Generalized Multi-Release Framework for Fault Prediction in Open Source Software

Shozab Khurshid, University of Kashmir, Srinagar, India
A.K. Shrivastava, International Management Institute, Kolkata, West Bengal, India
Javaid Iqbal, University of Kashmir, Srinagar, India

ABSTRACT
Software developing communities are shifting to open source software (OSS) because of the reason that software development takes place in successive releases, thereby improving its quality and reliability. Multi-release development of OSS can provide an opportunity to inculcate the dynamic needs of the user in a very short span of time to survive in the market. In spite of having these benefits, numerous challenges can be faced during the multi-release OSS development. Some of the challenges can be the generation of errors during the addition of new features. To address the changing fault detection process, a change point phenomenon is considered so as to give more practicality to the model. In this article, we present a general framework for multi-release OSS modelling incorporating imperfect debugging and change points. Parameter estimation and model validation is done on the three releases of Apache, an open source software project.

KEYWORDS
Change Point, Error Generation, Imperfect Debugging, Multi Release, Non-Homogenous Poisson Process (NHPP), Open Source Software (OSS), Ranking, Software Reliability Growth Model (SRGM)

INTRODUCTION
In recent years, a rapid shift from closed source software to open source has been noticed. Reasons being availability of the source code, cutting of cost, development of skill, support from outside which in turn leads to more mature and reliable software product. A large number of users participate in testing the code and design of the OSS during its development process. Initially, OSS is developed by an individual or a small group of software developers for their own purpose. As OSS gains popularity, its volunteers increase tremendously and get involved in modifying the code and reporting the bugs in the software. The communities that are involved in the OSS development are moving towards multi up-gradation of the software so as to release the software early in the market, to meet the changing demands of the user and to correct the bugs reported by the users. Although this approach promotes feature addition of the software but there can be an increase in the fault content with the up-gradations. Thus, multi-release development approach should be embraced only if it provides better reliability.

DOI: 10.4018/IJSI.2019100105

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
Since multi up-gradation strategy provides benefits to the users as well as the software firms but it is considered as the most challenging task. When the software is being updated, an increase in the failure rate is being observed. This is because of the increase in the fault content due to the addition of the new functionality, the faults left in the previous release and also the bug reports from the users. Nevertheless, during the debugging process the failure rate tends to decrease gradually. A number of multi-version OSS projects are available namely Apache, Genome, Mozilla Firefox and R.

LITERATURE REVIEW

In the field of information technology, reliable software has become the most important concern. Software reliability is defined as “the probability of error free operation of the software for a given time period within some specified environment” (Kapur et al., 2011; Pham, 2006). Software Reliability Growth Models are used to quantitatively estimate the reliability of the software during the testing phase. A mathematical relationship between the testing time and total number of faults is given by SRGMs. Although a number of reliability models have been proposed so far, but all these are based on some limitations (Musa, 2004; Kapur et al., 2011).

Tremendous work has been done to assess the reliability and also to find a suitable model for Open Source Software. The phases that are involved in the development process of OSS was given by (Jorgensen, 2001). A comparative study between open and closed software was done by (Ven et al., 2008). In their study, comparison between the two was carried with respect to the development, cost, number of users and software design. Due to the limitations in the time and the resources, it becomes impossible for the developers to deliver a software product meeting the desired level of reliability within a single Software Development Life Cycle. Successive release is planned by the software developers which provides a number of benefits over a single release policy. The benefits that are provided by successive release policy has been studied by (Garmabaki et al., 2012). Kapur et al. (2014) proposed a multi up-gradation model for closed software that considers the issues related to the generation of faults when new functionality is added in the software. The SRGM proposed by them has the capability to capture the faults that are generated due to the add-ons along with the faults from the previous release that have been left undetected.

Although a lot of research pertaining to multi-release modelling has been put forth by a number of researchers but only a little work has been done considering multi-release modelling for Open Source Software. The research on the optimal time for the up-gradation of the version based on the maintenance effort of the software was done by Tamura and Yamada (2007). Li et al. (2011) proposed an optimization problem based on multi attribute utility for determining the time that is optimal to release Open Source Software’s next version. A multi-release Open Source Software reliability framework based on fault detection and correction process has been provided by Yang et al. (2016) in which the debugging is taken as a delayed fault removal process. Very few studies considering factors such as perfect or imperfect debugging (Goel, 1985), change point (Zhao, 1993) that help in assessing the reliability accurately under multi-release strategy for OSS projects have been done. Under perfect debugging assumptions, a multi-release model for OSS was proposed by Garmabaki et al. (2015). Singh et al. (2010) incorporated change point to assess the reliability growth for OSS based on the instructions executed so as to cater the diverse user profile.

In the present investigation, multi-release modelling for OSS considering the imperfect debugging assumptions and change point concept is studied on real life data set. In OSS, the debugging is carried out by a number of users which are connected through the internet across the world. The users take part in testing of OSS in its operational phase and report the bugs in the software which in turn helps to increase the reliability of the product. When the first release is being developed no reported bugs from the volunteers are there to improve the software quality. It is after the first release that bug reporting and new functionalities as per the market demand are incorporated to improve the quality of the next version. Addition of functionalities increases the fault content and the associated risk. In
The Impact of Ideology on the Organizational Adoption of Open Source Software
[www.igi-global.com/chapter/impact-ideology-organizational-adoption-open/29482?camid=4v1a](www.igi-global.com/chapter/impact-ideology-organizational-adoption-open/29482?camid=4v1a)

An Emotional Scene Retrieval Framework for Lifelog Videos Using Ensemble Clustering
[www.igi-global.com/article/an-emotional-scene-retrieval-framework-for-lifelog-videos-using-ensemble-clustering/126612?camid=4v1a](www.igi-global.com/article/an-emotional-scene-retrieval-framework-for-lifelog-videos-using-ensemble-clustering/126612?camid=4v1a)