Calibration and Validation of a Maintainability Model

Model
The software Improvement Group (SIG) developed a Maintainability model based on the ISO 9126 standard [8]. The ISO 9126 defines Maintainability in terms of 4 sub-characteristics: Analyzability, Changeability, Stability, and Testability. This definition, however, does not provide clues on how to assess these sub-characteristics. SIG operationalized the model by introducing source code metrics to assess each of the sub-characteristic [6]. These metrics cover source code aspects such as volume, code duplication, coupling, complexity, among others.

Source Code Metrics
Metrics at different levels (unit, modules, component, system) are calculated via static analysis of source code.

Software Analysis Warehouse
All source code snapshots (6404) of all systems (273, from which 20 are open source) that SIG analyzed are represented in the Software Analysis Warehouse [7], including the raw metrics and ratings computed from them. These metrics are calculated for a total of 67 different languages.

Source Code Ratings
Ratings are computed for all elements of the model: system properties (volume, duplication, etc.), sub-characteristics (analyzability, changeability, etc.), and finally for maintainability.

Calibration
Together with the TUV Informationstechnik GmbH (TUViT), SIG uses the model to evaluate and certify the maintainability of software systems [3]. The model is re-calibrated yearly in order to keep up to date with the state-of-the-practice in software engineering. In each calibration cycle, new thresholds have so far been calculated for:
1) low level metric interpretation and aggregation (Metric Thresholds) enabling the distinction between good and bad coding [2];
2) mapping of source code measurements to star ratings (Profile Thresholds) [1].

Validation
A way to validate a model that calculates maintainability as a function of source code internal metrics is to test correlations with software development external metrics. Such external metrics can be derived from issue tracking systems, where defects and enhancements are recorded. Two empirical studies revealed more issues are solved [5] and faster [4] in the presence of more maintainable source code. This alone does not show causality, but increases the confidence in the ratings calculated by the SIG maintainability model.

Metric Thresholds
Metric thresholds enable a qualitative interpretation of quantitative measurements. These thresholds define four risk categories (low, medium, high and very high) for each metric [2].

Profile Thresholds
Profile thresholds map source code risk categories to ratings. These risk categories are a direct product of applying metric thresholds. [1]

Issue Trackers
Issue tracking systems record both defects and enhancements regarding software products. This data reveals external properties of the software products (e.g. number of defects).

Issue Handling Metrics
So far SIG as explored issue handling productivity metrics as proxies for developer productivity. We found significant positive correlations between the ratings produced by the model and both (1) issue resolution time (the time between opening and closing an issue) [4] and (2) the number of solved issues [5]. Interestingly, the correlations we found are stronger at the upper levels of the model, which seems to reveal a reinforcing effect of the model’s aggregation steps.

Correlation Analysis
The hypotheses being tested in the validation studies that SIG has been conducting follow a generic template: “Do the higher ratings as computed by the model correlate to better performance of software developers?” [4, 5]

References