Quantitative Evaluation of Software Quality Metrics in Open-Source Projects

Henrike Barkmann
Abstract

The validation of software quality metrics lacks statistical significance. One reason for this is that the data collection requires quite some effort. As a help to solve this problem, we develop tools for metrics analysis of a large number of software projects. Moreover, a validation of software quality metrics should focus on relevant metrics, i.e., correlated metrics need not to be validated independently.

First, we extract the source code and associated meta-data from the SVN and CVS repositories of SourceForge.NET projects. We developed SF Extract for collecting the metadata, which can connect to SourceForge, read the list of all Java projects available, follow the links to each project home page, parse the home page (HTML) to extract the relevant properties, and store them in the database for further processing. Since the metadata also contains information about the url of the SVN or CVS repository containing the source code, we could develop SVN Extract, which downloads the source code of these projects based on the information in the database in an automated and efficient way. The files and folders of the individual projects in the repository are stored in local working copies. To assure that the downloaded projects are complete and compiled, we import them manually into an Eclipse workspace as Java projects. So far, we could not find a way to automate this step in our process. The VizzAnalyzer metrics tool is fed with low-level information from the Eclipse project (syntax, cross references, etc.) and computes the metrics. We implemented an export engine to store the computed metrics in a database for later processing. For the actual statistical analysis of the metrics, we use MS Excel and SPSS both with access to the database.

We select 146 open-source Java projects randomly apart from the practical constraints. Altogether, 32% of the 146 projects needed manual fixes in order to enable analysis. The software metrics selected are a collection of the most popular metrics discussed in literature. They are taken from different well know metrics suites like Chidamber & Kemerer [5].

For all pairs of metrics (considered in this study) we aim at invalidating either of the following hypothesis:

H0 The pair of metrics values is independent in all software systems (considered in the study).

H1 The pair of metrics values is dependent, that is, showing a statistically significant correlation between the measured values in all software systems (considered in the study).

Our first contribution provides tool support for collecting large amounts of quantitative data on open-source software systems written in Java. Our second contribution reduces the number of metrics to validate. We could show correlation among the individual metrics, indicating that some of them seem to measure the same properties. The coefficients of correlation show strong connections between 5 metric pairs with results greater or equal 0.90. Four of the six metrics involved can be excluded, since they are redundant. Based on these findings, we can reject our hypothesis H0 for some pairs of metrics and thus support our alternative hypothesis H1. Finally, we describe metric values statistically, getting a first overview of the absolute value ranges for some wellknown metrics.

1http://sourceforge.net
2http://www.arisa.se
3http://www.spss.com
suits. They deliver hypotheses for possible absolute thresholds, which in turn could be used to identify classes that are threats to quality. Experimental evaluations of the thresholds is future work. We plan to extend our existing database with more projects. During this, we hope to be able to improve our process and tools to allow the fully automatic download and analysis of projects. The ultimate goal, however, is to experimentally validate Software Quality Models that are based on a number of independent, automatically assessable metrics by showing their correlations to well-accepted quality attributes, like costs of ownership, that are not automatically predictable.

The original paper has been accepted for publication at The 2009 IEEE International Workshop on Quantitative Evaluation of large-scale Systems and Technologies, May 26-29, 2009, Bradford, UK in conjunction with The IEEE 23rd International Conference on Advanced Information Networking and Applications (AINA-09). It will be published by the IEEE Computer Society Press and archived in the IEEE Digital Library.

References


