

## Chapter 37

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

**Shozab Khurshid**

*University of Kashmir, Srinagar, India*

**A.K. Shrivastava**

 <https://orcid.org/0000-0001-7794-7129>

*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.*

DOI: 10.4018/978-1-7998-9158-1.ch037

## **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.

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.

21 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/generalized-multi-release-framework-for-fault-prediction-in-open-source-software/286601](http://www.igi-global.com/chapter/generalized-multi-release-framework-for-fault-prediction-in-open-source-software/286601)

## Related Content

---

### A Systematic Review of Attributes and Techniques for Open Source Software Evolution Analysis

Munish Saini and Kuljit Kaur Chahal (2021). *Research Anthology on Usage and Development of Open Source Software* (pp. 1-23).

[www.irma-international.org/chapter/a-systematic-review-of-attributes-and-techniques-for-open-source-software-evolution-analysis/286563](http://www.irma-international.org/chapter/a-systematic-review-of-attributes-and-techniques-for-open-source-software-evolution-analysis/286563)

### Two Level Empirical Study of Logging Statements in Open Source Java Projects

Sangeeta Lal, Neetu Sardana and Ashish Sureka (2015). *International Journal of Open Source Software and Processes* (pp. 49-73).

[www.irma-international.org/article/two-level-empirical-study-of-logging-statements-in-open-source-java-projects/170476](http://www.irma-international.org/article/two-level-empirical-study-of-logging-statements-in-open-source-java-projects/170476)

### Software for Feedback System Using Adaptive Categorization and Authenticated Recommendation

Ayan Banerjee and Anirban Kundu (2019). *International Journal of Open Source Software and Processes* (pp. 37-69).

[www.irma-international.org/article/software-for-feedback-system-using-adaptive-categorization-and-authenticated-recommendation/233513](http://www.irma-international.org/article/software-for-feedback-system-using-adaptive-categorization-and-authenticated-recommendation/233513)

### Open Source and Economic Models in an Evolutionary Approach

Marco Berlinguer (2023). *Business Models and Strategies for Open Source Projects* (pp. 18-49).

[www.irma-international.org/chapter/open-source-and-economic-models-in-an-evolutionary-approach/326638](http://www.irma-international.org/chapter/open-source-and-economic-models-in-an-evolutionary-approach/326638)

### Using Open Source to Building and Accessing Learning Objects and Online Courses

Christos Bouras and Maria Nani (2007). *Open Source for Knowledge and Learning Management: Strategies Beyond Tools* (pp. 266-297).

[www.irma-international.org/chapter/using-open-source-building-accessing/27815](http://www.irma-international.org/chapter/using-open-source-building-accessing/27815)