A New Insight into Software Reliability Growth Modeling

P.K. KAPUR¹, ANU G. AGGARWAL¹, and SAMEER ANAND²

¹Department of Operational Research, University of Delhi, Delhi-110007.
²S.S. College of Business Studies, University of Delhi, Delhi-110095.

(Received on October 24, 2008; Final Revision on December 9, 2008)

Abstract: Several software reliability growth models have been presented in the literature in the last three decades. They have been developed for uniform and non-uniform operational profile. Some of them are flexible whereas others are not. Model selection becomes an uphill task. Of late, some authors have tried to develop a unifying approach so as to capture different growth curves, thus easing the model selection process. Some of these approaches use (a) Random lag function (b) Infinite server queueing theory (c) Hazard rate function. The purpose of this paper is to show that all these approaches are equivalent and further show that hazard rate approach is more general and can handle both Imperfect Debugging and Fault generation. This paper thus provides a new insight into the model development and it is shown that how a wide variety of existing software reliability can be unified.

Keywords: Fault detection, fault correction, infinite server queue, hazard rate, imperfect debugging, fault generation

1. Introduction

With increased complexity of products design, shortened development cycles and highly destructive consequences of software failures, a major responsibility lies in the areas of software debugging, testing and verification. Testing is defined as the execution of a program to find the faults, which might have been introduced in it during various stages of the development cycle. It is also performed to judge the performance, safety, fault-tolerance or security of the software. More importantly, testing provides a mathematical measure of software reliability (i.e., failure/execution time) which forms a vital input to the release decision.

A large number of Software Reliability Growth Models (SRGM), which relate the number of failures (faults identified/corrected) and execution time, have been discussed in the literature [6,8]. These SRGM assume diverse testing environment like distinction between failure and correction processes, learning of the testing personnel, possibility of imperfect debugging and fault generation, constant or monotonically increasing / decreasing Fault Detection Rate (FDR) or randomness in the growth curve. But no SRGM can be claimed to be the best as the physical interpretation of the testing and debugging changes due to numerous factors e.g., design of test cases, defect density, skills and efficiency of testing team, availability of testing resources etc. The plethora of SRGM makes the model selection a tedious task. To reduce this difficulty, unified modeling approaches have been proposed by many researchers. These schemes have proved to be successful in obtaining several existing SRGM by following single methodology and
thus provide an insightful investigation for the study of general models without making many assumptions. Some of these approaches are based on (a) Infinite server queuing theory (b) Random lag function (c) Hazard rate function.

The work in this area started as early as in 1980s with Shantikumar [10] proposing a Generalized birth process model. Gokhale and Trivedi [2] used Testing coverage function to present a unified framework and showed how NHPP based models can be represented by probability distribution functions of fault –detection times. Dohi et al [1] proposed a unification method for NHPP models describing test input and program path searching times stochastically by an infinite server queuing theory. Inoue [3] applied infinite server queuing theory to the basic assumptions of delayed S-shaped SRGM [13] i.e. fault correction phenomenon consists of successive failure observation and detection/correction processes and obtained several NHPP models describing fault correction as a two stage process.

Another unification methodology is based on a systematic study of Fault detection process (FDP) and Fault correction process (FCP) where FCPs are described by detection process with time delay. The idea of modeling FCP as a separate process following the FDP was first used by Schneidewind [9]. More general treatment of this concept is due to Xie et al [11] who suggested modeling of Fault detection process as a NHPP based SRGM followed by Fault correction process as a delayed detection process with random time lag. The recent unification scheme (due to Kapur et al [4]) is based on Cumulative Distribution Function for the detection/correction times. They have extended the concept of unified modeling by incorporating the concept of change point in Fault detection rate [5]. In this paper, we discuss above-mentioned three approaches in detail and show that how these unifying tools, though derived under different sets of assumptions are mathematically equivalent. The paper also highlights the importance of hazard rate function based unifying technique.

This paper has been organized as follows: Section 2 discusses three types of unification schemes reported in the literature. This section has been divided into three subsections. Subsection 2.1 describes unification approach for modeling software reliability Growth using infinite server queuing theory (Inoue [3]). In subsection 2.2 we describe the unification approach based on random time lag functions (Xie et al [11]). We include Hazard rate function based unifying technique in subsection 2.3 (Kapur et al [4]). Section 3 proves the equivalence of these unifying models. Section 4 highlights the importance of Hazard rate function scheme with respect to its capability to incorporate Imperfect Debugging and Fault generation. Finally, the paper concludes with a brief summary and directions for future research in Section 5.

Notation

\[ m_d(t), m_c(t) \] Mean value function (MVF) or the expected number of faults detected and corrected by time \( t \).

\( a \) Constant, representing the initial number of faults lying dormant in the software when the testing starts.

\[ \lambda_d(t), \lambda_c(t) \] Intensity function for FDP and FCP or Fault Detection and Correction rate per unit time.

\[ F_d(t), F_c(t) \] Distribution Function for Fault Detection and Correction Times

\( f_c(t) \) Probability Density Function for Fault Correction Time

\( * \) Convolution.

\( \circ \) Steiltjes convolution.
2. Unification Approaches for Modeling Software Reliability Growth

2.1 Unification Approach for Modeling Software Reliability Growth Using Infinite Server Queuing Theory (Inoue [3])

The model is based on the following assumptions

1. Software system is subject to failure during execution caused by faults remaining in the system.
2. The number of faults detected at any time instant is proportional to the remaining number of faults in the software. Further, All faults are mutually independent from failure detection point of view.
3. On a failure, correction effort starts and fault causing the failure is corrected with certainty. Number of corrected faults lags behind the total number of detected faults, hence there is a time lag between the Fault correction and detection processes.
4. The fault detection process is modeled by NHPP. The fault correction times are assumed to be independent with probability distribution $F_c(t)$.

The representation of Software Fault correction process as a infinite server queuing model is depicted in Fig. 1.

![Fig 1: Software Fault Correction Process](image)

Let the counting processes $\{X(t), t \geq 0\}, \{N(t), t \geq 0\}$ represent the cumulative number of software fault detected, faults corrected respectively up to time $t$ and the test begun at time $t=0$. Then the distribution of $N(t)$ is given by

$$\Pr\{N(t) = n\} = \sum_{j=0}^{\infty} \Pr\{N(t) = n \mid X(t) = j\} \frac{(m_d(t))^j e^{-m_d(t)}}{j!}$$

(1)

If failure observations count is $j$ then probability that $n$ faults are corrected via the fault correction process is given as

$$\Pr\{N(t) = n \mid X(t) = j\} = \binom{j}{n} (p(t))^n (1 - p(t))^{j-n}$$

(2)

where $p(t)$ is the probability that an arbitrary fault is corrected by time $t$, which can be defined using the Stieltjes convolution and the concept of the conditional distribution of arrival times, given as

$$p(t) = \int_0^t F_c(t-u) \frac{dm_d(u)}{m_d(t)}$$

(3)
The distribution function of cumulative number of faults corrected up to time $t$ using equations (2) and (3) is given as

$$\Pr\{ N(t) = n \} = \left( \int_0^t F_c(t-u) dm_d(u) \right)^n \frac{e^{-\int_0^t F_c(t-u) dm_d(u)}}{n!}$$

$$ \tag{4}$$

Equation (4) describes that $N(t)$ follows an NHPP with MVF $\int_0^t F_c(t-u) dm_d(u)$ i.e.,

$$m_c(t) = \int_0^t F_c(t-u) dm_d(u) \tag{5}$$

Hence knowing the MVF for detection times $m_d(t)$ and distribution of correction times $F_c(.)$ we can compute the MVF $m_c(t)$ of a two stage fault isolation/detection and correction process for the various existing SRGM.

2.2 Unification Approach For Modeling of Software Fault Detection and Fault Correction Process (Xie et al [11])

This approach is based upon separate analysis of fault detection process (FDP) and fault correction process (FCP). Given $\lambda_d(t)$, the mean value function (MVF) $m_d(t)$ satisfies

$$m_d(t) = \int_0^t \lambda_d(x) dx \tag{6}$$

Specifically, a fault can be corrected only after its detection, and a FCP can be modeled as a delayed FDP. Such delay could be modeled as deterministic or random, but the deterministic assumptions on correction time are quite simplistic. In fact, it is more practical to model the correction time with random variables. Given the fault detection intensity function $\lambda_d(t)$, the fault correction intensity function is the expectation of $\lambda_c(t-x)$ i.e., $\lambda_c(t) = E \left[ \lambda_c(t-x) \right]$

or $\lambda_c(t) = \int_0^\infty \lambda_c(t-x) f_c(x) dx \tag{7}$

Then FCP can be described by the following MVF

$$m_c(t) = \int_0^t \lambda_c(u) du \tag{8}$$

2.3 A Unified Approach For Developing Software Reliability Growth Models Using Hazard Rate (Kapur et al [4])

Most of the SRGM reported in the last few decades assume detection process is followed by immediate fault correction. While in reality, each detected faults is reported, diagnosed, corrected and then verified. Therefore, the time from detection to correction should not be neglected in software testing process. When the correction in done in two stages, then the fault correction process is given by the following differential equation:

$$\frac{dm_c(t)}{dt} = \frac{(f_c * f_d)(t)}{[1-(F_c * F_d)(t)]} [a - m_c(t)] \tag{9}$$
where \( \frac{(f_c * f_d)(t)}{[1 - (F_c \otimes F_d)(t)]} \) is fault detection - correction rate per fault or the hazard rate function. Solving the above, we have
\[
m_c(t) = a(F_c \otimes F_d)(t)
\] (10)

It is an extension of one stage failure-detection/correction process given in Musa [7].

3. **Equivalence of Three Unification Approaches**

In first step we show the equivalence of scheme due to Xie et al [11] with Infinite server approach due to Inoue [3].

Considering \( m_c(t) = \int_0^t \lambda_c(y) dy \) (equation 8)

From (7) and (8) we have
\[
\lambda_c(y) = \int_0^y \lambda_d(y-x)f_c(x)dx
\]

So,
\[
m_c(t) = \int_0^t \int_0^y \lambda_d(y-x)f_c(x)dx dy = \int_0^t \int_0^t \lambda_d(y-x)dy f_c(x)dx
\]

or
\[
m_c(t) = \int_0^t m_d(t-x)f_c(x)dx
\]

\[
= \int_0^t F_c(t-x)dm_d(x)
\]

which is the mean value function of infinite server model as given in equation (5)

The next step establishes the equivalence of infinite server queuing model to unification scheme based on hazard rate (Kapur et al [4])

Considering \( m_c(t) = \int_0^t m_d(t-x)dF_c(x) \) (equation 5)

\[
= \int_0^t F_c(t-x)dm_d(x) = F_c(t) \otimes m_d(t)
\]

Using \( m_d(t) = aF_d(t) \) [Musa, 7], we get
\[
m_c(t) = a(F_c \otimes F_d)(t)
\]

which is the mean value function of hazard rate approach as given in equation (10)

Hence we show that the above-described three approaches are equivalent.

4. **Advantages of Hazard Rate Function Based Unification Approach**

This section highlights the advantages of the unifying technique with hazard rate function. First, let us observe the unifying models as given by infinite server queuing theory and
Xie et al [11] These two schemes express MVF for Fault correction process in terms of MVF for Fault detection process and probability distribution function for fault correction times. These two schemes fall short of generalization when we wish to incorporate the possibility of imperfect debugging and/or fault generation. It should be noted that imperfect debugging is possible when attempts are made to correct the cause of the failure. During the correction process we can have imperfect debugging under following three cases:

(i) The fault is wrongly isolated followed by inaccurate correction,
(ii) The underlying fault is partially removed,
(iii) Few additional faults are introduced while correcting the underlying fault.

All these cases are possible only when efforts are made to remove the fault. This concept cannot be incorporated in Fault detection process. So \( m_d(t) \) cannot be taken as imperfect debugging NHPP model.

We consider a unified model based on hazard rate with Imperfect Debugging and Fault Generation, which is given by the following differential equation:

\[
\frac{dm(t)}{dt} = \frac{(f_c * f_d)(t)}{[1 - (F_c \otimes F_d)(t)]} p(a + \alpha n(t) - m(t))
\]  

(11)

where \( p \) is the probability of perfect debugging and \( \alpha \) is the rate at which faults are introduced while removing/correcting a fault from software.

Solving the above differential equation, we get the solution as:

\[
m(t) = \frac{a}{(1 - \alpha)} [1 - (F_c \otimes F_d)(t))]^{\rho(1-\alpha)}
\]

(12)

Here, regardless of the probability distribution followed by the detection and correction times, we have MVF for fault correction process with both imperfect debugging and fault generation. Above has been termed as Generalized Non-homogeneous Poisson Process (GINHPP) SRGM in the presence of Imperfect Debugging and Fault Generation. (Kapur et al [4]). For \( \rho=1 \) and \( \alpha=0 \), we obtain model for perfect debugging and no fault Generation as given in equation (10).

5. Conclusions

In this paper, we have discussed three different approaches to unify a wide range of software reliability growth models under a common modeling framework. Though these three schemes have been derived under different sets of assumptions but they are proved to be mathematically equivalent. Further, we have mentioned certain advantages that come with hazard rate based unifying methodology. This scheme provides an integrated common platform for not only growth models with perfect debugging but also for imperfect debugging and fault generation. Here, we have considered the unification of continuous time NHPP based models but research can be done to work out the unification platform for the discrete time models. The Testing efforts function based models can also be integrated with one of the approaches described in this paper. So far, we have restricted ourselves to two-stage fault detection followed by fault correction process. The study can be extended to situations where fault correction takes place in three stages—failure observation, fault isolation/detection and finally fault correction. We are also
working on the unification of stochastic calculus based models. The unification framework of modeling seems to be quite interesting and promising for easing the problem of model selection.

References


**P. K. Kapur** is Professor and former Head in the Department of Operational Research, University of Delhi. He has edited special issues of International Journal of Reliability, Quality and Safety Engineering (2004, 2007), OPSEARCH, India (2005), International Journal of Performability Engineering (IJPE India, 2006) and a special issue of Communications on Dependability and Quality Management, Belgrade, Serbia. He has organized three International Conferences in the year 2000, 2003 and 2006 on Quality Reliability and Information Technology and has published more than 150 research papers in the areas of hardware reliability, optimization, queuing theory, maintenance and software reliability. He has edited four volumes on Quality, Reliability and IT published in the year 2004, 2005, 2007 and 2008 and co-authored a book "Contributions to Hardware and Software Reliability" published from World Scientific, Singapore. He has guided M.Tech. / Ph.D. theses in Computer Science as well as in Operations Research. He
has traveled extensively in India and abroad and delivered invited talks. His research interests include Hardware and Software Reliability, Optimization, Queuing Theory, Innovation Diffusion Modeling, Numerical Computation of Stochastic models and Marketing. He is former President of Operational Research Society of India. Currently he is president of Society for reliability Engineering, Quality and Operations Management (SREQOM).

Anu G. Aggarwal received her Ph.D. degree in Software Reliability from University of Delhi, Delhi. She is presently on the faculty in the Department of Operational Research, University of Delhi. She has published several papers in the area of software reliability. Her current research interests include soft computing. She is a member of Society for Reliability Engineering, Quality and Operations Management (SREQOM).

Sameer Anand received his M.Sc. and M.Phil. degree in Operational Research from University of Delhi, Delhi. He is presently lecturer in the Department of Computer Science at S.S. College of Business Studies, University of Delhi. He has published several papers in the area of software reliability and Marketing. He is a member of Society for Reliability Engineering, Quality and Operations Management (SREQOM).