QMetric - A Metric Tool Suite for the Evaluation of Software Process Data

Holger Schackmann, Martin Jansen, Christoph Lischkowitz, Horst Lichter
Research Group Software Construction, RWTH Aachen University
{schackmann|lichter}@swc.rwth-aachen.de {martin.jansen|christoph.lischkowitz}@rwth-aachen.de

Abstract

Configuration and change request management systems offer valuable information for the assessment of process quality characteristics. The definition of appropriate metrics that address the information needs of an organization is an intricate task. We present the QMetric tool suite which provides a general infrastructure for specifying metrics, relating them to organization-specific quality models, and automatic evaluation based on empirical comparison data.

1. Introduction

Managing a large portfolio of software products requires continuous monitoring of process quality characteristics. Collecting the required data by regularly status reporting can be expensive and intrusive and ignores the past history of the process. This motivates mining data from routinely collected **software process data** as it is available in change request management (**CRM**) systems or version control systems (**VCS**).

However existing metric tools are targeted at a single source of information, and typically provide only a number of fixed metric evaluations with limited adaptability [1]. Metrics appropriate for organization-specific information needs must be implemented in custom scripts. Hence developing and validating new metrics is time-consuming and costly. Furthermore available tools lack a flexible approach on how to model the relation of metrics to higher-level goals. This was the motivation for the development of the QMetric tool suite that offers the following characteristics:

- Concept to specify metrics in a declarative manner which simplifies metric development and validation.
- General infrastructure to evaluate metrics on software process data.
- Flexible tool support to define and evaluate quality models based on software metrics.

Figure 1 gives an architectural overview of the tool suite. The components and the related underlying concepts are explained in the following.

2. Declarative Metric Specifications

Metric calculation in the QMetric tool suite is based on user defined metric specifications that abstract from the way the information is stored. Basic building blocks of these specifications are filters for information fields of a change request (**CR**) (e.g. its severity), and events that occur in the history of a CR (e.g. reopen a CR).

Each metric specification contains a **base filter** that defines which CRs are considered in the calculation (e.g. only CRs for a certain product). Further on the considered time period and the time granularity are defined.

Then one of several predefined **value calculators** can be applied to calculate a value for individual CRs in each time interval. Examples are the calculation of the length of a time interval between two specified events in the lifecycle of a CR, or the calculation of the number of occurrences of certain events. Optional weights can be applied (e.g. weight based on estimated remaining workload). The outcome of value calculators can be combined with operations like sum or mean value to calculate a result for a certain time interval. This approach offers a large flexibility for the specification of metrics, and simplifies developing and validating metrics [3].

The core component of the tool suite is the **Metric Calculator** that implements the metric evaluation algorithm [1]. Its input is a declarative metric specification. The algorithm operates on information fields and events that abstract from the way the required information is retrieved. **Data Source Wrappers** facilitate that the Metric Calculator can operate on information from heterogeneous repositories in a uniform way. Each data source provides a number of fields, and an interface for accessing related events. Data Source Wrappers for CVS and Subversion are based on linking changes in a version control system (VCS) to related CRs. This enables to define metrics that consider VCS events (e.g. commit changes to files) and code size information.

The **Metric Query Tool** enables users to calculate metrics, browse metric results, and define and save their

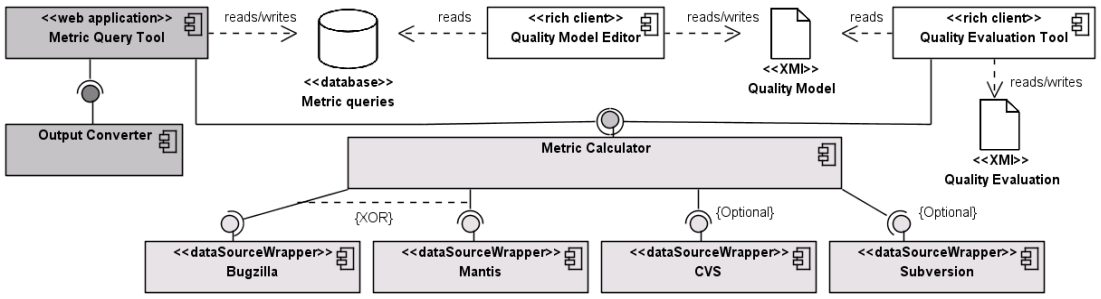

Fig. 1. Architectural Overview of the QMetric Tool Suite

own metrics by using graphical wizards accommodated to different expertise of the users.

The **Output Converter** is a server-side component that transforms metric results given in XML into several output formats, e.g. charts, or spreadsheets. Results can be enriched with links to related CRs which eases validation and interpretation of metric results.

3. User-defined Quality Models

In order to relate metrics to improvement goals we use hierarchical quality models that lean on the approach of bidirectional quality models introduced by Simon et al. [2]. Basic concepts are on the one side **quality characteristics** that reflect high-level requirements on the quality. An example of a quality characteristic is planning precision which can be subdivided into the quality characteristics adherence to schedule, adherence to planned effort, and process transparency.

On the other side the **quality indicators** are used to describe how quantitatively measurable attributes of an entity (i.e. product, process, or system) can be interpreted with respect to a quality characteristic. An approach to guide interpretation is for example the classification of measurement results according to the value distribution in a peer group of measured entities.

The **Quality Model Editor** enables to define such hierarchical quality models. In general the model is a directed acyclic graph (DAG). Source nodes represent quality indicators. Each quality indicator contains a declarative metric specification that defines how a metric is retrieved from a metric tool.

Inner nodes and sink nodes of the quality model represent quality characteristics. For each quality characteristic it must be defined how a value of the quality characteristic is calculated based on the values of the nodes connected by incoming edges as arguments. This is done by combining one or several basic functions. Available functions are the identity function, a threshold function, normalization to a certain value interval, a user-defined custom mapping, or the assignment of a value based on quantile classification with respect to a set of empiric values. Different weighting of incoming edges is typically expressed by a linear equation.

These functions enable to express a wide range of quality models. The fulfillment of an envisaged quality level can for example be modeled by using a threshold function at a sink node of a DAG. Different quality levels can then be modeled in sub graphs of the DAG with tightened thresholds in each level.

The **Quality Evaluation Tool** is used to evaluate the process quality with respect to a given quality model. In order to configure an evaluation the user has to specify:

- the quality model to be used,
- the entity to be evaluated (by defining a filter on the CRs related to a product or project)
- and optionally the comparison data (e.g. a group of similar projects, or an earlier time span)

The Quality Evaluation Tool can automatically trigger all required metric calculations. Results can be browsed in a tree view, drilled down to individual results for each time interval, and visualized.

The QMetric tool suite had been published open source. Further information and related case studies can be found on *www.qmetric.org*.

4. References

[1] L. Grammel, H. Schackmann, and H. Lichter (2007), "BugzillaMetrics - An adaptable tool for evaluating metric specifications on change requests", *Intl. Workshop on Principles of Software Evolution (IWPSE'07)*, Dubrovnik, Croatia, 2007, pp 35-38.

[2] F. Simon, O. Seng, T. Mohaupt, *Code Quality Management*, Dpunkt-Verlag, Heidelberg, 2006.

[3] H. Schackmann, H. Lichter, "Comparison of Process Quality Characteristics Based on Change Request Data", in R. Dumke et al. (Eds.): *Proc. of IWSM / MetriKon / Mensura 2008*, LNCS 5338, Springer, Berlin, 2008, pp. 127-140.