MACHINE-LEARNING APPROACH FOR MODELING SOFTWARE QUALITY

H.M.S.H. Bandara

Postgraduate Institute of Sciences

University of Peradeniya

Sri Lanka

Software Quality Management is an important aspect and has always been a priority in the software industry. Among the widely used metrics for software quality measurement, defect count with their frequency of occurrence is highly significant. Defects (bugs) represent an important profile of susceptibility of the system to crashes, failures or access violations and security breaches. Early detection of defects will therefore lead to the development of a robust and failure-proof system, a feature especially useful for large scale and long running projects. A host of prediction models have been developed and defect prediction has developed considerably. Directly measurable software process attributes have been used as reliable indicators for bug occurrence.

The primary objective of the present study was to explore the possible correlation between the directly-measurable software process attributes and software bugs. This will contribute notably to project management by allowing developers to quantitatively plan and steer software projects according to the expected number of bugs and their bug fixing effort. The aim was to develop a new a set of rules from the directly-measurable software process attributes that provide the best indication of software bugs using a machine learning approach.

Eclipse project was selected as the data source for bug prediction. Bug-relevant features were extracted from the downloaded log files of some randomly selected components of Eclipse. Subsequent connection was established to Bugzilla database to download the supplementary information related to bugs. The data were stored in a MySQL database and features were generated using a Java-based program. The J48 decision tree learner was applied to selected time frame for constructing the prediction model using the computed features (predictors) in file-level in the given time frame, together with the goal value (i.e. bug or no bug).
A Pareto analysis revealed that a feature called 'revision' has the highest frequency of occurrence and hence can be used as a reliable bug indicator along with few others. It further reveals that most recently revised and recently fixed components are more bug-prone.

It can be concluded that the features related to revision of a file such as number of revisions during a considered time period, lineOperationRRevisionis and GrownperMonth are highly important for defect prediction. In this study, the whole feature space derived from revision was not explored and hence, further studies are required.