Chapter 13
Fault Prediction Modelling in Open Source Software Under Imperfect Debugging and Change-Point

Shozab Khurshid
University of Kashmir, Srinagar, India

A. K. Shrivastava
Fortune Institute of International Business, New Delhi, India

Javaid Iqbal
University of Kashmir, Srinagar, India

ABSTRACT

Instant demand of products and services by technologically active users has increased the demand for open source software (OSS)-based applications. Unfortunately, with the complexity and lack of understanding of OSS-based systems, it becomes difficult for a testing team to remove the faults and the fault removal rate becomes low in comparison to what it should be. This also results in generating new faults during removal. Also, the rate at which the testing team detects/corrects fault need not be same during the entire process of testing due to various reasons viz. change in testing strategy, understanding of code, change in resources, etc. In the existing literature on OSS, authors have developed many models considering the above aspects separately. In this article, all of the above aspects have been combined to develop a general framework for predicting the number of faults in OSS. The comparison of eight models on the basis of their prediction capability on two well-known Open Source Software datasets is created and then ranked using normalized criteria distance approach.

DOI: 10.4018/978-1-7998-3016-0.ch013

1. INTRODUCTION

Software is considered as an essential component as its need is felt in every walk of life. In modern era, there is a rapid expansion in the use of software and thus the software engineers strive hard in producing error free and quality software for customer satisfaction. Software development can be done in one of the two ways i.e. either in-house (Closed Source Software) or Open Source Software (OSS). OSS differs significantly from the closed source software in terms of its designing and development. Open Source Software can be defined as the software in which user can approach the source code and modify it over the internet. This helps different users sitting at different locations to interact and develop the software at any time. This also enhances the speed of software development according to the need. On the other hand, the source code of CSS is inaccessible to its users, with the result, the users can run the software but lack the opportunity of modifying the code (Najeeb ullah et al., 2012). The increasing demands of tech savvy users in no time has forced IT firms and software developers to collaborate so that requirements can be fulfilled on time. The reason that OSS is satisfying the desires and interests of users incisively is that it follows the methodology of parallel development and debugging.

The concept of OSS in which sharing of ideas and source code was believed to be the fundamental principle was given by Richard Stallman, a software developer from America in the 1970's. The key characteristics of OSS are that the source code must be included in the software distribution of Open Source Software and the user must be able to reproduce or modify the software. The step wise release cycle of open source is as:

1. Open Source Software is released on Internet
2. Users implement the OSS
3. Users report the bugs in software
4. These reports are sent for verification and validation
5. Accordingly, the source code is modified and same is uploaded by the developer
6. Repeat Step 1

Since OSS is developed by massive volunteers throughout the world, there are some issues with it as well. No documentation is present as there is no contractual responsibility for OSS, poor user-interface design that makes it less significant to be used, and lastly, thousands of contributors participate in OSS testing and put forth their bug reports which can enhance the software reliability many times but still there are some loopholes in its security.

As OSS is free with respect to the licensing, and is being updated constantly by various developers, measuring its quality in terms of reliability becomes a prime requisite. The failure pattern of OSS is different from that of CSS and as such their reliability grows in a slightly different manner. The reason behind this is the frequent change in the OSS source code. In the current study, we will study the reliability growth of Open Source Software projects and propose a unified framework of software reliability growth model catering different needs with respect to the diverse testing and debugging environment (T&D).

Notations

$\Lambda(t)$: Cumulative number of faults detected or removed by time t

$\omega(t)$: Time dependent initial fault content of software.

I: Probability of perfect debugging.