Chapter 31
Enhancing Information Retrieval System Using Change-Prone Classes

Deepa Bura
Manav Rachna International Institute of Research and Studies, India

Amit Choudhary
Maharaja Surajmal Institute, India

ABSTRACT

In today's competitive world, each company is required to change software to meet changing customer requirements. At the same time, an efficient information retrieval system is required as changes made to software in different versions can lead to complicated retrieval systems. This research aims to find the association between changes and object-oriented metrics using different versions of open source software. Earlier researchers have used various techniques such as statistical methods for the prediction of change-prone classes. This research uses execution time, frequency, run time information, popularity, and class dependency in prediction of change-prone classes. For evaluating the performance of the prediction model, sensitivity, specificity, and ROC curve are used. Higher values of AUC indicate the prediction model gives accurate results. Results are validated in two phases: Experimental Analysis I validates results using OpenClinic software and OpenHospital software and Experimental Analysis II validates result using Neuroph 2.9.2 and Neuroph 2.6.

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 & 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

DOI: 10.4018/978-1-7998-9158-1.ch031

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 & 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 outcomes with different types of data sets. (2) Secondly it becomes essential to test whether the prediction model provides good results on another data set or not.

This study aims to build an efficient change prone prediction model that will predict change prone classes. The study includes the following objectives:

1. It aims to explore the association amongst different Object-Oriented metrics and change prone classes.