect to time \( t \). Using Ito formula, solution to equation (5) with initial condition \( m(0) = 0 \), is obtained as:

\[
m(t) = a \left[ 1 - \exp \left( \int_0^t \left[ b(x) dx - \sigma W(t) \right] \right) \right]
\]

The Wiener process \( W(t) \), is a Gaussian process and has the following properties:

\[
P_r[W(0) = 0] = 1, E[W(t)] = 0, E[W(t)W(t')] = \min[t, t']
\]

2.2 STOCHASTIC FORMULATION OF SUCCESSIVE RELEASES OF SOFTWARE MODEL

First we discuss the general framework for modeling the reliability growth of successive releases of software assuming that there is no interrelationship between their failure phenomenon. See figure 1 in order to understand the pattern of introduction of \( n^{th} \) releases of the software, where \( t_{i+1} \) is the time of introduction of releases \( i \).
Next we write the failure rate equations and explicit closed form solutions for the various releases of the software released at different time points starting a time $t_0 (=0)$.

Let $\gamma_{ij}(t)$ be a standard Gaussian white noise and $\sigma_{ij}$ a positive constant representing magnitude of the irregular fluctuations for $i^{th}$ release in time period $(t_{j-1}, t_j]$. So $r_{ij}(t)$ can be written as:

$$r_{ij}(t) = b_{ij}(t) + \sigma_{ij} \gamma_{ij}(t)$$

Now, the equation for the rate of fault detection for the $i^{th}$ release, in the time interval $(t_{j-1}, t_j]$ can be written as

$$d\frac{dt}{dt} m_{ij}(t) = (b_{ij}(t) + \sigma_{ij} \gamma_{ij}(t)) [a_{ij} - m_{ij}(t)] \tag{7}$$

**Time period: $t_0 (=0) \leq t \leq t_1$**

The first release of software is released at time $t_0$, failure rate equation for the first release of the software in time interval $t_0 \leq t \leq t_1$ is given by

$$\frac{d}{dt} m_{i1}(t) = \frac{d}{dt} m_{i1}(t - t_0) = \left[ \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} + \sigma_{i1} \gamma_{i1}(t) \right] [a_{i1} - m_{i1}(t - t_0)] \tag{8}$$

The above equation can be written as the following stochastic differential equation of an $\hat{Ito}$ type.

$$dm_{i1}(t) = dm_{i1}(t-t_0) = \left[ \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} - \frac{\sigma_{i1}^2}{2} \right] [a_{i1} - m_{i1}(t - t_0)] + \sigma_{i1} [a_{i1} - m_{i1}(t - t_0)]dW_{i1}(t) \tag{9}$$

Solving equation (9) under the initial condition at $t=0$, $m_{i1}(t-t_0)=0$. Therefore, the transition probability distribution of the above is obtained as follows.

$$m_{i1}(t-t_0) = a_{i1} \left[ 1 - \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} - \frac{\sigma_{i1}^2}{2} \right] e^{b_{i1}(t-t_0)} \tag{10}$$

As we know that the Brownian motion or Weiner Process follows normal distribution, thus the mean number of fault removal is given as

$$E(m_{i1}(t-t_0)) = a_{i1} \left[ 1 - \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} - \frac{\sigma_{i1}^2}{2} \right] e^{b_{i1}(t-t_0)} \tag{11}$$

**Time period: $t_1 < t \leq t_2$**

In this time period, second release of the software is released at time $t_1$. The expected remaining number of faults present in the first release is $m_{i2} = a_{i1} - E(m_{i1}(t_1 - t_0))$. Failure rate equations for two releases corresponding to the failures reported by their own users in time interval $(t_1, t_2] \tag{12}$ are:

$$\frac{d}{dt} m_{i2}(t-t_1) = \left[ \frac{b_{i2}}{1 + \beta_2 e^{-\beta_2 t_{i2}}} + \sigma_{i2} \gamma_{i2}(t) \right] [a_{i2} - m_{i2}(t - t_1)] \tag{13}$$

$$\frac{d}{dt} m_{i2}(t-t_1) = \left[ \frac{b_{i2}}{1 + \beta_2 e^{-\beta_2 t_{i2}}} + \sigma_{i2} \gamma_{i2}(t) \right] [a_{i2} - m_{i2}(t - t_1)] \tag{14}$$

Using equation (13) and (14) expected number of faults removed for release one and two by time $t_2$ are given by

$$E(m_{i1}(t-t_1)) = a_{i1} \left[ 1 - \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} - \frac{\sigma_{i1}^2}{2} \right] e^{b_{i1}(t-t_1)} + \sigma_{i1}^2 \frac{t_1}{2} \tag{15}$$

$$E(m_{i2}(t-t_1)) = a_{i2} \left[ 1 - \frac{b_{i2}}{1 + \beta_2 e^{-\beta_2 t_{i2}}} - \frac{\sigma_{i2}^2}{2} \right] e^{b_{i2}(t-t_1)} + \sigma_{i2}^2 \frac{t_1}{2} \tag{16}$$

Using equation (10),(15)&(16) expected value of total number of faults removed from release one and two any time, assuming no interdependence between their phenomenon, after the time $t_1$ are given by

$$E(m(t)) = E(m(t_1 - t_0)) + E(m(t-t_1))$$

$$E(m(t)) = a_{i1} \left[ 1 - \frac{b_{i1}}{1 + \beta_1 e^{-\beta_1 t_{i1}}} - \frac{\sigma_{i1}^2}{2} \right] e^{b_{i1}(t-t_1)} + \sigma_{i1}^2 \frac{t_1}{2} \tag{17}$$

$$E(m(t)) = a_{i2} \left[ 1 - \frac{b_{i2}}{1 + \beta_2 e^{-\beta_2 t_{i2}}} - \frac{\sigma_{i2}^2}{2} \right] e^{b_{i2}(t-t_1)} + \sigma_{i2}^2 \frac{t_1}{2} \tag{18}$$

Copy Right © INDIACom-2011 ISSN 0973-7529 ISBN 978-93-80544-00-7
Similarly we can generalize the equation for expected number of failures reported by its own users in time interval (t_{n-1}, t_n] as:

$$E(m_i(t-t_{n-1})) = a_m \left( 1 + \frac{\beta_i \exp^{\frac{d}{a_i}}}{1 + \beta_i \exp^{\frac{d}{a_i}}} \right) \exp \left( \frac{a_i(t-t_{n-1})}{2} + \frac{\sigma_i^2}{2} \right)$$

Equation for expected cumulative number of fault removal for i^{th} release of software by time t is given by:

$$E(m_i(t)) = \sum_{j=i}^{n} a_m \left( 1 + \frac{\beta_j \exp^{\frac{d}{a_j}}}{1 + \beta_j \exp^{\frac{d}{a_j}}} \right) \exp \left( -b_j(t-t_{j-1}) + \frac{\sigma_j^2}{2} t_{j-1} \right) + a_m \left( 1 + \frac{\beta_j \exp^{\frac{d}{a_j}}}{1 + \beta_j \exp^{\frac{d}{a_j}}} \right) \exp \left( -b_i(t-t_{n-1}) + \frac{\sigma_i^2}{2} t_{n-1} \right)$$

CONCLUSION
In this paper we have proposed an SRGM for the successive releases of the software using stochastic differential equation. The scope of this paper is restricted to only the theoretical framework of the model.

FUTURE SCOPE
In this model we have assumed no interaction between the sales of the successive releases of the product type software, however latest releases may have substitution effect on the sale of earlier releases. We are working on a stochastic differential equation marketing model that can describe the substitution effect on sales of the successive releases. In this paper, we assume that adoption process of newer successive technology evolves according to a general stochastic process, which can be represented by a stochastic differential equation. The stochastic process can best be thought of as capturing market uncertainty generated by stochastic fluctuations in price or demand at the product market stage. These are some interesting works to be done in future.

REFERENCES