

Facilitating the Practical Evaluation of Knowledge-Based Systems and Organizational Memories Using the Goal-Question-Metric Technique

Authors:

Markus Nick
Klaus-Dieter Althoff
Carsten Tautz

Submitted to the
Knowledge Acquisition Workshop 99,
Track "Evaluation of KE Techniques"

IESE-Report No. 029.99/E
Version 1.0
May 31, 1999

A publication by Fraunhofer IESE

Fraunhofer IESE is an institute of the
Fraunhofer Gesellschaft.
The institute transfers innovative software
development techniques, methods and
tools into industrial practice, assists com-
panies in building software competencies
customized to their needs, and helps them
to establish a competitive market position.

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
Sauerwiesen 6
D-67661 Kaiserslautern

Executive Summary

It is an important industrial need to deliver high-quality knowledge-based systems and organizational memories (e.g., to support service management or knowledge management in general). Evaluation is required to ensure this high quality and guide the development and maintenance. We present an approach for facilitating practical evaluation of knowledge-based systems and organizational memories that meets the requirements for good measurements in knowledge engineering. The base of this methodology is the Goal-Question-Metric (GQM) technique, which is an industrial-strength technique for goal-oriented measurement and evaluation from the field of software engineering. The practical benefit of GQM is demonstrated by a case study where GQM was applied to an existing case-based reasoning system/application.

Table of Contents

1	Introduction	1
2	GQM – An Appropriate Approach To Implement “Good” Measurements for KBSs and OMs	3
2.1	GQM: Objectives and Basics	3
2.2	How Does GQM Ensure Good Measurements?	3
3	The Application of The GQM Technique to CBR-PEB	9
3.1	The System CBR-PEB	9
3.2	The GQM Process for CBR-PEB	9
4	Systematic Evolution of the GQM Program	16
4.1	Evaluation for One Domain	17
4.2	Roll-Out for Several Domains	19
5	Related Work	20
6	Summary and Conclusion	21
	References	23

1 Introduction

Knowledge management is seen as a critical factor for an enterprise's competitiveness and success. This requires the optimal use and capturing of the resource "knowledge" for enabling learning from experience, continuous process improvement, and the extension of a company's creativity potential (Abecker et al., 1998; Prahalad and Hamel, 1990; Althoff et al., 1999b). Knowledge-based systems (KBSs) and organizational memories (OMs)¹ can be employed to (partially) automate and facilitate this task.

It is widely accepted that there is an industrial need to deliver high-quality KBSs (Benjamins et al., 1997) as well as high-quality organizational memories (Abecker et al., 1998). To ensure this need for high quality, evaluation is required that can guide the development and maintenance of a KBS or OM (Benjamins et al., 1997; Kirchhoff, 1994). Problems with existing evaluation studies (Menzies, 1998a) show that there is a need for a systematic approach that helps conducting "good measurements". In (Menzies, 1998c; van Harmelen, 1998) it is stated what are the requirements for good measurements in the field of KBSs. For OM this was also worked out in (Nick and Tautz, 1999). These requirements are mainly based on the existing knowledge in the field of software measurement (Basili, 1992; Fenton, 1991; Rombach, 1991) and evaluation in artificial intelligence (Cohen, 1995).

That both Artificial Intelligence (AI) and Software Engineering (SE) have some joint interest is nothing new and can be seen from events like the annual International Conference on Software Engineering and Knowledge Engineering (SEKE) and many others. While SE has its particular strengths in the systematicity of its approaches and its inherent focus on real-life applications, AI provides research results concerning the development of innovative software products as well as theoretical foundation². Evaluation of KE methods is a topic that is extremely important for the development and improvement of experimental prototype systems as well as for dealing with real-life applications. As such it lies in the center of both AI and SE. While the awareness of such approaches has increased in the AI community in the last years (e.g., in the workshops on

1 For the purpose of this paper, we see an organizational memory (OM) as a knowledge-based system (KBS). A KBS can store formal as well as informal knowledge. A KBS can also include the organizational infrastructure that is required to set up, use and maintain the system. In the field of software engineering we call such a KBS an experience factory (Basili et al., 1994a). The experience base is the facility that is used for storing the knowledge.

2 Of course, this is not intended to give a complete picture of the respective fields but rather to underline certain strengths in these fields to achieve some synergy in combining (parts of) them.

knowledge acquisition, modeling, and management: van Harmelen, 1998), Experimental Software Engineering (ESE) is already a well established subfield in SE (Basili, 1992).

(Menzies and van Hamelen, 1999) differentiate between

- big evaluations for evaluating KA/KE methodologies,
- small evaluations for evaluating KA/KE components,
- micro evaluations for evaluating a particular KA/KE component.

This is a similar definition as given in (Althoff, 1995) and (Althoff, 1997) who describe the most detailed "small evaluation" currently available with respect to case-based reasoning systems. In the meantime this has been supplemented by a micro evaluation of one concrete case-based reasoning application (Nick and Tautz, 1999), a continuation of which is described in this paper. Based on this evaluation work the ground has been prepared for defining methodologies for developing case-based reasoning and/or experience factory applications (Althoff and Aamodt, 1996; Althoff and Bartsch-Spörl, 1996; Bergmann and Althoff, 1998; Bergmann et al., 1999; Althoff et al., 1998; Althoff et al., 1999b) and hopefully in the future for evaluating these methodologies.

This paper presents a methodology that facilitates the evaluation of knowledge-based systems and organizational memories by ensuring inherently that the requirements for good measurements are addressed, that is, it systematically helps to select good, meaningful measures, interpret the measurement data, improve system and evaluation, and learn about system and evaluation. This methodology was successfully applied to evaluate the existing system CBR-PEB (CBRPEB, 1998; Nick and Tautz, 1999).

The evaluation methodology is based on the Goal-Question-Metric (GQM) paradigm (Rombach, 1991) for goal-oriented measurement and includes the process and guidelines for the application of GQM (Gresse et al., 1995; Briand et al., 1996). GQM is an industrial-strength technique that has been successfully used in the field of software engineering at, for example, NASA-SEL, Robert Bosch GmbH, Allianz Lebensversicherungs AG, Digital SPA, Schlumberger RPS (CEMP Consortium, 1996).

Section 2 shows why GQM is an appropriate approach for the evaluation of KBSs with respect to the requirements for "good measurements". Section 3 illustrates how GQM was applied to an existing OM as a case study. Section 4 describes how a GQM measurement program can be systematically improved and rolled out. Section 5 states the relation to existing work in the field. Section 6 gives some conclusions and future work.

2 GQM – An Appropriate Approach To Implement “Good” Measurements for KBSs and OMs

To understand why GQM is an appropriate approach, it is necessary to know its objectives and basics. The latter is subject of Section 2.1, the former of Section 2.2.

2.1 GQM: Objectives and Basics

As already stated, GQM is an industrial-strength technology for goal-oriented software engineering measurement (Basili et al., 1994b; Gresse et al., 1995; Rombach, 1990), which has been successfully applied in several companies. GQM helps to define and implement operational and measurable software improvement goals.

Motivations for goal-oriented measurement with GQM according to (Briand et al., 1996) are ensuring adequacy, consistency, and completeness of a measurement plan¹, dealing with the complexity of measurement programs, and stimulating a structured discussion about measurement. Additionally, GQM also helps systematically develop quality models and validate them in a context.

In GQM programs, the analysis task of measurement is specified precisely and explicitly by detailed measurement goals, called GQM goals, that reflect the business needs/goals. Relevant measures are derived in a top-down fashion based on the goals via a set of questions and quality/resource models. This refinement is precisely documented in a GQM plan, providing an explicit rationale for the selection of the underlying measures. The data collected is interpreted in a bottom-up fashion considering the limitations and assumptions underlying each measure. These principles are also depicted in Figure 1.

2.2 How Does GQM Ensure Good Measurements?

(Menzies, 1998c) and (van Harmelen, 1998), which is also related to (Menzies, 1998a), list some criteria that must be met by methods that implement good measurements for KBSs. Other widely accepted criteria are taken from the field of software measurements (Fenton, 1991; Basili et al., 1994b; Rombach, 1990; Briand et al., 1996). In the following, GQM is analyzed using

¹ A measurement plan defines how and when what data has to be collected and validated by whom.

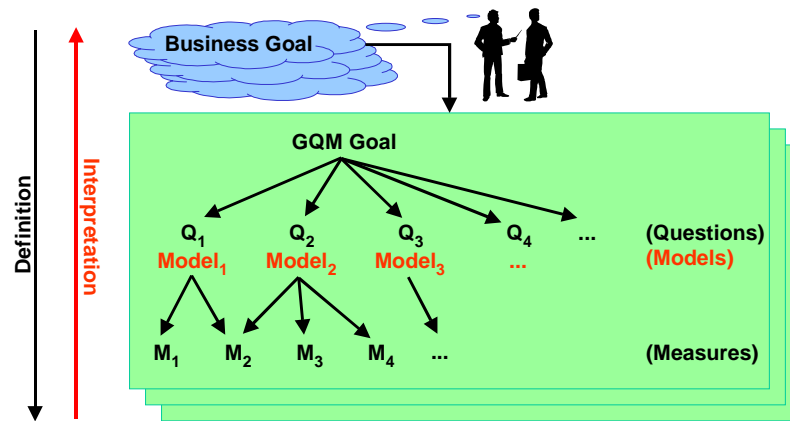


Figure 1: The basic principle of GQM.

these criteria. For each requirement it is explained how the state of the art in goal-oriented measurement with GQM addresses the requirement.

- Hypothesis requirement (Menzies, 1998c; van Harmelen, 1998)
To collect meaningful data, it is necessary to first state hypotheses that will be confirmed or refuted by the collected data. Two kinds of explicit quantitative hypothesis are stated for each goal:
 - 1 hypothesis about the expected quality (“baseline hypothesis”) and
 - 2 hypothesis about the expected impact of parameters (so-called *variation factors*) on the quality (“impact of variation factors”).
- Relation to business case (Menzies, 1998b)
To conduct meaningful, adequate, realistic measurements, it is necessary to relate the measurement program to the business case, that is, the business goals of the company as well as the actual usage of the system in practice.

For GQM, (Briand et al., 1996) distinguishes between improvement and measurement goals. The improvement goals state and reflect the business needs. The measurement goals are derived from the improvement goals. This aims at bringing goals, questions, and measures closer together as well as relating the measurement activities to the company’s business/improvement goals. Typically, measurement goals are changed more often than improvement goals.

For example, an improvement goal could be “improve the management of knowledge in company X”. The measurement goals would first deal with the characterization of the current status of knowledge management in the company to capture a baseline, then deal with the evaluation of the knowledge management in the company (i.e., assess the quality of knowledge manage-

ment), and finally deal with the control and change of the knowledge management activities (i.e., influencing knowledge management to alleviate risks and/or improve its quality).

- Measurement requirement (Menzies, 1998c; van Harmelen, 1998)
In the field of SE (Fenton, 1991) names three classes of entities and attributes to be subject to measurements: process (i.e., collection of software-related activities; e.g., duration of process, effort for several activities), product (i.e., any artifacts, deliverables, or documents that result from a process activity), resource (i.e., entities required by a process activity; e.g., personnel, materials, experience/skills) each requiring different categories of measures. In practice, it turned out that it is sufficient to distinguish only between measures for quality factors (that describe the quality of an object) and measures for variation factors (that are expected to have an impact on the quality factors) for all three classes (Briand et al., 1996). With GQM the related entity is given by the object of measurement stated in the GQM goal, which can be a product, process, or resource. Additionally, measures for data validation can be required in some environments to allow checking the validity of the collected data.

Quality models can be used as a “pool of ideas” or a kind of framework for developing measurement plans. Note that it is absolutely necessary to select the adequate parts of the model and/or select or derive adequate measures (Briand et al., 1996). Examples for quality models in the field of knowledge engineering are OMI’s cause-effect model for the factors that influence the usefulness as perceived by the user (see Figure 7; Althoff et al., 1999b), the quality criteria for the development of KBSs from (Benjamins et al., 1997), and the criteria for evaluating CBR systems from (Althoff, 1995).

GQM can also help to elicit such a quality model and validate it in the context of the measurement program. For example, OMI’s cause-effect model was in parts based on the experience gathered in the evaluation of CBR-PEB (see Section 3).

- Statistical requirement (Menzies, 1998c)
“Statistical theory requires several samples of a space before we can be confident that a representative portion of the sample space has been covered. Statistical validity is discussed extensively in (Cohen, 1995).”
(Menzies, 1998c)

GQM does not explicitly address the issues of statistical validity because its focus is on systematically developing an adequate, consistent, and complete measurement plan and stimulating and institutionalizing a structured discussion as well as feedback.

To improve the statistical validity in advance, (Briand et al., 1996) suggests to relate questions and measures closer to goals and to check the measures and data collections procedures against the underlying quality models before the actual data analysis takes place. This has the effect that the chosen measures and their scales can be better reviewed before effort is put into data collection, data analysis, interpretation, and a redefinition of the measurement program.

- Degradation studies (van Harmelen, 1998)
In a degradation experiment it is studied how algorithms, methods, ontologies, etc. behave when certain parameters are changed. To get meaningful results, the change of parameters should be plausible and not random.

GQM not only facilitates the systematic elicitation of the detailed measurement criteria regarding the quality but also the elicitation of parameters (*variation factors*) that influence the quality and, therefore, have to be considered when developing the measurement plan and planning experiments. Because the variation factors are elicited from experts regarding the context, they can be asked regarding meaningful and plausible variations of the variation factors.

In addition to these requirements stated by (Menzies, 1998b; Menzies, 1998c; van Harmelen, 1998), we identified the following criteria as vital for successful measurement programs.

- Explanation for deviations in repeated measurements
Variation factors provide explanations for deviations if measurements (i.e., experiment or case study) are repeated.

This also facilitates the roll-out of a case study to an experiment by helping to identify the variables that can be used to control the experiment. Obviously, these can be chosen from the variation factors.

- Clear and explicit documentation
The clear and explicit documentation of the measurement program, its rationale, and its related activities facilitates the adaption of a measurement program when necessary (e.g., because of changing environments). A well-documented measurement program can also be better understood by, for example, persons involved in a roll-out or reuse. Additionally, a good documentation improves communication among measurement program participants and users (Briand et al., 1996).

GQM helps and guides you to explicitly document all the important parts of the measurement program to make it well traceable (this is illustrated in Section 3): The goals are documented using templates. The rationale of the measurement program is documented by the decomposition of goals into ques-

tions and measures in the so-called GQM plan. The actual what, when, how, and who is documented in the measurement plan. The interpretations of the analyzed measurement data, results, and decisions are also documented. Lessons learned about measurement with GQM are also documented to help people in future measurement activities. A schema for describing lessons learned can be found in (Birk and Tautz, 1998).

- Consistency and completeness (Briand et al., 1996)
To collect only required and meaningful data, it must be ensured that the measurement plan is consistent and complete. The experience in the SE field shows that without these precautions it often happens that, for example, unneeded data¹ is collected or necessary data is not collected that would be required for the analysis and interpretation. The former makes the measurement program unnecessarily expensive, the latter would be a threat to the case study or experiment if it were detected too late.

The systematic development of the measurement plan according to the GQM method ensures its consistency and completeness. GQM has demonstrated this ability in various uses in the software industry, for example, see (CEMP Consortium, 1996).

- Identification of threats to validity
Variation factors support analyses regarding threats to validity such as a sample that is not representative (e.g., not addressing all required user groups of a system) or changes to system or measurement program.

Additionally, all this facilitates the assessment of an evaluation study regarding the following characteristics that are requested by (van Harmelen, 1998):

- Reproducibility: The reproducibility is supported by the good documentation as well as by the variation factors providing explanations for deviations in repeated measurements.
- Generalizability of the results: The elicitation and validation of quality models make the results generalizable. That is, the quality models can be used and validated in other contexts.
- Realism: The relation to the business case makes a study realistic.
- Well-controlled: The identification of variation factors helps controlling a study.

We have seen that the GQM methodology provides a way for conducting good measurements according to widely accepted requirements/criteria. Additionally,

¹ This is especially true for data that is expensive to collect. For example, data that is collected via a larger number of personal interviews.

the GQM methodology facilitates conducting a measurement program systematically, involving “users”, improving the measurement program itself, and learning about measurement activities. This is illustrated by the application of GQM to the existing system CBR-PEB in Section 3.

3 The Application of The GQM Technique to CBR-PEB

This section describes the application of our approach to the existing system CBR-PEB. First, the system CBR-PEB is introduced. Then it is shown how GQM has been applied to evaluate CBR-PEB.

3.1 The System CBR-PEB

CBR-PEB is an experience base that has been developed for supporting CBR system development (Althoff et al., 1999a). Emphasis is placed on providing decision support for reusing existing CBR system know-how for the development of new systems. To make the system easily accessible by CBR developers all around the world, the system has been made publicly available via the WWW (CBRPEB, 1998).

This experience base is based on a number of research efforts: Althoff et al. (Althoff, 1997) developed the classification criteria for CBR systems. Bartsch-Spörl, Althoff, and Meissonnier (Bartsch-Spörl et al., 1997) conducted a survey about CBR systems and made these experiences reusable by means of CBR technology, that is, each questionnaire has been represented as a structured case. Finally, an evaluation program was developed in order to show the usefulness of the system from the viewpoint of its users (Nick and Tautz, 1999).

3.2 The GQM Process for CBR-PEB

This section introduces GQM in detail and shows how GQM has been applied to evaluate the existing experience base CBR-PEB. General hints and information are marked with an arrow (\Rightarrow).

GQM can be applied in cycles. Each cycle refines the measurement program and – as a side effect – the EB system as well. Figure 2 shows the complete GQM cycle and its steps in general, and its instantiation for CBR-PEB (the first iteration). In spirit the GQM cycle is based on the Quality Improvement Paradigm (QIP), which is compatible with TQM (Basili, 1993). The single steps are described in detail in the following.

Prestudy. The first phase of a GQM program has the objective of collecting information relevant to the introduction of a GQM-based measurement program. This includes a description of the environment, “overall project goals”, and “task of the system”. This helps the person(s) responsible for the measure-

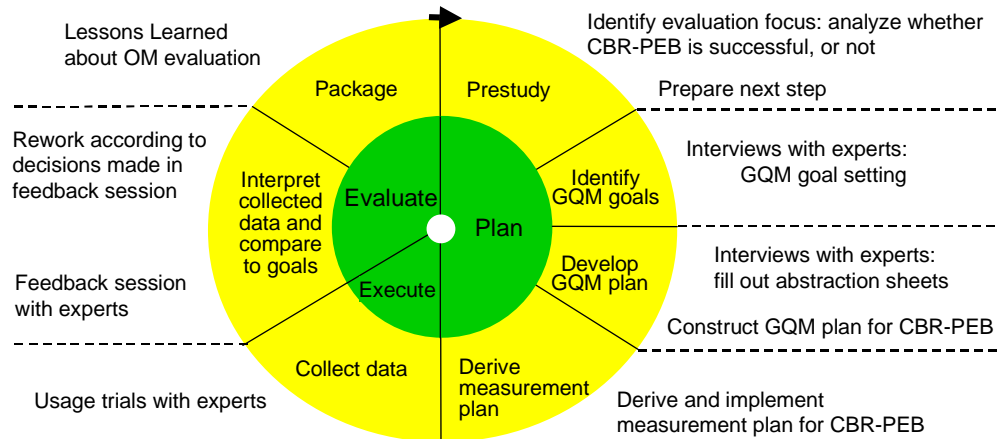


Figure 2: The standard GQM cycle and its instantiation for CBR-PEB.

ment program become familiar with the topic making it possible to appear in interviews as a competent person and partner. Usually, the participants of the measurement program are also trained in this phase.

- ⇒ A good source for information are also existing measurement programs and lessons learned recorded from these measurement programs as well as publications on general issues and models regarding evaluation and measurement (e.g., OMI's cause-effect model (see Figure 7; Althoff et al., 1999b), quality criteria from (Benjamins et al., 1997), evaluation criteria from (Althoff, 1995)). With this knowledge in mind, the interviewer can ensure that the goals, questions, and measures are really measurable and make sense. Without this knowledge it will take much more time to come up with a meaningful measurement program.

In the prestudy for CBR-PEB, a usage process model for CBR system development with CBR-PEB was developed to allow identification of definite points for measurement.¹ It was also proposed that the public installation in the WWW will lead to certain problems with the collection of measurement data, that is, it must be possible to distinguish between measurement data from surfers and from real use.

¹ Note that OMI's cause-effect model (Althoff et al., 1999b) was not available when the evaluation of CBR-PEB started. The quality criteria from (Benjamins et al., 1997) focused more on the development and did not conform with the chosen goals. However, the criteria described in (Althoff, 1995) and (Althoff, 1997) influenced the evaluation work.

Identify GQM Goals. The objective of identifying goals is to get a list of well-specified and ranked goals. First, informal goals are collected. Second, they are formalized according to the template for GQM goals. Third, the goals are ranked, and, fourth, the ones to be used in the measurement program are selected.

	Goal 1 "Technical Utility"	Goal 2 "Economic Utility"	Goal 3 "User Friendliness"
Analyze	retrieved information	retrieved information	organizational memory
for the purpose of	monitoring	monitoring	characterization
with respect to	technical utility	economic utility	user friendliness
from the viewpoint of	the CBR system developers	the CBR system developers	the CBR system developers
in the context of	decision support for CBR system development.	decision support for CBR system development.	decision support for CBR system development.

Figure 3: The list of formal, ranked, and selected GQM goals for CBR-PEB.

The resulting GQM goals (see Figure 3) focus on the technical and economic utility of the retrieved information (Goal 1 and 2) and on the user friendliness of the EB system (Goal 3). The viewpoint is represented by the interviewees and the context by environment and task of the system. Here, it is obvious that the other goals have the same viewpoint and context as Goal 1.

⇒ The identification of the GQM goals can be supported by existing work in the field such as OMI's cause-effect model (Althoff et al., 1999b), the quality criteria from (Benjamins et al., 1997), or the evaluation criteria from (Althoff, 1995) as well as more general models for evaluation such as the EFQM¹ model for business excellence (EFQM, 1997).

For example, if a company uses the EFQM model then actual GQM goals can be derived from the EFQM criteria. These EFQM criteria reflect the business goals that are the basis for the GQM goals. Lower level EFQM criteria can also support the identification and selection of GQM goals, questions, and measures. This is also supported by our experience from an ongoing project in the health care sector. For example, the business goal could be to improve customer satisfaction (EFQM criterion 6). Then a measurement goal could be derived from the lower level criterion 6a "assessment of the service in the hospital from the viewpoint of the customer": "Analyze the service for the purpose of evaluation with respect to reliability from the viewpoint of the customer in the context of the hospital X." To actually define "reliability of service", a GQM interview would be conducted to develop a GQM plan for this goal.

¹ EFQM = European Foundation for Quality Management

Develop GQM Plan. The objective of this step is to develop a GQM plan for each goal, that is, an operational refinement of a GQM goal via questions into measures including the analysis models that specify how the measurement data is analyzed to help answer the questions. This is done as a two-step process:

First, people representing the viewpoint according to the GQM goal are interviewed to make their implicit, relevant knowledge about the GQM goal explicit. For this purpose, *abstraction sheets* are filled out in an interview (e.g., see Figure 4 for the abstraction sheet of Goal 2 for CBR-PEB). An *abstraction sheet* represents the main issues and dependencies of a GQM goal in four quadrants: The “quality factors” describe the properties of the object in the goal to be measured, the “baseline hypothesis” describes the current knowledge with respect to the properties to be measured, the “variation factors” are factors that are expected to have an impact on the properties to be measured, the “variation factors” are factors that are expected to have an impact on the properties to be measured. This impact is described under “impact of variation factors”. Variation factors should only be listed if their impact is stated as well.

Goal: <i>Analyze the retrieved information for the purpose of monitoring economic utility from the viewpoint of the CBR system developers in the context of decision support for CBR system development.</i>		Names: M.M., N.N. Date: 97/10/01
Quality factors:	Variation factors:	
1. similarity of retrieved information as modeled in CBR-PEB (Q-12) 2. degree of maturity (desired: max.) [development, prototype, pilot use, daily use] (Q-13) [...]	1. amount of background knowledge a. number of attributes (Q-8.1.1) [...] 2. case origin [university, industrial research, industry] [...]	
Baseline hypothesis:	Impact of variation factors:	
1. M.M.: 0.2; N.N.: 0.5 (scale: 0..1) [...] The estimates are on average.	1. The higher the amount of background knowledge, the higher the similarity. (Q-8) 2. The more “industrial” the case origin, the higher the degree of maturity. (Q-9) [...]	

Figure 4: Abstraction sheet for Goal 2 “Economic Utility”. The numbers of the related questions in the GQM plan are included to improve traceability (here limited to questions addressed in the paper).

Second, for each goal a *GQM plan* is derived from the abstraction sheet. For each issue addressed in the quadrants “quality focus” and “impact of variation factors”, a top-level question is derived. For each of these top-level questions, a model (if available), a data presentation, and the hypothesis are also given. The hypothesis also includes a statement saying what the hypothesis from the abstraction sheet means with respect to the analysis results. The top-level questions are refined as necessary. For example, the question for the variation

hypothesis 2 for Goal 2 is refined into questions for the involved variation factor "case origin" and the involved quality factor "degree of maturity" (see Figure 5). So, the GQM plan documents all relevant and necessary information for the evaluation and, thus, shows a rationale for the measurement program.

- ⇒ Measures in GQM plans can be qualitative (e.g., quality of case source) and quantitative (e.g., completeness and similarity) as well as subjective (e.g., completeness estimated by the user) and objective (e.g., similarity automatically collected by the system).
- ⇒ According to our experience, it is useful to plan the data analysis as far as possible in advance (e.g., by defining (or using) an explicit model as shown in Figure 5). This makes it much easier to verify the measures, their scales, etc. before effort is put into data collection, analysis, etc. This is in line with the experience of (Briand et al., 1996).

Q-9 What is the impact of the case origin on the degree of maturity?

Q-9.1 What is the case origin?
M-9.1.1 per retrieval attempt: for each chosen case: case origin
[university, industrial research, industry]

Q-9.2 What is the degree of maturity of the system?
M-9.2.1 per retrieval attempt: for each chosen case: case attribute "status" ["prototype", "being developed", "pilot system", "application in practical use", "unknown"]

Model: Distribution of retrieval attempts by degree of maturity and case origin — see data presentation below.
The percentage of retrieval attempts per degree of maturity m (including "unknown") and case origin c is calculated as follows:

$$\text{att\%}_{\text{maturity}}(m, c) = \frac{\sum_{k=1}^{\text{\#retrieval attempts}} \frac{\text{\#chosen cases in attempt } k \text{ in } (m, c)}{\text{\#chosen cases in attempt } k}}{\text{\#retrieval attempts}}$$

A *chosen case* is a retrieved case that is regarded useful by the user.

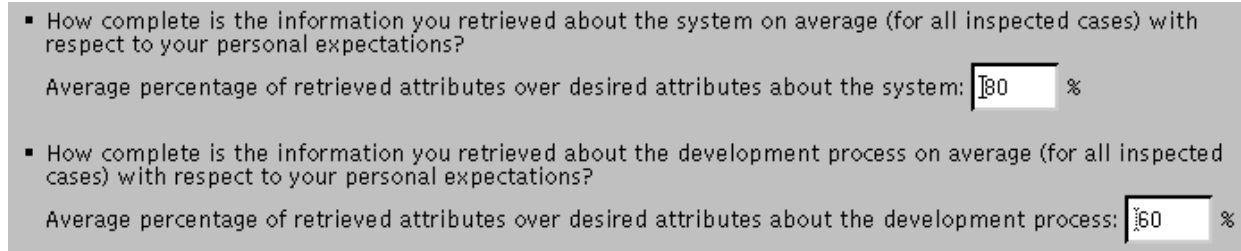
*Data presentation:*¹

degree of maturity	case origin			
	university	industrial research	industry	unknown
development	19.9%	5.8%		
prototype	19%	10.4%		
pilot use	2.2%	2.9%		
practical use	4.7%	2.3%	16%	
unknown	16.8%			

Hypothesis: The more "industrial" the case source ("industry" is more industrial than "industrial research" is more industrial than "university"), the higher the degree of maturity. That is, there should be a cumulation on the diagonal from (prototype, university) to (practical use, industry) in the data presentation below.

¹ These data reflect the status CBR-PEB from May 12, 1999.

Figure 5: Excerpt from GQM plan for Goal #2 "Economic Utility", Question 9. The data presentation contains the results from the usage trials for this question.



▪ How complete is the information you retrieved about the system on average (for all inspected cases) with respect to your personal expectations?

Average percentage of retrieved attributes over desired attributes about the system: %

▪ How complete is the information you retrieved about the development process on average (for all inspected cases) with respect to your personal expectations?

Average percentage of retrieved attributes over desired attributes about the development process: %

Figure 6: Excerpt from on-line questionnaire for Question 5 (completeness).

Derive Measurement Plan. The objective of this step is to implement the data collection. Thus, the GQM plans must be linked with the usage processes of the EB system. This is documented in the form of a *measurement plan* that describes for all measures from all GQM plans of the measurement program what measurement data is collected when, how, by whom, and who validates and stores the data. Finally, the data collection procedures are implemented, that is, questionnaires (paper-based or on-line – see Figure 6 for an example) and automatic data collection procedures are developed.

Data Collection. For the first iteration of the measurement program, only data from usage trials with the experts and some other persons were collected. This helped validate the data collection procedures and get first results (Nick and Tautz, 1999).

For the second iteration of the measurement program, data from real use is being collected (as is in this paper: see Figure 5 “Data Presentation”). This also includes data about queries derived from questions from the ai-cbr mailing list. The real-use data collected so far is currently being evaluated.

The collected data must also be validated. For example, with CBR-PEB it can happen that a user submits some measurement data, later notices that he has forgotten to enter some measurement data and submits the measurement data a second time (this is always possible in the WWW and difficult to avoid). Then the first data record is obviously invalid and has to be removed.

Interpret Collected Data. The collected and analyzed data is interpreted in *feedback sessions* with the experts (for exemplary data see Figure 5 “Data Presentation”). Thus, the objectives of these feedback sessions are the interpretation of the results of measurement data analyses, the verification of the hypotheses stated in the GQM plans (impact of variation factors and baseline), the comparison of the results to the goals, the evaluation of the measurement program, and the identification of the possibilities for improvement of both the software system and the measurement program [Gresse et al., 1995, p151]. The interpretations, results, and decisions are explicitly written down in the minutes for the feedback session.

In case of CBR-PEB, possibilities for improvement of both the measurement program and the retrieval process (i.e., the EB system) were identified. For the measurement program and the on-line questionnaires, for example, one should try to receive feedback from the Internet by means of a subjective rating of the whole system by the user and a text field for comments about the system. This feedback should be evaluated by the experience factory staff and during future feedback sessions. For the EB system, the experts suggested to improve the on-line help regarding certain terms which are not obvious.

Package. Some of the experience gained in the measurement program was packaged as lessons learned to make it explicit. These lessons learned can be used as guidelines for creating and maintaining a successful experience base. For example:

- High-quality artifacts are required and the information about these artifacts should be as up-to-date and as complete as possible.
- An effective retrieval mechanism is needed for identifying suitable (i.e., relevant) artifacts for reuse (e.g., with the 60 attributes of CBR-PEB, an exact match would be pure chance and very rare). This relevance must be modeled appropriately.
- The first iteration of the GQM process for CBR-PEB showed that the GQM approach is useful for evaluating an experience base and led to a meaningful result.
- The experience gathered in the evaluation of CBR-PEB was also compiled into parts of OMI's cause-effect model (Althoff et al., 1999b) – see Figure 7. For example, the factors in the branch "conceptual knowledge" address the impact of "attributes" and "concepts" on the "completeness" (this was stated in one of the GQM plans as impact of variation factors). Systematic Evolution of the GQM Program

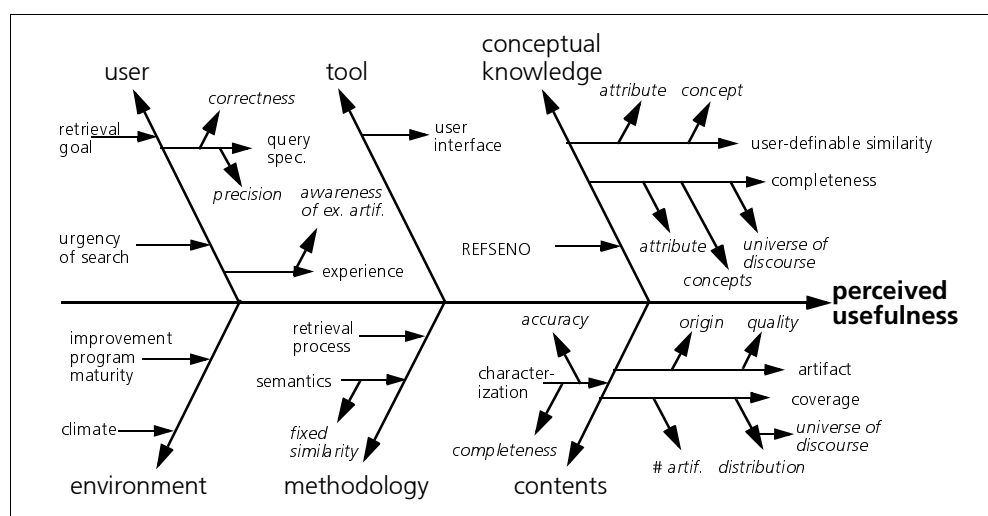


Figure 7:

OMI's cause-effect model: The usefulness as perceived by the user is influenced by many factors. (Althoff et al., 1999b)

4 Systematic Evolution of the GQM Program

GQM-based measurement programs are evolved systematically. The evolution should adhere the following three principles:

- 1 It is typical and a good strategy to start small with items that are well understood and easily measurable (Briand et al., 1996). This leads to a better understanding of domain, system, and measurement program. Based on the better understanding, the measurement program can be improved in each cycle. For example, the identification of problems with the data collection and analysis can lead to changes in the definition of the measurement program (e.g., to improve statistical validity you could try to better track additional variation factors and better track changes to variation factors).

This strategy also addresses the cost/benefit aspect of a measurement program. That is, in the beginning, it is important to demonstrate the benefits of measurement (i.e., measurement is useful to everybody) and minimize the risks and remain on the safe side.

- 2 The evaluation should guide the development and improvement of the system (Kirchhoff, 1994). This can be seen as a specialization of the guidance of AI research by evaluation as proposed by (Cohen, 1989) and (at least partially) carried out in the INRECA¹ project (Althoff, 1997; Althoff, 1995; Althoff et al., 1995).
- 3 The evaluation may not interfere with the evolution and improvement of the system. That is, regular changes to the system must be considered by the measurement program. It is not acceptable to delay, for example, updates of information just for the sake of measurement activities. This is also related to the second principle.

The systematic evolution takes place in two dimensions. First, for one domain the measurement program changes the focus with the maturity of the KB or OM system in this domain. Second, the measurement program must be extended when a new domain is added.² The latter mainly addresses KBSs or OM that

¹ INRECA = INduction and REasoning from CAses. ESPRIT III project No. 6322. May 1992 - November 1995.

² Note that you could also define a hierarchy of domains (e.g., CBR would be a super domain for CBR systems and CBR projects). For the following, it is sufficient to address just the implemented domain. Adding a new sub domain can be handled like adding a new domain.

are intended to cover several domains. All this leads to an evaluation/measurement program that addresses several domains in different phases.

For each step in the evolution a new iteration of the GQM cycle is started. Note that in a second and following iteration some steps can often be kept much shorter and simpler. For example it is not necessary to make a GQM interview for minor changes to a measurement program (especially if these changes had already been decided in a previous feedback session).

4.1 Evaluation for One Domain

The evaluation for one domain mainly bases on Principle 1 “start small and evolve continuously” and considers the current state of the art in KBS evaluation. Thus, the idea is to evolve continuously toward an ideal measure (economic viability). This evolution takes place in phases corresponding to the development of the system.

The Ideal Measure/Model. The ideal model would be an objective economic indicator such as return on investment. This would require to measure the effort or money saved by using the KBS for each usage of the system and compare this to the effort for setting up and running the KBS. It is obvious that it is very difficult to measure the effort or money saved because the measurement data will be meaningful only after the initial period in which the system changes frequently. Thus, measurement activities must start with items that are more stable and easier to measure.

Phases for Development and Evaluation. We distinguish three phases for the evaluation of a KBS (oriented at the maturity of the system): The first phase consists of the setting up of the KBS. This includes also the usage in the beginning where the system is changed quite often according to new/changed requirements and wishes of the users. The second phase is characterized by the regular use of the system. The third phase is characterized by the understanding and analysis of the economic viability of the system. These phases are not strictly separated, but rather overlapping.

In the first phase, the acceptance can be simply measured by the usage of the system (e.g., percentage/number of persons that used the system several times). Allowing textual feedback has also proven (in the evaluation of CBR-PEB) to be a good source of hints for the improvement of the system.

The focus on the acceptance is meaningful because the acceptance of a KBS by the intended users is crucial because such a system can only yield benefits if it is accepted and used.

Because the system tends to be changed relatively often in the beginning, it is very difficult to address the actual quality of the system regarding its contents, retrieval mechanisms etc. with a measurement program.

Applying GQM in this phase to the contents, retrieval mechanisms, etc. of the system will lead to a list of requirements rather than actual evaluation criteria including wishes for the criteria instead of hypotheses.¹ Thus, it would be an overkill to establish a measurement program in the beginning of a KBS project just for the sake of validating the system.

The second phase focuses on the guidance of the development by the evaluation (Principle 2). Ideally this should be conducted as a controlled experiment.

Thus, to measure the impact of changes to the system, changes must be well planned to obtain statistically valid measurement data. These changes are captured by the variation factors in the measurement program. Obviously, changes only make sense if there is at least one hypothesis regarding the impact of the change on the quality of the system.

On the other hand, evaluation must adhere to Principle 3 (do not disturb improvement). Thus, the evaluation must be able to deal with regular maintenance such as, for example, adding, modifying, or deleting cases in a CBR-based OM. This also must be reflected by respective variation factors.

The evaluation of our system CBR-PEB (see Section 3) is currently at the beginning of Phase 2. The experts know what they can expect from the system regarding its retrieval mechanisms and contents. The evaluation of the user friendliness was difficult because the WWW interface had not been finished when the GQM interviews were made.

The third phase is dominated by trying to measure actual costs and benefits to determine the economic value of the KBS (e.g., via its return on investment). Obviously, there must be a positive balance between costs and benefits. A detailed theoretical analysis of costs and benefits for experience bases can be found in (Nick, 1998). Only the usage of the system can yield the benefit.

On the side of the costs we must distinguish between several types of costs (Nick, 1998): the costs for setting up the system, the costs for running and

1 If GQM were used in such a way it would be more like knowledge acquisition than evaluation (i.e., GQM would be used as technique for structured interviews about requirements and the expected performance of the system). Note that such a GQM plan could also be derived from the requirements, and vice-versa. For example, the GQM plan for CBR-PEB for the goal "user friendliness" was more like a list of requirements regarding the maximum number of text input field, maximum number of questions asked by the system in the beginning of the query, etc. Such information is typically part of the requirements. Because our focus here is the evaluation we will not discuss this additional benefit of GQM in detail.

maintaining the system, and the costs for using the system. These costs can relatively easily be measured right from the start of KBS project (i.e., measure the effort that is needed to set up and maintain the system). This is necessary to get a complete picture of the costs and benefits of the KBS. The costs for using the system are required to compare them directly with the benefit established by the usage.

The actual benefit is much more difficult to measure. Theoretically this would have to be the subjective value of the information to the user (Cooper, 1997). Because this is practically and economically very difficult to measure, we have to use other measures. One idea could be to assign a kind of standard value to each package that is attributed to each usage of this package, but you have to be careful to make this meaningful rather than arbitrary.

In the end of one phase and in the beginning of the next phase, case studies and experiments help improving the awareness regarding the actual quality of the system. This was one of the major benefits of the first iteration of the evaluation program for CBR-PEB in which the experts recognized that the system was much better than expected (usage trials and feedback session). In another project we plan to conduct a field test to test the acceptance of an KBS by the users (regarding user interface and domain model)

Nice-To-Have Models. It would be very nice to have a model to estimate costs and benefits of the installation and usage of an KBS. This could be used as a very good argument for selling a KBS (if the figures are good).

At the moment, this is not necessary in most cases because KBSs and especially OMs require and still get strong support by the management in the beginning. Thus, measurement can first focus on acceptance and utility without addressing cost and benefit in terms of time/effort/money directly. In the future, we will have to be able to estimate the expected value of a KBS or OM in advance. This statement is supported by reports from the CBR community where buyers now begin asking for the economic benefit of the system to be installed.

4.2 Roll-Out for Several Domains

Although, in the case of CBR-PEB our approach has only been applied to one domain (CBR system development) and one class of users (CBR system developers), it can be easily rolled out to the general case with several domains (e.g., management processes, documents, experiences) and several classes of users (e.g., line and project managers) by defining GQM goals for each context and viewpoint according to domain and user class. Thus, the measurement program can be scaled according to a company's needs by refining domain and/or user class.

5 Related Work

In (Althoff, 1997), existing CBR systems were evaluated¹ to derive the requirements for the INRECA system (Althoff et al., 1995), which was the basis for several industrial CBR systems combining the benefits of existing CBR systems. Although the goal of this study was a different one, the context was the same as for CBR-PEB, and the procedure and the kind of expected results were similar to the GQM process: The evaluation criteria were categorized and refined. The existing CBR systems were analyzed according to the evaluation criteria to derive the requirements for the INRECA system. Several iterations were necessary to get the final results.

In (Kirchhoff, 1994), it was proposed to use evaluation to guide the incremental development of knowledge-based systems. An experimental methodology for this guidance was developed. It was stated that people (knowledge engineers and users) need to be involved in development and evaluation. This is similar to the involvement of experts in GQM interviews and feedback sessions.

In (Menzies, 1998b), the need for business-level evaluations was underlined. An approach using critical success metrics (CSMs) for such evaluations was presented. CSMs indicate success if some number inferred from the system passes some value. This approach was applied to a knowledge-based system used in the petrochemical industry. In contrast to GQM, the approach does not allow subjective measures, which are typical for more complex applications like experience bases, and allows only small changes to a running system.

Within OMI (Althoff et al., 1999b) GQM can be used to determine the utility/usefulness criteria, that is, GQM can support the identification of GQM goals as well as the identification and selection of quality and variation factors (depending on the aspects of the cause-effect model addressed by the measurement program).

¹ according to (Menzies and van Hamelen, 1999) it can be considered as a (very detailed) small evaluation.

6 Summary and Conclusion

Most of the existing evaluation approaches for knowledge-based systems (KBSs) in general and organizational memories (OMs) in particular are difficult to adapt to company-specific needs (Menzies, 1998b). To be useful in practice, evaluation techniques need to be able to cope with various environments, viewpoints, and measurement objectives as well as with the incremental build-up of the KB or OM system. A detailed evaluation can also guide the development of an OM. Our approach uses the Goal-Question-Metric (GQM) technique to evaluate an OM for software engineering knowledge. GQM is an industrial-strength technique for software engineering measurements, especially designed to deal with practical needs. It does so by involving the data collectors (in this case the users of the OM), as they are considered to be the experts in developing and/or applying the measured objects.

The paper focused on three main issues: how GQM facilitates evaluation, an exemplary case study (i.e., the successful application of GQM to an experience base on CBR systems), and how GQM can be iterated to guide and support the systematic evolution and improvement of the evaluation program and the system.

The GQM methodology facilitates the evaluation of KBSs and OMs by explicitly including requirements for good measurements in the field of KBSs and OMs (Menzies, 1998b; Menzies, 1998c; van Harmelen, 1998) as well as software engineering (Fenton, 1991; Briand et al., 1996) into the evaluation method, for example, the relation to the business case, explicit hypotheses, and support for experiments and case studies. The systematic conducting and documentation of the measurement activities, rationales, and results (Gresse et al., 1995) makes measurement programs well-traceable and repeatable.

The systematic evolution and improvement of the measurement program and the system is based on principles for measurement, evaluation, and system development. The systematic evolution addresses the phases in the development (i.e., the maturity of the system and evaluation program) in a single domain as well as the roll-out for several domains.

In our experience, the maturity of the evaluation program progresses through three phases for KBSs and OMs: (a) prototypical use, (b) use on a regular basis, and (c) wide-spread use. Each phase is associated with typical measurements. For instance, during prototypical use (where the system is still undergoing frequent changes), acceptance measured in terms of system usage (e.g., number of accesses per user within a certain time span) and informal user feedback are

of primary importance. Once the system is in regular use, the focus of the evaluation program shifts to guidance for improvement of the system based on more formal user feedback. Finally, cost/benefit calculations are the primary focus during the phase of wide-spread use. This experience is in line with the progression of the purpose of measurement goals in general: characterization/monitoring, evaluation, and control/change as reported in (Briand et al., 1996).

Currently, we are evaluating the data collected from the real use of CBR-PEB. In the future, we plan to apply GQM to the evaluation of organizational memories in the health care sector in a research project funded by the Fraunhofer Society (Munich, Germany) as well as in other public projects currently being proposed. For future research, we also see the development of quality models for KBSs as an important point.

References

- Abecker, A., Decker, S., and Kühn, O. (1998). Organizational memory (in German). *Informatik-Spektrum*, 21(4):213–214.
- Althoff, K.-D. (1995). Evaluating case-based reasoning systems. In *Proceedings of the Workshop on Case-Based Reasoning: A New Force in Advanced Systems Development*, pages 48–61, Unicom Seminars Ltd., Brunel Science Park, Cleveland Road, Uxbridge, Middlesex UB8 3P, UK.
- Althoff, K.-D. (1997). Evaluating case-based reasoning systems: The Inreca case study. Postdoctoral thesis (Habilitationsschrift), University of Kaiserslautern.
- Althoff, K.-D. and Aamodt, A. (1996). Relating case-based problem solving and learning methods to task and domain characteristics: Towards an analytic framework. *AICom - Artificial Intelligence Communications*, 9(3):109–116.
- Althoff, K.-D., Auriol, E., Bergmann, R., Breen, S., Dittrich, S., Johnston, R., Manago, M., Traphöner, R., and Wess, S. (1995). Case-based reasoning for decision support and diagnostic problem solving: The INRECA approach. In *Proceedings of the Third German Conference on Knowledge-Based Systems (XPS-95)*.
- Althoff, K.-D. and Bartsch-Spörl, B. (1996). Decision support for case-based applications. *Wirtschaftsinformatik*, 38(1):8–16.
- Althoff, K.-D., Nick, M., and Tautz, C. (1998). Concepts for reuse in the experience factory and their implementation for CBR system development. In *Proceedings of the Eleventh German Workshop on Machine Learning (FGML-98)*. <http://demolab.iese.fhg.de:8080/Publications/fgml98/>.
- Althoff, K.-D., Nick, M., and Tautz, C. (1999a). CBR-PEB: An application implementing reuse concepts of the experience factory for the transfer of cbr system know-how. In *Proceedings of the Seventh Workshop on Case-Based Reasoning during Expert Systems '99 (XPS-99)*, Würzburg, Germany.
- Althoff, K.-D., Nick, M., and Tautz, C. (1999b). Improving organizational memories through user feedback. In *Workshop on Learning Software*

Organisations at SEKE'99, Kaiserslautern, Germany.

- Bartsch-Spörl, B., Althoff, K.-D., and Meissonnier, A. (1997). Learning from and reasoning about case-based reasoning systems. In *Proceedings of the Fourth German Conference on Knowledge-Based Systems (XPS97)*.
- Basili, V. R. (1992). The experimental paradigm in software engineering. In Rombach, H. D., Basili, V. R., and Selby, R. W., editors, *Experimental Software Engineering Issues: A critical assessment and future directions*, pages 3–12. Lecture Notes in Computer Science Nr. 706, Springer-Verlag.
- Basili, V. R. (1993). The Experience Factory and its relationship to other improvement paradigms. In Sommerville, I. and Paul, M., editors, *Proceedings of the Fourth European Software Engineering Conference*, pages 68–83. Lecture Notes in Computer Science Nr. 717, Springer-Verlag.
- Basili, V. R., Caldiera, G., and Rombach, H. D. (1994a). Experience Factory. In Marciniak, J. J., editor, *Encyclopedia of Software Engineering*, volume 1, pages 469–476. John Wiley & Sons.
- Basili, V. R., Caldiera, G., and Rombach, H. D. (1994b). Goal Question Metric Paradigm. In Marciniak, J. J., editor, *Encyclopedia of Software Engineering*, volume 1, pages 528–532. John Wiley & Sons.
- Benjamins, V. R., Fensel, D., Pierret-Golbreich, C., Motta, E., Studer, R., Wielinga, B., and Rousset, M.-C. (1997). Making knowledge engineering technology work. In *Proceedings of the Ninth Conference on Software Engineering and Knowledge Engineering*, pages 18–20, Madrid, Spain.
- Bergmann, R. and Althoff, K.-D. (1998). Methodology for building case-based reasoning applications. In Lenz, M., Bartsch-Spörl, B., Burkhard, H.-D., and Wess, S., editors, *Case-Based Reasoning Technology - From Foundations to Applications*, number 1400 in LNAI, pages 299–328. Springer-Verlag, Berlin, Germany.
- Bergmann, R., Breen, S., Göker, M., Manago, M., and Wess, S. (1999). *Developing Industrial Case-Based Reasoning Applications – The INRECA Methodology*. Springer Verlag.
- Birk, A. and Tautz, C. (1998). Knowledge management of software engineering lessons learned. In *Proceedings of the Tenth Conference on Software Engineering and Knowledge Engineering*, San Francisco Bay,

CA, USA. Knowledge Systems Institute, Skokie, Illinois, USA.

- Briand, L. C. and Differding, C. M. (1996). Practical guidelines for measurement-based process improvement. Technical Report ISERN 96-05, Fraunhofer Institute for Experimental Software Engineering, Sauerwiesen 6, 67661 Kaiserslautern, Germany.
- Briand, L. C., Differding, C. M., and Rombach, H. D. (1996). Practical guidelines for measurement-based process improvement. *Software Process*, 2(4):253–280.
- CBRPEB (1998). Cased-Based Reasoning Product Experience Base CBR-PEB. <http://demolab.iese.fhg.de:8080/>.
- CEMP Consortium, T. (1996). Customized establishment of measurement programs. Final report, ESSI Project Nr. 10358, Germany.
- Cohen, P. R. (1989). Evaluation and case-based reasoning. In Hammond, K., editor, *Proceedings of the Second DARPA Workshop on Case-Based Reasoning*, pages 168–172. Morgan Kaufman.
- Cohen, P. R. (1995). *Empirical Methods for Artificial Intelligence*. MIT Press, 55 Hayward Street, Cambridge, MA 02142.
- Cooper, W. S. (1997). On selecting a measure of retrieval effectiveness. In Jones, K. and Willet, P., editors, *Readings in Information Retrieval*, pages 191–204. Morgan Kaufmann Publishers.
- EFQM (1997). *The European Quality Award (in German)*. European Foundation for Quality Management, Brussels.
- Fenton, N. E. (1991). *Software Metrics: A Rigorous Approach*. Chapman & Hall, London.
- Gresse, C., Hoisl, B., and Wüst, J. (1995). A process model for GQM-based measurement. Technical Report STTI-95-04-E, Software Technologie Transfer Initiative Kaiserslautern, Fachbereich Informatik, Universität Kaiserslautern, D-67653 Kaiserslautern.
- Kirchhoff, S. (1994). *Abbildungsqualität von wissensbasierten Systemen: eine Methodologie zur Evaluierung*. Verlag Josef Eul, Bergisch Gladbach, Germany.
- Menzies, T. (1998a). Evaluation issues for problem solving methods. In *Proceedings of the Eleventh Knowledge Acquisition for Knowledge-Based Systems Workshop*.

- Menzies, T. (1998b). Evaluation issues with critical success metrics. In *Proceedings of the Eleventh Knowledge Acquisition for Knowledge-Based Systems Workshop*.
- Menzies, T. (1998c). Requirements for good measurements. <http://www.cse.unsw.edu.au/timm/pub/eval/cautions.html>.
- Menzies, T. and van Hamelen, F. (1999). *The Second Banff KAW'99 Track on: Evaluation of KE Methods*. <http://www.cse.wvu.edu/timm/banff99/>.
- Nick, M. (1998). Implementation and Evaluation of an Experience Base. Diploma thesis, Fraunhofer IESE, University of Kaiserslautern.
- Nick, M. and Tautz, C. (1999). Practical evaluation of an organizational memory using the goal-question-metric technique. In *XPS'99: Knowledge-Based Systems - Survey and Future Directions*. Springer Verlag, Würzburg, Germany. LNAI Nr. 1570.
- Prahalad, C. K. and Hamel, G. (1990). The core competence of the corporation. *Harvard Business Review*, 68(3):79–91.
- Rombach, H. D. (1990). Practical benefits of goal-oriented measurement. In *Proceedings of the Annual Workshop of the Centre for Software Reliability*, pages 217–235. Elsevier.
- Rombach, H. D. (1991). Practical benefits of goal-oriented measurement. In Fenton, N. and Littlewood, B., editors, *Software Reliability and Metrics*, pages 217–235. Elsevier Applied Science, London.
- van Harmelen, F. (1998). Notes from evaluation working group at Banff'98. <http://www.cs.vu.nl/frankh/spool/eval.html>.

Document Information

Title:	Facilitating the Practical Evaluation of Knowledge- Based Systems and Orga- nizational Memories Using the Goal-Question- Metric Technique
Date:	May 31, 1999
Report:	IESE-029.99/E
Status:	Final
Distribution:	Public

Copyright 1999, Fraunhofer IESE.
All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means including, without limitation, photocopying, recording, or otherwise, without the prior written permission of the publisher. Written permission is not needed if this publication is distributed for non-commercial purposes.