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.

Calibration

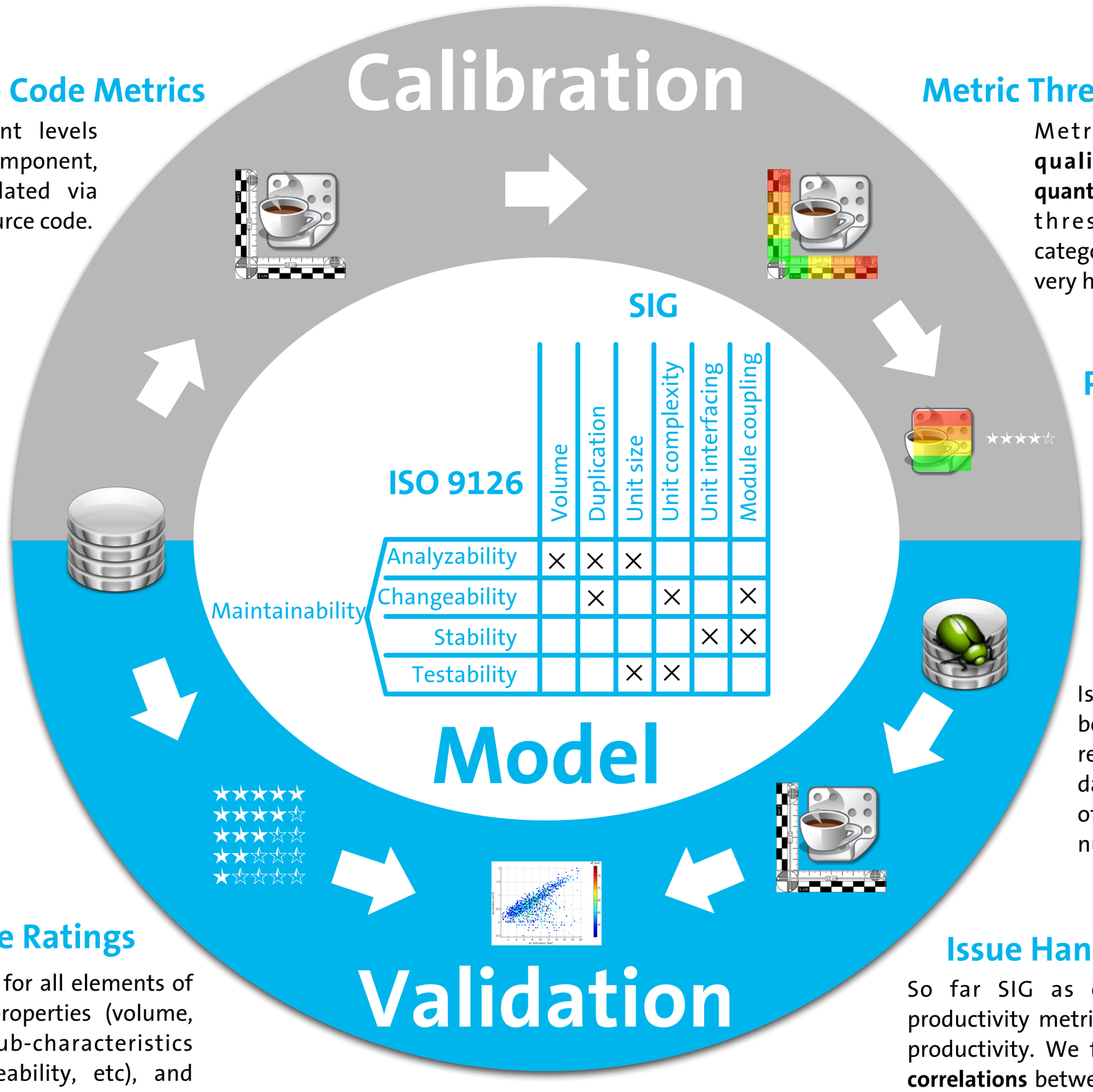
Together with the TÜV Informationstechnik GmbH (TÜViT), SIG uses the model to evaluate and certify the maintainability of software systems [3]. The model is re-calibrated yearly in order to keep up to date with the state-of-the-practice in software engineering. In each calibration cycle, new thresholds 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.



Source Code Metrics

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

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]

Software Analysis Warehouse

All source code snapshots (6040) 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.

Issue Trackers

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

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.

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]

Software Improvement Group
Miguel Ferreira
m.ferreira@sig.eu
http://www.sig.eu
http://twitter.com/sig_eu

References

[1] Tiago Alves, José Pedro Correia and Joost Visser, *"Benchmark-based Aggregation of Metrics to Ratings"*, under review, 2011
[2] Tiago Alves, Christiaan Ypma, and Joost Visser, *"Deriving Metric Thresholds from Benchmark Data"*, ICSM 2010, IEEE Computer Society, 2010
[3] Robert Baggen, José Pedro Correia, Katrin Schill and Joost Visser, *"Standardized Code Quality Benchmarking for Improving Software Maintainability"*, To be published in issue 20 of Software Quality Journal, Springer, 2011
[4] Dennis Bijlsma, Miguel Alexandre Ferreira, Bart Luijten and Joost Visser, *"Faster Issue Resolution With Higher Technical Quality of Software"*, To be published in issue 20 of Software Quality Journal, Springer, 2011
[5] Dennis Bijlsma, *"Indicators of Issue Handling Efficiency and their Relation to Software Maintainability"*, MSc thesis, University of Amsterdam, Amsterdam, The Netherlands, 2010
[6] José Pedro Correia, Yiannis Kanellopoulos and Joost Visser, *"A Survey-based Study of the Mapping of System Properties to ISO/IEC 9126 Maintainability Characteristics"*, ICSM 2009, September, 20-26, 2009, Edmonton, Alberta, Canada
[7] José Pedro Correia and Joost Visser, *"Benchmarking Technical Quality of Software Products"*, WCRE 2008, October 15-18, 2008, Antwerp, Belgium
[8] Ilja Heitlager, Tobias Kuipers, and Joost Visser, *"A Practical Model for Measuring Maintainability"*, QUATIC 2007, 30-39, IEEE Computer Society Press, 2007