Chapter 34
Predicting Change Prone Classes in Open Source Software

Deepa Godara
Uttarakhand Technical University, Sudhowala, India

Amit Choudhary
Maharaja Surajmal Institute, Delhi, India

Rakesh Kumar Singh
Uttarakhand Technical University, Sudhowala, India

ABSTRACT

In today's world, the heart of modern technology is software. In order to compete with pace of new technology, changes in software are inevitable. This article aims at the association between changes and object-oriented metrics using different versions of open source software. Change prediction models can detect the probability of change in a class earlier in the software life cycle which would result in better effort allocation, more rigorous testing and easier maintenance of any software. Earlier, researchers have used various techniques such as statistical methods for the prediction of change-prone classes. In this article, some new metrics such as execution time, frequency, run time information, popularity and class dependency are proposed which can help in prediction of change prone classes. For evaluating the performance of the prediction model, the authors used Sensitivity, Specificity, and ROC Curve. Higher values of AUC indicate the prediction model gives significant accurate results. The proposed metrics contribute to the accurate prediction of change-prone classes.

DOI: 10.4018/978-1-7998-9158-1.ch034

Copyright © 2021, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

INTRODUCTION

The unending growing complexity and dependency has led to a rise in demand of high quality software that can be maintained at cheaper costs. Finding software change proneness is a significant and essential activity for improving software feature and reducing maintenance effort formerly the software is installed in real world. Koru and Liu (2007) proved change prone classes as a significant peripheral quality attribute that signifies degree of alterations in a class through various versions of software. Software industry is expanding manifolds day by day. Software changes to incorporate new features or to remove errors. This rapidly changing software demand has resulted in significant increase of effort from development to testing phase in software life cycle. Developing and maintaining software requires resources such as development time, cost to build and effort required. But all these resources are limited. Research has also been carried out to find the association between fault prone classes and object-oriented metrics. Weak classes of any software can be predicted using these quality attributes. Changes if predicted during earlier stages of life cycle, can help a developer in efficiently allocating project's resources by properly allocating the appropriate resources to weaker change prone class, so that such type of classes can be maintained properly and tested rigorously. Predicting such changes can be useful as such evaluations can be utilized to forecast changes from one release to next.

Maintenance phase is considered as one of the costly and significant phases of software. Malhotra and Khanna (2013) identified maintenance cost incurs 40-70% of entire cost of software. Estimation of change in classes i.e. probability with which class will modify or not needs to be evaluated as it can help in reducing maintenance cost and testing. As the software evolves it demands more rigorous testing so that good quality software can be developed with less changes and defects. By focusing on weak change prone classes utilization of resources can be done in a better way. Detection of such classes earlier in life cycle model of software can reduce maintenance costs as because if an error is detected early in a product, it would require lesser amount of resources to correct that error. Else in a later stage the cost of correcting an error increases exponentially in every unnoticed phase. Quality problems related to design can be identified in software before implementing codes, if developers are able to identify change prone classes early in life cycle of the software. Similarly, existing design can be customized, or alternate designs can be selected easily. These types of prediction models give high return on investment. As a result, change proneness prediction model contributes in improving quality of the software and reduces development cost also. Thus, change prediction model serves to deliver high quality software at optimal costs, as lesser changes and faults are carried forward in later stages of software life cycle.

Various object-oriented metrics are used throughout the software process. It is not possible to use a single metric to quantify various aspects of OO application. Various different metrics are required to completely analyse software. To predict change prone classes, various researchers have used various object-oriented metrics like size, cohesion, coupling, inheritance, etc. This research summarizes different object-oriented features which can be utilized to predict amount of change in classes. This will benefit researchers to get through various metrics elaborated here. In addition to that, it will help the researchers to predict more parameters for estimating changes in a class.

Researchers and Practitioners have used various object-oriented metrics throughout the software process. But it is not feasible to use a single metric for quantifying various aspects of Object Oriented application. Several different metrics and methods are needed for completely analyzing the software. Things that need to be considered before developing an efficient change proneness prediction model: (1) it is required to review the effectiveness of several methods as various methods may give different

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/predicting-change-prone-classes-in-open-source-software/286598

Related Content

Using Design of Experiments to Analyze Open Source Software Metrics for Change Impact Estimation
Miloud Dahane, Mustapha Kamel Abdi, Mourad Bouneffa, Adeel Ahmadand Henri Basson (2019). *International Journal of Open Source Software and Processes (pp. 16-33).*
www.irma-international.org/article/using-design-of-experiments-to-analyze-open-source-software-metrics-for-change-impact-estimation/228980

Tools Interoperability for Learning Management Systems
Nikolas Galanis, Enric Mayol, María José Casanyand Marc Alier (2017). *Open Source Solutions for Knowledge Management and Technological Ecosystems (pp. 25-49).*
www.irma-international.org/chapter/tools-interoperability-for-learning-management-systems/168978

Innovation, Imitation and Open Source
Rufus Pollock (2009). *International Journal of Open Source Software and Processes (pp. 28-42).*
www.irma-international.org/article/innovation-imitation-open-source/4088

Optimization Scenarios for Open Source Software Used in E-Learning Activities
Utku Köse (2018). *Optimizing Contemporary Application and Processes in Open Source Software (pp. 102-123).*
www.irma-international.org/chapter/optimization-scenarios-for-open-source-software-used-in-e-learning-activities/197108

The Virtual Computing Lab (VCL): An Open Source Cloud Computing Solution Designed Specifically for Education and Research
Andy Rindos, Mladen Voukand Yaser Jararweh (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications (pp. 1480-1492).*
www.irma-international.org/chapter/the-virtual-computing-lab-vcl/120982