Mathematical and Computer Modelling 54 (2011) 3046–3064

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

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

Analyzing the effect of imperfect debugging on software fault detection and correction processes via a simulation framework

Chu-Ti Lin *

Department of Computer Science and Information Engineering, National Chiayi University, Chiayi, Taiwan

ARTICLE INFO

Article history:
Received 12 October 2010
Received in revised form 26 July 2011
Accepted 26 July 2011

Keywords:
Software testing
Software debugging
Software reliability
Simulation-based approach
Imperfect debugging
Non-homogeneous continuous time Markov chain

ABSTRACT

During a debugging operation, there is a high probability that an additional fault will be introduced into the program when removing an existing fault. Thus, perfect debugging is an ideal but impractical assumption when modeling software reliability. If the debugging of a software system is imperfect, more faults may be introduced and detected. In such cases, it may be necessary to add more staff to the debugging team to share the load and ensure the quality of the software. To investigate the effects of imperfect debugging, we simulate the fault detection and correction processes by a single-queue multichannel queuing model with feedback. In this paper, two debugging procedures are discussed. The first, called PROCEDURE_PERFECT_DEBUGGING, is based on a single-queue multichannel queuing model and deals with the case of perfect debugging. Then, we relax the restriction on perfect debugging, and further propose PROCEDURE_IMPERFECT_ DEBUGGING based on a queuing model with feedback to address the case of imperfect debugging. We demonstrate the implementation of the procedures via two case studies in which we quantify the effects of imperfect debugging in terms of throughput, time consumption, and debugger utilization. Finally, based on the measurement results, we determine the most suitable staffing level (i.e., the number of debuggers required) for a debugging system under different degrees of imperfect debugging.

1. Introduction

Software testing and debugging are essential operations in the development of reliable software systems. During the testing and debugging phase of a system, some records about the execution of the underlying software may be collected, e.g., resource consumption, raw data about faults and failures, and the time required to fix bugs [1]. Scientific analysis of those records could provide insight into the status and characteristics of the software system. Such records constitute a valuable source of information that can help project managers guide the execution of software projects. For example, the information enables managers to assess the progress of the testing phase, determine whether the allocated testing resources are sufficient, analyze the failure process, and decide the optimal time to stop testing and release the system. Among the collected data, the records of software failures are particularly important for ensuring the quality of the underlying software systems.

The software reliability growth model (SRGM), a type of parametric technique, is one of the most effective approaches for predicting the software failure process [2–5]. To date, more than 100 models have been proposed [6]. However, Tausworthe and Lyu [2] observed that the underlying assumptions of most SRGMs may oversimplify the failure process. Some non-parametric techniques have also been proposed for predicting software failure processes, e.g., the artificial neural network

* Tel.: +886 5 2717227; fax: +886 5 2717741.
E-mail address: chutilin@mail.ncyu.edu.tw.

0895-7177/$ – see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2011.07.033