Computers and Mathematics with Applications 60 (2010) 2015–2030

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Software reliability analysis and assessment using queueing models with multiple change-points

Chin-Yu Huang [a,*], Tsui-Ying Hung [b]

[a] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

[b] Customer Service Systems Laboratory, Chungwha Telecom Co., Ltd., Taipei, Taiwan

ARTICLE INFO

Article history:
Received 31 May 2009
Received in revised form 6 July 2010
Accepted 23 July 2010

Keywords:
Software reliability growth model (SRGM)
Change-point
Software testing
Debugging
Non-homogeneous Poisson process (NHPP)

ABSTRACT

Over the past three decades, many software reliability growth models (SRGMs) have been proposed, and they can be used to predict and estimate software reliability. One common assumption of these conventional SRGMs is to assume that detected faults will be removed immediately. In reality, this assumption may not be reasonable and may not always occur. During debugging, developers need time to reproduce the failure, identify the root causes of faults, fix them, and then re-run the software. From some experiments or observations, the fault correction rate may not be a constant and could be changed at certain points as time proceeds. Consequently, in this paper, we will investigate and study how to apply queueing models to describe the fault detection and correction processes during software development. We propose an extended infinite server queueing model with multiple change-points to predict and assess software reliability. Experimental results based on real failure data show that the proposed model can depict the change of fault correction rates and predict the behavior of software development more accurately than traditional SRGMs.

1. Introduction

Software is a relatively new product (and design approach) compared to hardware-based systems. Most software engineers have always spent a lot of time and effort trying to figure out how to create reliable software with minimum errors. According to ANSI's definition, software reliability is the probability of failure-free software operation for a specified period of time in a specified environment [1,2]. Software reliability mainly depends on the code that results from several intermediate processes that must be considered in the final analysis. Time and money can be saved when early reliability prediction is feasible and measurable. Over the past three decades, many non-homogeneous Poisson process (NHPP)-based *software reliability growth models* (SRGMs) have been proposed for estimating and assessing the reliability growth of products [3,4].

In general, SRGMs allow engineers to determine how software reliability varies with time during testing and help them decide when to stop testing [5]. SRGMs have been widely used in many practical applications [6–8]. The applications of SRGMs include important and safety-critical systems such as the shuttle's on-board system software [9] and weapon systems [10]. In addition to ANSI/AIAA, SRGMs have also been adopted or recommended by a number of leading companies or research institutions, such as AT&T, North Telecom, Bell Canada, JPL and AFOTEC (the Air Force Operational Evaluation Center) [11–13]. Baumer et al. [14] used NHPP models to predict fault inflow in some large-scale software projects of Ericsson AB. According to their experiences, the concave NHPP models performed well in short-term predictions in iterative processes. NHPP models are still adopted in industry.

* Corresponding author. Tel.: +886 3 5742972; fax: +886 3 5723694.
E-mail address: cyhuang@cs.nthu.edu.tw (C.-Y. Huang).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.07.039

Software reliability estimation is generally affected by three main factors: fault detection, fault removal, and operational profile. Fault(s) can be detected and moved when a program fails and a failure occurs during execution. One common assumption of most conventional SRGMs is that detected faults are immediately removed [15]. In reality, this assumption may not be reasonable and may not always occur. There is often a considerable time delay between finding a fault and fixing it, once the correction process has started. In the past decade, some researchers have discussed how to use queueing-based approaches to explain the debugging behavior and to allocate limited testing resources [16–20]. In the queueing-based approaches of modeling software reliability, detected faults and debuggers are usually regarded as customers and servers, respectively. The time between a customer entering the queue and leaving after service completion is considered to be the time needed to remove a bug.

In addition, most SRGMs typically assume that the rate of fault correction is a constant. In reality, this rate strongly depends on the skill of the debuggers, the difficulty of the problems (errors), debugging environments and tools, etc. Thus the fault correction rate may be neither constant nor smooth. Instead, it could be changed at certain moments in time called *change-points* (CPs) [21,22]. In general, a CP is the time instant when a model's parameter experiences a discontinuity in time. Zhao [23] suggested that, before the software is delivered to users, a CP may occur when the testing strategy and resource allocation are changed. That is, the running environment may change at that moment. The availability of testing facilities and other random factors can be the causes of the CPs. Some reliability CP models have been published in the past, such as the Weibull change-point model, the Jelinski–Moranda de-eutrophication model with a change-point and the Littlewood model with one change-point [23].

In this paper, we will propose an extended infinite server queueing model (EISQM) with multiple CPs. The proposed model can help managers and developers measure software reliability, and can understand the possible change of the fault correction rate. We also propose a method to locate the CP(s). That is, this method can greatly help engineers to estimate and provide inferences for the location of CP(s). The applicability of our proposed model will be demonstrated using two sets of real software failure data. Experimental results show that the proposed models give a better fit to the data sets and predict the future behavior well.

The remainder of this paper is organized as follows. Section 2 gives a review of some existing SRGMs. We further survey the application of queueing theory in software reliability modeling and discuss the problem of time lag between failure detection and fault correction processes in Section 3. In Section 4, we will show how to derive an EISQM with multiple CPs. We estimate the parameters of the proposed models based on real data sets, and compare the performance of the proposed models with a number of SRGMs in Section 5. Finally, conclusions are given in Section 6.

2. Review of some software reliability models

In the past three decades, many SRGMs have been proposed and studied [2,24]. For example, Musa et al. [1] focused on the study of the NHPP and further proposed the logarithmic Poisson execution time model. Another important model, the Schneidewind model, proposed by Schneidewind, is also a type of NHPP model [2,7,9]. The most important feature is that the applications of NHPP models are diverse and critical in practice. Both the logarithmic Poisson execution time model and the Schneidewind model are recommended by ANSI/AIAA. Among various models, NHPP models are very attractive. Xie [25] reported that NHPP models are simple and easy to use, and they are the type of software reliability model that is most widely applied. Besides, Farr and Lyu [2] also pointed out that the NHPP model has formed the basis for models using the observed number of faults per unit time group. But it should be noticed that there are no universally applicable SRGMs so far, i.e., no existing models can be trusted to provide accurate predictions of reliability in all situations. In the following, some widely used NHPP SRGMs are presented and discussed.

2.1. The Goel–Okumoto model

This model, first proposed by Goel and Okumoto [2,24], is one of the most popular and famous NHPP-based models. The assumptions of the Goel and Okumoto model are as follows.

1. The fault removal process follows the NHPP.
2. The software system is subject to failures at random times caused by faults remaining in the system.
3. The mean number of failures detected in the time interval $(t, t + \Delta t]$ is proportional to the mean number of faults remaining in the system.
4. All faults are mutually independent and have the same chance of being detected.
5. When a failure is observed and a fault is detected, the detected fault will be removed immediately, and no new faults are introduced.

Based on the above assumptions, we have the following differential equation:

$$\frac{\mathrm{d}m_d(t)}{\mathrm{d}t} = b\left[a - m_d(t)\right], \tag{1}$$

where $m_d(t)$ is the expected number of faults detected by time t, a is the expected number of faults which will be eventually detected, and b is the fault detection rate. Solving Eq. (1) under the boundary condition $m_d(0) = 0$, we have

$$m_d(t) = a(1 - \mathrm{e}^{-bt}), \quad a > 0,\ b > 0. \tag{2}$$