Measurement-based Software Quality Evaluation

Constanza Lampasona, Adam Trendowicz, Michael Kläs, Jens Heidrich

Fraunhofer Institute for Experimental Software Engineering

Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

{constanza.lampasona, adam.trendowicz, michael.klaes, jens.heidrich}@iese.fraunhofer.de

Abstract

Managing product quality during the development, operation, and maintenance of software-intensive systems is a challenging task. Although many organizations have already identified various quality aspects they need to measure, control, and improve, a standard process for quality evaluation is still missing. The quality models and quality evaluation methods proposed in recent decades are typically missing an associated improvement-oriented evaluation process. Even though an ISO/IEC standard for product quality evaluation (ISO/IEC 14598) exists, it has a number of weaknesses that makes it hard to effectively apply this process in practice [10]. This article proposes a continuous quality evaluation process that is founded on well-known and widely accepted technologies such as the Quality Improvement Paradigm and the Goal/Question/Metric approach. The objective is to define a comprehensive, goal-oriented, and adaptable process that provides mechanisms for quantitative feedback and thus facilitates continuous improvement. This work is being conducted in the context of the German research project Quamoco. The article summarizes existing approaches, describes the principles of the proposed evaluation process, illustrates current and future challenges, and gives an outlook on future work in the context of Quamoco.

Keywords

Software quality model, quality evaluation, measurement, aggregation

1 Introduction

Nowadays, the demand for high-quality software is growing rapidly. This growth is related to the increasing complexity of software, shorter development cycles, and fast-changing technologies. Effective and efficient approaches are required for managing the quality of software-intensive systems during their development, operation, and maintenance. Modeling the quality of software has become one of the key aspects of software quality management. In general, quality models define what quality means in a particular context. For utilizing such a quality model in practice, a quality evaluation process is needed that makes (quantitative) statements about the quality of an evaluated artifact.

A number of different approaches for software product evaluation have been proposed over the last decades. Typical approaches include quality evaluation meth-

ods that focus on *particular* software products (subject) and use *particular* evaluation techniques to determine *particular* characteristics of a product [11]. Example subjects include requirement documents, architecture descriptions (e.g., [6]), source code, and executable software. Example techniques include static or dynamic analysis (e.g., static code analysis [4]), reviews and inspections, or conformity assessments. Instead of defining one systematic evaluation process, a couple of evaluation procedures have been defined, driven by a particular (fixed) context, objective, technique, and quality model. Furthermore, there is a lack of feedback on the performance of the evaluating process and no continuous improvement of the process itself.

The approach presented in this paper provides a systematic evaluation process focusing on continuous improvement aspects of an organization based on QIP (Quality Improvement Paradigm) [2]. The development is done as part of the German public research project Quamoco (grant no. 01 IS 08 023 C), which focuses on developing a comprehensive software quality standard for different application domains. In section 2, we provide an overview of related work; section 3 of this article describes the principles of the proposed evaluation process; section 4 illustrates current and future challenges (such as dealing with missing or mixed-type data); and section 5 gives an outlook on future work in the field.

2 Related Work

Many approaches attempt to provide a framework for the assessment of software quality. The ISO/IEC standard 14598 [7] for software product quality evaluation is a partial attempt to address the issues described in section 1 by defining a generic product quality evaluation process. The intention was to replace the process initially defined as part of the ISO/IEC 9126 standard [9], i.e., the software quality model. However, it has been heavily criticized in the software community. Typical problems in utilizing this process are related to [12]: (1) insufficient support for specifying and evaluating the achievement of evaluation goals, (2) implicit relationships between activities, (3) no attention to the trade-offs between goals and resources, and (4) insufficient feedback on process performance.

GQM+Strategies® [3] is an approach for explicitly linking measurement goals to an organization's higher-level goals, and also to goals and strategies at the level of the overall business with a focus on software-related organizations. It is based on the Goal/Question/Metric paradigm (GQM) [1]. Selection and adaptation of predefined goals and strategies as well as definition of new goals and strategies is driven by *context factors* and *assumptions*. Context factors are environmental variables that represent the organizational environment and affect the kind of models and data that can be used. Assumptions are estimated unknowns that can affect the interpretation of the data.

The W-Process, as proposed by Punter et al. [12], also uses the GQM approach for specifying evaluation goals. However, there is no explicit treatment of context factors and underlying assumptions, which are required for interpreting the outcome of the evaluation process. Furthermore, the mechanisms for continuous improvement of the evaluation process itself are limited (e.g., regarding quantitative feedback on its performance and identification of improvement potentials). Moreover, there is no linkage between evaluation goals and other organizational objectives (project, IT, business) and no systematic process for deriving product evaluation goals from business objectives. For example, the W-Process proposes tracking evaluation costs and performing a trade-off analysis between evaluating goals and resources, but it is not clear where the cost objective comes from, and whether it is the only important objective interfering with the evaluation objectives.

The international standard ISO/IEC 15939 [8] defines a measurement process. It identifies measurement steps within an overall project or organizational measurement structure.

Having a well defined software quality evaluation method only makes sense if it is integrated in a continuous SPI (software process improvement) cycle. There are many well known approaches that support process improvement, such as PDCA (Plan-Do-Check-Act) [5], Six Sigma [13], and QIP. We chose QIP as a framework because it explicitly focuses on software development and does not make use of statistical control, as developing software is human-based and not a manufacturing activity.

3 A Process for Software Quality Evaluation

This section describes the major activities of a quality evaluation process that addresses the deficits previously identified. We propose founding the process on the well-known and widely accepted QIP, which consists of six fundamental steps: characterize context; set goals; choose process; execute process; analyze data; and package knowledge and experiences.

The objective of the quality evaluation method is to support a software organization in two areas:

- **Organization level**: At this level, the standard quality model and the associated methods and processes developed or adjusted for a certain application context are subject to continuous improvement - like any other technology within a learning software organization.
- **Product level**: At this level, the quality model defined and maintained at the organization level is applied for assessing, analyzing, and improving the quality of concrete software products.

The quality assessment method we propose is embedded into two improvement cycles (Figure 1). Although the product-level improvement cycle is presented in

the figure as part of the "Execute Quality Assessment" of the organization-level cycle, in practice, multiple product-level improvement cycles can run in parallel to the organization-level improvement cycle.

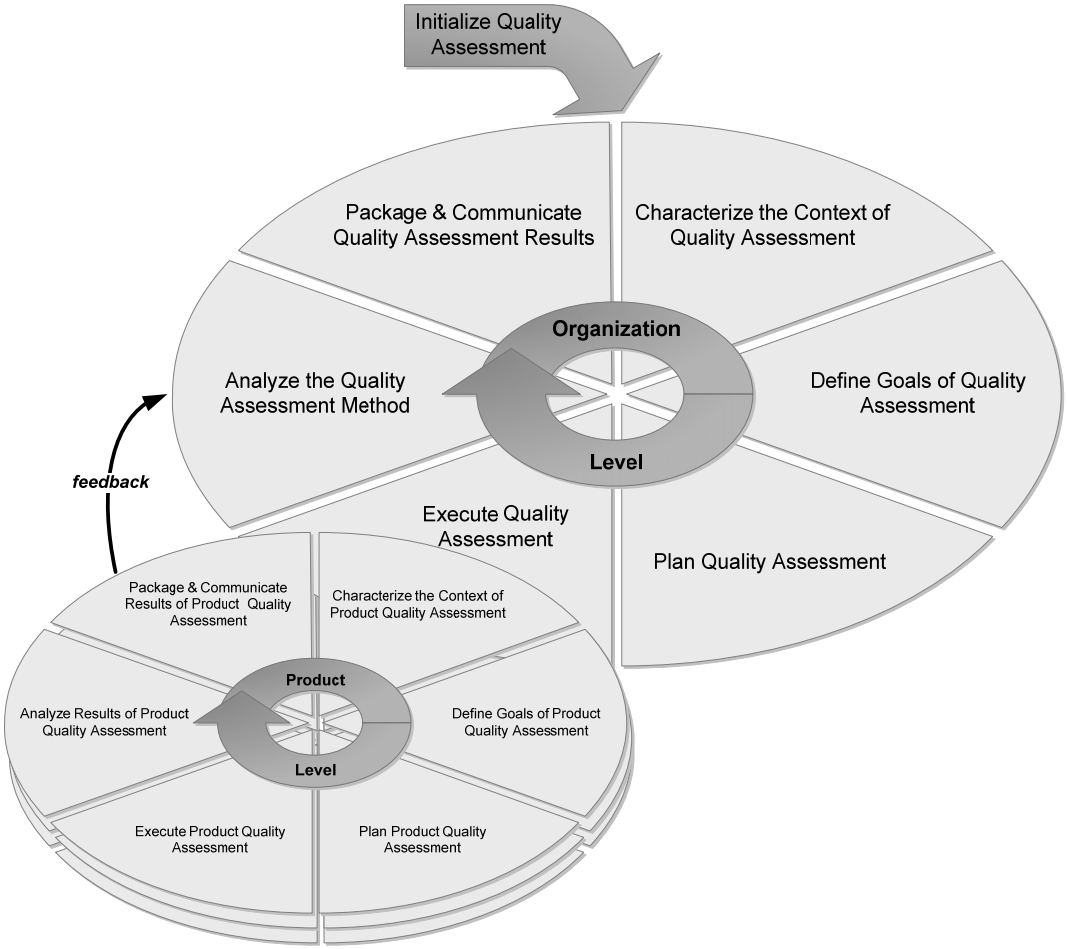

Figure 1 Overview of the Quality Assessment Method

3.1 Initialize Quality Assessment

The objective of this step is to prepare the implementation of the quality assessment method. Like any other technology introduced in the context of a software development organization, it requires the commitment of all involved parties, assignment of required resources and infrastructure, and training of the personnel involved.

This step includes the following activities:

- Getting commitment from high-level management and personnel involved in the product's quality assessment (in particular, persons on whom the successful application of the quality assessment method depends, e.g., personnel collecting measurement data);

- Obtaining resources required for applying the quality assessment method (e.g., budget, personnel);
- Preparing the infrastructure required for the successful application of the quality assessment method (e.g., measurement, data storage, and analysis software tools);
- Training involved personnel, if necessary (e.g., making appropriate personnel familiar with the quality assessment method).

3.2 Characterize the Context of Quality Assessment

The objective of this activity (QIP: Characterize context) is to determine the application scope of the quality assessment method and to explicitly document the actual and assumed characteristics of the quality assessment context. The context characteristics include attributes that (1) may be potentially useful by interpreting the data on the quality assessment method's performance (outputs of the "Execute Quality Assessment" step), and (2) explain potential deficits in the performance of the quality assessment method, identified in the "Analyze the Quality Assessment Method" step during the previous execution of this iterative process.

The specification of the application scope includes, for instance, determining those parts of the organization in which the software quality assessment is going to take place.

The actual characteristics of the quality assessment environment are referred to as context characteristics and are the actual values (probability = 1) of environmental characteristics. Assumptions, on the other hand, refer to uncertain, assumed values (probability < 1) of the environmental characteristics.

Both context characteristics and assumptions cover the relevant environmental attributes responsible for the successful application of the quality assessment method, i.e., those having a significant impact on the performance of the method. The relevance of certain environmental characteristics can be based on human expertise (subjective beliefs) or on the results of an analysis of quantitative data collected from already completed (historical) applications of the quality assessment method (i.e., software product quality assessments).

The involved stakeholders are a special case of the quality assessment context. Their characteristics (expertise, expectations, availability, etc.) typically have a significant influence on the performance and results of quality assessment.

Context and assumptions should be revised and updated throughout all steps of the quality assessment method, not only in the "Characterize the Context of Quality Assessment" step. It may, for example, happen that some of the context factors that were initially considered later on turn out not to be relevant, while others that initially were not considered do. On the other hand, it may be that initial assumptions can be clarified and turned into context characteristics, based on the informa-

tion available in the later steps of the quality assessment method. This step is summarized in the following activities:

- Determining the application scope of the quality assessment (e.g., those parts of the organization in which the quality assessment is going to be performed);
- Specifying characteristics of the context of the quality assessment (e.g., if there is a limited budget);
- Specifying (explicitly) assumptions made while introducing and applying quality assessment (e.g., reducing defects by 10% reduces maintainability costs by 5%);
- Specifying stakeholders involved in the quality assessment and their characteristics (e.g., the software architect who provides information about design issues).

3.3 Define Goals of Quality Assessment

The objective of this activity (QIP: Set goals) is to define the goals of the quality assessment. Different types of goals are defined at the organization level and at the product level. Organization-level goals refer to objectives regarding the performance of the quality assessment method. Product-level goals refer to objectives regarding the quality of a software product.

Both types of goals need to be aligned with goals and strategies on other organizational levels (e.g., business, IT, etc.). A simple reason for this is that software quality assessment, like any other activity within an organization, must be justified by organizational strategic business objectives and must stay in harmony with all related goals.

Aligning Goals

For the purpose of deriving and aligning goals across all organizational levels, we recommend using the GQM^{+}Strategies® method [2], which is abstractly illustrated in Figure 2.

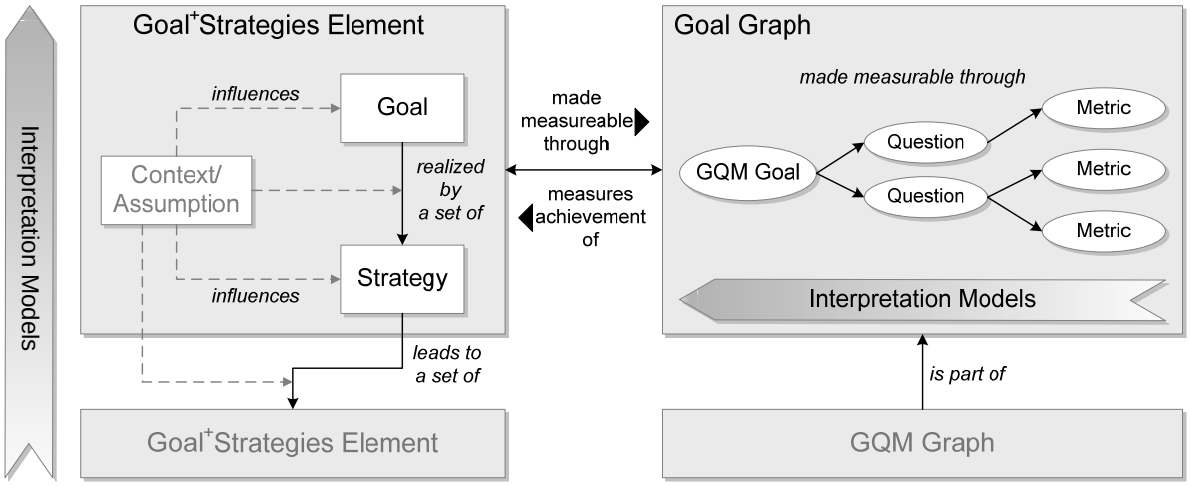

Figure 2 The GQM+Strategies® method

GQM+Strategies® is a measurement planning and analysis approach, whose output is a detailed and comprehensive model that defines all the elements necessary for a measurement program. In extending GQM, the GQM+Strategies® approach makes the business goals, strategies, and corresponding lower-level goals explicit. Strategies are formulated to deal with business goals such as improving customer satisfaction, garnering market share, reducing production costs, and more, taking into account the context and making any assumptions explicit. Strategies also help to define lower-level goals that can be assigned to different parts of the organization, e.g., software goals, hardware goals, marketing goals, etc. The number of goal/strategy levels depends on the structure of an organization. In the context of Quamoco, we mainly focus on software-related goals because we are concerned with relating software quality measurement and evaluation to higher-level business goals. GQM+Strategies® also makes explicit the relationships between goals/strategies and measurement goals. Measurement goals are broken down into concrete metrics using the GQM approach. Interpretation models (based on the metrics) are defined for determining whether a strategy was successful and a related goal could be achieved. Attached to goals and strategies at each level of the model is information about relevant context factors and about assumptions. The entire model provides an organization with a mechanism for not only defining measurement consistent with larger, upper-level organizational concerns, but also for interpreting and rolling up the resulting measurement data at each level. GQM+Strategies® linkages and measures ensure that the business goals are fulfilled.

Quantifying Goals

In order to make each of the defined goals verifiable quantitatively, appropriate measures need to be assigned. For this purpose, we propose using the GQM approach [1]. Figure 3 illustrates the principal elements of the GQM approach.

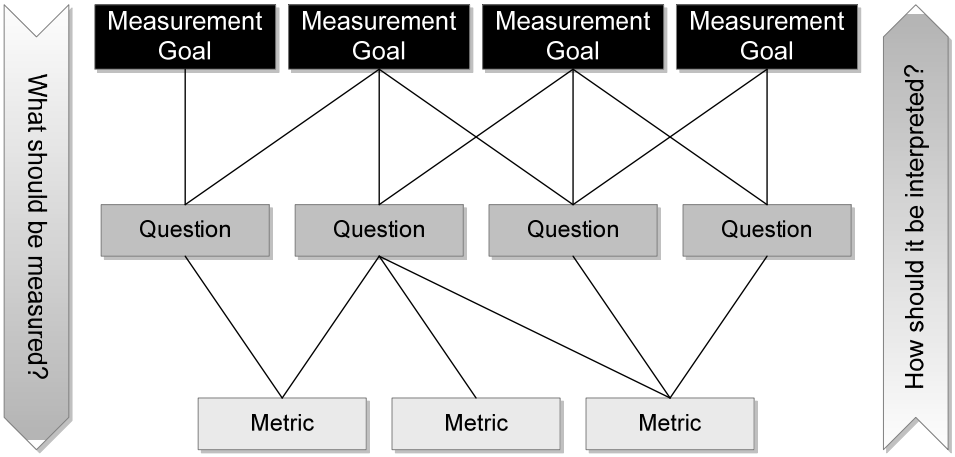

Figure 3 The GQM method

Within the GQM approach, each goal is formalized by specifying the following aspects, within a so-called goal template:

- Object: What is the object of interest and on which level of granularity should the object be analyzed? Typical object categories include software products, processes, and resources.
- Purpose: What is the purpose of measurement?
- Focus: What is the quality aspect/focus to be addressed?
- Viewpoint: Who is the stakeholder, who is interested in getting the analysis results?
- Context: The context in which the analysis takes place.

Table 1 illustrates differences between organization- and product-level goals by means of a GQM goal template.

Goal Template	Organization Level	Product Level
Object	Quality assessment method	Software product
Purpose	Improvement	Assessment
Focus	Performance	Quality aspect
Viewpoint	Management	Project/quality manager
Context	Organization/domain	Project

Table 1 Organization- vs. product-level GQM goal template

Besides formalizing goals and deriving appropriate measures, GQM supports specifying means for interpreting measurement results and verifying achievement of associated goals. Interpretation means may include thresholds on measurement

data, decision rules, and mechanisms for aggregating evaluation results from measures to goals. This step contains the following activities:

- Defining quality assessment goals (organization- and product-level):
    - Identify the object of the assessment (e.g., a software product);
    - Specify the purpose of the assessment (e.g., evaluation);
    - Specify the subject of the assessment (e.g., defects);
    - Specify the perspective of the assessment (e.g., quality assurance manager);
    - Revise and update (if appropriate) the assessment context and assumptions;
- Aligning quality assessment goals to relevant objectives and strategies at all organizational levels (e.g., decrease customer-reported software defects);
- Defining measures for defined goals (e.g., #defects /#KLOC);
- Defining decision criteria (threshold and targets) for defined measures (e.g., #defects/#KLOC<10);
- Defining aggregation mechanisms (e.g., using the simple weighted sum of measurements).

3.4 Plan Quality Assessment

The objective of this activity (QIP: Choose process) is to design the execution of the quality assessment according to goals, measures, and interpretation models defined in the "Define Goals of Quality Assessment" step. This includes preparing the required infrastructure, planning available resources (e.g., personnel and budget), and scheduling activities within the three remaining steps of the quality assessment method (i.e., execute, analyze, and package). For example, a measurement plan is created that specifies which measures should be collected by whom, when, and how often. Moreover, data collection, analysis, storage, and visualization tools are selected and appropriate resources and necessary trainings are associated. This step can be summarized in the following activities:

- Preparing the measurement plan;
- Selecting (and adapting) evaluation techniques and methods;
- Planning the required resources and infrastructure (e.g., tool support);
- Planning the necessary trainings;
- Preparing a detailed schedule.

3.5 Execute Quality Assessment

The objective of this activity (QIP: Execute) is to collect measurement data according to the measurement plan and to analyze it according to interpretation models defined in the "Define Goals of Quality Assessment" step. In this step, quality assessment is performed on the product level, probably for multiple products.

Characterize the Context of Product Quality Assessment

In this step, specific characteristics of a particular software project where quality assessment of output products is to be performed are determined. It is important to identify potential deviations of the project context from the context defined on the organizational level ("Characterize the Context of Quality Assessment" in section 3.2). Identified deviations may help to clarify a possible deficit in the performance of the quality assessment on the organization level. This step includes the following activities:

- Specifying project-specific context characteristics and assumptions (e.g., defect slippage data available from past similar projects);
- Identifying potential deviations from the context/assumptions made on the organization level.

Define the Goals of Product Quality Assessment

In this step, specific product quality goals on the project level are defined (e.g., analyze the code for the purpose of evaluation of the reliability from the point of view of the quality assurance manager in the context of the software development department of the company). Based on these goals, an appropriate quality model from the organization level is adapted for the purpose of product quality assessment. If a need for critical changes in the quality model is identified on the product level (e.g., deficits that make the model non-operational), it should be communicated to the organization level where appropriate improvements can be made.

Plan Product Quality Assessment

In this step, quality assessment activities within the project are planned, i.e., a project-level measurement plan is prepared, measurement and analysis activities are scheduled, and resources are assigned.

Execute Product Quality Assessment

In this step, measurement data are collected (e.g., #defects and #LOC) and validated (e.g., are the data consistent and complete?) according to the measurement plan.

Analyze Results of Product Quality Assessment

In this step, the measurement data collected in the project are analyzed and interpreted according to the quality model and the associated interpretation rules and procedures. This phase of the assessment process includes applying decision criteria, aggregating evaluation results, interpreting this information (e.g., if code complexity is greater than 0.8, then, in order to provide maximum understandability, code should contain at least one comment per line of code), identifying improvement potentials of the product and development process, and planning additional evaluation work (if needed).

Package and Communicate Results of Product Quality Assessment

This step includes packaging the results of the product quality evaluation (e.g., analysis results, experiences, and improvement potentials), storing these results in the organizational data repository, and communicating the outcome to the appropriate stakeholders.

3.6 Analyze the Quality Assessment Method

The objective of this activity (QIP: Analyze) is to analyze the performance of the quality assessment method (i.e., its contribution to the achievement of the goals defined in the "Define Goals of Quality Assessment" step) on the basis of measurement data collected on the product level. In this step, potential deficiencies are identified and improvement actions are planned for future process applications. In this step, interpretation models defined in the "Define Goals of Quality Assessment" phase are applied in terms of the context characteristics and assumptions defined in the "Characterize the Context of Quality Assessment" phase and updated throughout the subsequent phases.

3.7 Package and Communicate Quality Assessment Results

The objective of this activity (QIP: Package) is to package, store, and communicate the results of the analysis phase as well as relevant experiences regarding the application of the quality assessment method. These data should be the basis for implementing improvements in the next application of the quality assessment method (on both the organization and the product levels).

4 Challenges: Measure, Evaluate, and Aggregate

Independent of the specific evaluation procedure, there are three major activities in each product evaluation: measuring, evaluating, and aggregating. Evaluation and aggregation activities may occur in a different order depending on the chosen evaluation procedure, or even in cascades of evaluations and aggregations. Based

on our experience, each of these three activities is afflicted with its own challenges.

Measurement: In order to evaluate product quality, product (and process) properties have to be measured that affect or indicate the quality of the product.

C1. Reliability of measurement: If the same property of the same product is measured twice (e.g., by different persons), how certain are we to obtain the same result (i.e., how objective is the measure?).

C2. Validity of measures: Do the chosen measures sufficiently operationalize the concept of product quality in this context? In other words, (1) do the measures really measure what they claim to measure and (2) are the chosen measures sufficient for capturing the measured concept in the considered context?

C3. Cost of measurement: How much effort is required to conduct the measurement? To which degree can the measurement be automated? Are expensive measurement tools or infrastructures required?

Evaluation: Collected measurement data have to be evaluated to decide whether the considered product is, e.g., excellent, acceptable, or poor with respect to a certain quality aspect.

C4. Reliability of evaluation: If the evaluation is repeated (e.g., by a different person), how certain are we to obtain the same result (i.e., are there evaluation guidelines or is there a defined evaluation criterion? How objective is the criterion? How does the evaluation approach cope with missing data?).

C5. Validity of evaluation: What is the rationale behind the evaluation? If there is an evaluation criterion, where does it come from? Is it valid in the evaluation context? What are the prerequisites that have to be fulfilled by an evaluator?

C6. Cost of evaluation: How much effort is required to conduct the evaluation? To which degree can the evaluation be automated?

Aggregation: Finally, product quality is a multifaceted concept; therefore, different kinds of measurement data or evaluation results have to be aggregated in order to decide on the overall product quality.

C7. Reliability of aggregation: If the aggregation is repeated (e.g., by a different person), how certain are we to obtain the same result, i.e., are there aggregation rules or fixed weighting factors for aggregating quality aspects? How does the aggregation approach cope with missing data?

C8. Validity of aggregation: How do we assure the validity of the aggregation results if measurement data of mixed type (e.g., nominal, ordinal, interval, and ratio) are aggregated? Dependent on the evaluation context, different quality aspects may contribute to a different extent to the overall product quality. Does the aggregation consider this fact, e.g., by using appropriate weighting?

C9. Transparency of aggregation: Aggregation means loss of information. Are the rationales behind the aggregation intuitive and easy to understand (i.e., avoiding "magic numbers" as aggregation results) and is it possible to drill down through the aggregated information?

5 Conclusions and Further Work

This paper exposes the need for an adequate software quality evaluation process, which should be systematic and able to provide quantitative feedback in order to make continuous improvement possible. We propose a systematic quality evaluation process founded on the Quality Improvement Paradigm. This process addresses the problems present in existing quality evaluation models. We also explore the challenges related to measuring, evaluating, and aggregating, the major activities for product evaluation. These challenges are the reliability of the results obtained, their validity and cost, and, in the case of aggregation, its transparency.

Further work includes the instantiation and empirical evaluation of the Quamoco evaluation process with the aim of evaluating and measuring quality aspects based on a tailored quality model. This objective also includes the possibility of project monitoring and controlling. Furthermore, active involvement in the work on the quality series of standards ISO/IEC 2504x, the successors of the ISO/IEC 14598-x standards, is planned.

References

1. Basili V., Caldiera G., and Rombach D., "Goal, Question Metric Paradigm", J.J. Marciniak, ed., *Encyclopedia of Software Engineering*, vol. 1, John Wiley and Sons, 1994.

2. Basili V.R., and Rombach D.H., "The TAME project: towards improvement-oriented software environments", IEEE TSE, vol. 14, no. 6, pp. 758–773, 1988.

3. Basili V.R., Heidrich J., Lindvall M., Münch, J., Seaman C.B., Regardie M., Trendowicz A., "Determining the impact of business strategies using principles from goal-oriented measurement". In: Hansen, Hans Robert (Hrsg.) ; Karagiannis, Dimitris (Hrsg.); Fill, Hans-Georg (Hrsg.); Österreichische Computergesellschaft (OCG): Business Services: Konzepte, Technologien, Anwendungen. 9. Internationale Tagung Wirtschaftsinformatik. Band 1. Wien : Österreichische Computer Gesellschaft, pp. 545-554, 2009.

4. Briand L.C., Wüst J., Daly J.W., and Porter V.D., "Exploring the relationship between design measures and software quality in object-oriented systems", Journal of Systems and Software, vol. 51, pp. 245–273, 2000.

5. Deming W.E. Out of the crisis. Massachusetts Institute of Technology, Cambridge, Mass., 1986.

6. Dobrica L., and Niemela E., "A survey on software architecture analysis methods", IEEE TSE, vol. 28, no. 7, pp. 638–653, 2002.

7. International Organization for Standardization (ISO), "ISO/IEC 14598-1:1999(E):

Information technology - Software product evaluation - Part 1: General overview", 1999.

8. International Organization for Standardization (ISO), "ISO/IEC 15939:2007: Systems and software engineering – Measurement process", 2007.

9. International Organization for Standardization (ISO), "ISO/IEC 9126-1:2001(E): Software engineering - Product quality - Part 1: Quality model", 2001.

10. Plösch R., Gruber H., Hentschel A., Körner Ch., Pomberger G., Schiffer S., Saft M., Storck S., "The EMISQ Method - Expert Based Evaluation of Internal Software Quality", 31st IEEE Software Engineering Workshop (SEW 2007), pp.99-108, 2007.

11. Punter T., "Developing an evaluation module to assess software maintainability", Proceedings of Conference on Empirical Assessment in Software Engineering (EASE), Keele, 1998.

12. Punter T., Kusters R.J., Trienekens J.J.M., Bemelmans T., and Brombacher A., "The W-process for software product evaluation: a method for goal-oriented implementation of the ISO 14598 standard", Software Quality Journal, vol. 12, no. 2, pp. 137–158, 2004.

13. Tayntor C.B. Six Sigma Software Development. Boca Raton: Auerbach Publications, 2002.