Acquiring Knowledge for Linking Software Engineering Experience Maintenance with Evaluation

Markus Nick, Klaus-Dieter Althoff, Andreas Jedlitschka

Fraunhofer Institute for Experimental Software Engineering (IESE)
Sauerwiesen 6, D-67661 Kaiserslautern, Germany
Tel: +49 (0) 631-707 233
[nick | althoff | jedl]@iese.fraunhofer.de

Abstract. The value of a software engineering experience repository like for any other experience-based information systems tends to degrade with time. To keep the value of such a system, evaluation and maintenance is an essential. While evaluation monitors the "value" over time, maintenance has to preserve or improve this "value". Evaluation and maintenance should not simply happen ad-hoc but systematically and based on specific quality and maintenance knowledge, which is linked to base maintenance on evaluation. As a jump-start for evaluation and maintenance, the respective knowledge should be available right from the start of continuous operation. This paper describes how to acquire and develop such maintenance knowledge during system buildup to use and improve this knowledge during continuous operation for combined human- and computer-based maintenance. The described approach has been successfully applied in several projects. The method is illustrated with examples from two of these projects as case studies.

1 Introduction

The value of software engineering (SE) knowledge and experience tends to degrade with time, be it by external impacts on the organization's environment or by changes within an organization (e.g., the development of a new product). Therefore, also the value of the system/repository for storing and providing the SE knowledge and experience tends to degrade by time. This is particularly true for experience in the sense of , which case-specific knowledge[1], because such knowledge is gained almost continuously in daily work [28]. The new field of "Experience Management" (EM) [27, 7, 3] deals with all the relevant research and development issues of *experience based information systems* (EBIS) as well as with their integration into business processes. SE experience repositories are simply EBISs focusing on SE experience. Case-based reasoning (CBR) systems are the most prominent instantiations of EBIS, because the CBR community has been contributing real-life EBIS for more than ten years. Other examples of EBIS

[1] In the scope of this paper, the terms *experience*, *case*, and *case-specific knowledge* are considered as synonymous. *Lessons learned* and *best practices* are considered as specific kinds of experiences.

are experience databases, best practice repositories, or lessons learned systems. Thus, the ingredients of EM come from CBR as well as scientific fields like experience factory (EF) [6], knowledge management, and information systems.

In EM, maintenance is of particular importance, because a (more or less) continuous stream of experience has to be processed [8, 21]. For example, for a group of about 60 affected researchers at IESE, our CBR-based EBIS on project experience had an annual growth of about 500 lessons learned. In addition, our EBIS also includes best practice descriptions on business processes and information on projects as well as the links among these different knowledge and information types. Because this EBIS should be maintained with low effort and the EF staff as maintenance team can work only part-time for the EF/EBIS, tool support is highly regarded [21]. High quality of the retrieved knowledge is also requested by the users of such systems. The respective quality criteria should be related to organizational goals [27, 22].

All this demonstrates that maintenance has a certain complexity for such systems and is a knowledge-intensive task. Thus, guidance and decision support for maintenance is almost essential to successfully maintain and improve such a system. Due to the variety and amount of knowledge in an EBIS, authoring and maintenance support has to combine human- and computer-based maintenance activities. In [21], we presented the EMSIG framework and an integrated technical solution that operationalizes the support for maintenance regarding cases and conceptual model using specific maintenance knowledge. As a jump-start for continuous operation, this maintenance knowledge should be available at the end of the buildup, that is, when the initial acquisition and development of the system and of the knowledge in the "standard" containers (vocabulary, cases, similarity measures, adaptation [25]) has been finished. Manually performed maintenance processes (e.g., acquisition of new cases) are described at a coarse-grained level in methodologies such as INRECA [8] or DISER [27]. Automatic or tool-supported maintenance procedures are available, for example, from CBR research for very specific knowledge types for certain task and domain types [14, 15].

However, the maintenance knowledge for decision support and specific maintenance tasks is rather acquired "by chance" during continuous operation (so far). Thus, it might take long to learn the required maintenance knowledge for decision support. The problem is that existing methods such as INRECA or DISER only fill the "standard" knowledge containers of CBR systems. [26] provides a maintenance manual for a limited number of rather generic tasks that do not consider system-specific issues. Some of the tasks from [26] can be triggered based on measures for the case base from [24]. Therefore, there is still a lack of support for system-specific evaluation and maintenance, which is addressed by the work presented in this paper.

We have developed the EMSIG approach for maintenance and evaluation of EBIS/CBR systems, which extends DISER with maintenance. EMSIG combines human- and computer-based maintenance activities and respective decision support. Maintenance and evaluation support tools are implemented in an integrated system using CBR technology [21]. The EMSIG-KA method [17, 18] addresses in particular the development of maintenance and evaluation knowledge before going into regular use as well as the further usage and development of this knowledge during regular use. However, the method as described in [17, 18] still lacks the link between evaluation and maintenance

in the development of respective knowledge before regular use. This missing link between evaluation and maintenance is the focus of this paper.

The paper is structured as follows. In Section 2, we introduce the different maintenance knowledge types. Section 3 describes the part of the EMSIG-KA method for acquiring and using quality knowledge for the evaluation of the system during regular use. In Section 4, we give an overview on the how to acquire and derive maintenance decision knowledge. The focus is on the knowledge for the missing link between evaluation and maintenance. The resulting knowledge can be used with tools such as EMSIG's software components. Section 5 gives an overview on projects and two case studies for the described method. Finally, some conclusions are drawn.

2 Overview on CBR Maintenance Knowledge Types

Richter [25] describes four basic knowledge containers for a CBR system: Vocabulary, similarity, cases, and adaptation knowledge. For the purpose of integrating evaluation and maintenance into a CBR system, we add three new knowledge containers that are described in this Section.

2.1 Quality Knowledge

Quality knowledge describes how the quality of the EBIS is measured and the current status of the system with respect to quality as well as the rationale for the definition of quality [16]. Quality knowledge deals with quality aspects of the EBIS as a whole, that is, the EBIS contents and conceptual model as well as retrieval mechanisms, usability of the user interface, etc. An example for content-related quality knowledge is a definition of measures for the utility or value of single cases (see Section 3.2). There are several types of quality knowledge, which are related as follows: The measures define what data is collected. The data collection is performed automatically or manually by respective data collection procedures. The collected data is analyzed using predefined models or procedures. The results of the analyses can be used for justifying an EB and as input for decisions about maintenance [21, 22].

2.2 Maintenance Process/Procedure Knowledge

Maintenance process and procedure knowledge defines how the actual maintenance activities are performed. The actual maintenance can be performed as a mix of automatically and manually performed activities. For the *automatically* performed activities (*maintenance procedures*), tool support by components of the EBIS or separate tools is required. The remaining activities have to be performed *manually* (*maintenance processes*). To improve guidance for the maintainers, descriptions of these processes are provided (e.g., detailed description of the acquisition of new cases through collecting cases, reviewing these cases, and publishing them in the case base - see DISER [27] and INRECA methodology [8] for examples). To combine manual and automatic maintenance, a maintenance process can have automated subprocesses/steps, which use input from or provide input for manually performed steps.

2.3 Maintenance Decision Knowledge

Maintenance Decision Knowledge links the quality knowledge with the maintenance process knowledge. It describes under what circumstances maintenance processes/procedures should be executed or checked for execution. Such maintenance knowledge can be described in an informal manner as *maintenance policies* [15], which define when, why, and how maintenance is performed for an EBIS. The "why" addresses not only the reason of maintenance but also the expected benefits of the maintenance operation, which should be related to the objectives of the EBIS or to the general goal of maintenance (i.e., to preserve and improve the EBIS value [21]). Since these objectives are typically very high-level, it is not very meaningful to address the EBIS objectives directly. Instead, we use a refinement of the objectives: the quality criteria from the evaluation program or the recording methods. The "how" is a combination of maintenance processes and procedures with additional steps as "glue."

3 GQM-Based Evaluation of EBIS/CBR Systems

The success of an EBIS can be measured in many ways. [4] contains overviews and references. There are examples for specific views on evaluation mainly from the knowledge-based system field and related fields. Furthermore, some evaluation work has been done in software engineering (SE) in the area of software reuse (programs), mainly regarding economic success. Many of the economic models for software reuse can also be used for evaluating EBIS. Other evaluation criteria, most importantly *recall* and *precision*, come from library and information science. Cooper [11] proposes to measure the success of an information system by the »personal utility« of the delivered information to the user. However, as pointed out by Cooper, the ideal measurement of the usefulness as perceived by the user is practically and economically impossible.

Our evaluation methodology is based on the Goal-Question-Metric (GQM) paradigm [6] for goal-oriented measurement. It includes the process, templates, and guidelines for the application of GQM [9]. GQM is an industrial-strength technique that has been successfully used in the field of SE at, for example, NASA-SEL, Robert Bosch GmbH, Allianz Lebensversicherungs AG, Digital SPA, Schlumberger RPS [10].

In this section, we give a short overview on the parts of the method that are relevant for CBR-PEB and ESERNET: The principle of the GQM method (Section 3.1), the 3-phase quality model showing how evaluation goals, criteria, and metrics for EBIS evolve over time (Section 3.2), and the feedback channels (Section 3.3).

A detailed, comprehensive description of the method can be found in [3]. Parts of the method are described in [22, 4].

3.1 GQM: Objectives and Basics

The Goal-Question-Metric (GQM) method systematically facilitates planning and conducting measurements by helping to define and implement operational and measurable improvement goals [6, 9]. In [20], we showed that GQM also meets the requirements for 'good measurements' in knowledge engineering and, thus, is suitable for evaluating a knowledge-based system like a CBR system.

Motivations for goal-oriented measurement with GQM –according to [9]– are to ensure adequacy, consistency, and completeness of the measurement program, deal with the complexity of measurement programs, and stimulate a structured discussion about measurement with the relevant stakeholders. Additionally, GQM also helps to systematically develop quality models and validate them in a given context. GQM has no restriction regarding types of metrics to be used: Qualitative as well as quantitative metrics, metrics for products, processes, and resources can be combined.

The principle of GQM programs is as follows (Fig. 1): 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 or quality criteria and quality/resource models, providing an explicit rationale for the selection of the underlying measures. This refinement is precisely documented in (1) a GQM abstraction sheet stating for each goal the quality factors and the variation factors that impact the quality factors and the respective hypotheses (e.g., see Fig. 4, Fig. 9), (2) a GQM plan stating for each goal the questions derived from the quality factors and variation factors, measures, and analysis models. The data collected is interpreted in a bottom-up fashion considering the limitations and assumptions underlying each measure.

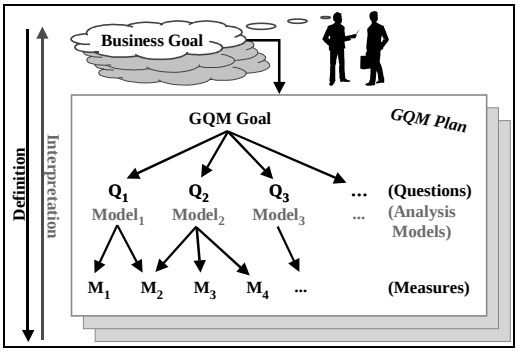


Fig. 1. The basic principle of GQM.

3.2 3-Phase Quality Model for Meaningful Goals & Metrics over Time

Depending on the maturity level of the EBIS, the measurement program has to be adapted. Hence, the quality models used for evaluating a single EBIS have to change over time. Fig. 2 illustrates these changes. In our model, we distinguish the three phases for the evaluation of an EBIS [22]: In the beginning of the usage of the system (i.e., Phase 1), we use our standard model for measuring indicators about the acceptance of the system [21, 12]. We combine measuring the usage of the system (e.g., number of queries) and feedback on the utility of the retrieved experiences. Combining usage and

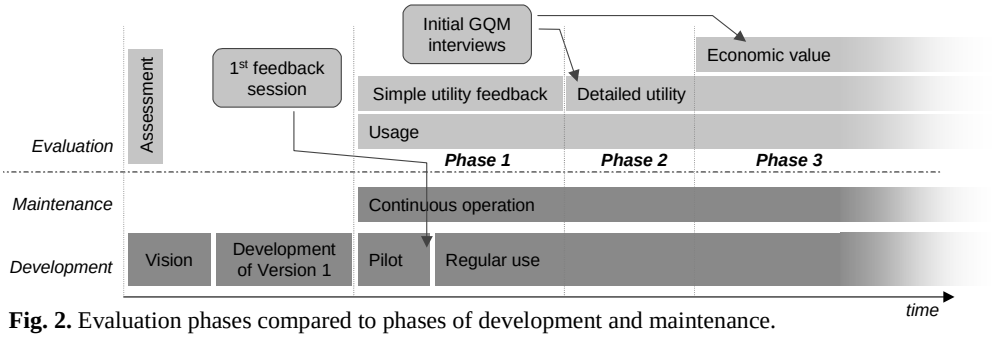


Fig. 2. Evaluation phases compared to phases of development and maintenance.

utility allows obtaining a picture on the acceptance more quickly than just monitoring usage because -in the beginning- usage can also be high because the system is new and everybody plays with it. Furthermore, the utility feedback helps to better understand what the users are really interested in. Furthermore, the coverage is measured by monitoring the number of cases in the EBIS over time. This is compared to the expected development of the number of cases. Later, in Phase 2, application-specific issues can be added for a more detailed evaluation. Phase 3 focuses on the economic value of the system.

3.3 Feedback Channels

The feedback channels allow the identification of usefulness problems based on user feedback. They explain causes for a selected number of factors from a cause-effect model [4] and the respective effects/problems that are experienced by the user (see Fig. 3 for an example). These effects are described using indicators. For each indicator, one or several causes are stated. From the causes, the respective measures for evaluation are derived. For each cause, a respective improvement action is stated. An improvement action can be implemented (a) as a recommendation from the user interface to the user or (b) as a maintenance policy.

Step	Indicator	Cause	Improvement action	Type of action
evaluate/ examine	Not enough useful artifacts in the result of the query	Imprecise or incorrect query	Re-specify query	direct
		Coverage too low	Record new artifacts	cumulative
		User overestimates EBIS	Clearly express purpose, focus, and abilities in user interface. Communicate these to users.	cumulative

Fig. 3. Excerpt from list of feedback channels [3]. This feedback channel has several causes.

For GQM, the cause-indicator pairs provide answers to questions about the indicated problems: If they occur, how often, and under which circumstances.

To select feedback channels, the users' viewpoint is taken or representatives of the users are interviewed. An interview has the advantage that it can be clarified which feedback the users are willing to give. A measurement specialist can now tell what in addition has to be measured to allow a meaningful analysis of the data to be collected.

Feedback channels are set up per retrieval goal or query interface. Details on the feedback channels can be found in [3, 4].

4 Defining Maintenance Decision Knowledge

Our method EMSIG-KA implements the principle of acquiring and developing maintenance decision knowledge while developing the EBIS. It derives operational maintenance knowledge from three major sources: (1) a knowledge life-cycle model (see [17]), (2) artifacts developed and information gained during buildup (see [18]), and (3) the parts of the quality knowledge defining what, when, and how to measure. From

these sources, rather informal maintenance policies are derived. In addition, generic, well-tested maintenance policies from CBR research and practice can be reused for general aspects (see [15] for an overview and [14] for the state of the art). Before using these, their application constraints must be checked and analyzed carefully.

The structure of a *maintenance policy* is as follows: A "trigger" states the condition for "firing" the policy. The "actions" describe what has to be done when a guideline is triggered, that is, how which knowledge/experience item has to be maintained. The "expected benefits" help justify the application of the policy, for example, by cost-benefit issues or quality improvements.

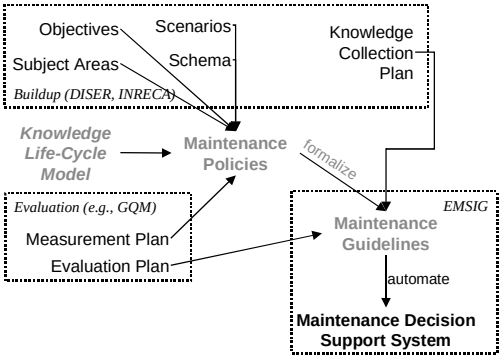


Fig. 4. Deriving maintenance decision knowledge from different sources.

In the following, we focus on the measurement program as a major source and on how to derive maintenance policies from it. For details on how to derive maintenance policies from the other sources please refer to [17, 18].

4.1 Deriving Informal Maintenance Decision Knowledge from the Evaluation Program

In general, the triggers can be derived from the evaluation program because this is the link from the maintenance decision knowledge to the quality knowledge, which is described by the evaluation program. For the trigger identification, the quality factors and variation factors from the abstraction sheets are analyzed: (see Fig. 5, Fig. 9)

- The quality factors together with the respective baseline define expectations regarding the quality. If there is no well-founded baseline available in the beginning, the first policy has to be defined with the expectation/estimation. After some iterations, the policy is revised to consider the well-founded baseline. The maintenance activity is to check if the expectations are met sufficiently.
- The variation factors and respective analysis model results can be used in three different ways: (1) They can be used directly in a trigger. (2) They can "adjust" a trigger by, for example, choosing respective values from the baseline. (3) They can be used as background information for human-made decisions if no explicit decision model has been developed yet.

After identifying the triggers, the respective actions are outlined. In a second step, the actual maintenance policies are "compiled", selected, and refined. In particular, the actions are defined, discussed, and reviewed by the maintenance personnel (see Fig. 10).

To document the relation between measurement program and maintenance policies & guidelines, the IDs of the referred questions/criteria/measures/models are stated in the maintenance policies & guidelines.

4.2 Operationalizing Information Maintenance Decision Knowledge for Tool Support

To allow tool support, the maintenance policies are further formalized as *maintenance guidelines*. The maintenance guidelines have the same basic structure as the maintenance policies. To allow tool support, the following major changes and extensions are made with respect to maintenance policies (see [21] for details):

A partial or complete formalization of the "trigger" is required to allow an automatic tool-based checking of the trigger. The formalized parts of the trigger can refer to items from the standard containers as well as measures, user feedback, periodic events, or events such as the end of a project. For more advanced analyses, the trigger can also refer to results from analysis tools such as data or text mining tools. The parts of the trigger that cannot be formalized are included in the maintenance guideline for manual checks by the responsible role. In case the actual trigger cannot be formalized at all, then the respective guideline can be triggered periodically as a reminder and an EF staff member does the actual condition check manually.

The "actions" refer to human-based maintenance processes, automatic maintenance procedures, and the "glue" among these. Thus, it is possible to combine automatic procedures with human-based execution of processes. Looseness or tightness of this integration depends on the tools, e.g., one could export cases for analysis with a data-mining tool each month (loose integration) or a data-mining component could be run automatically (tight integration).

5 Projects and Case Studies

Work on the approach was started in 1997 with the evaluation of CBR-PEB (Section 5.1), which laid the foundation for the tailoring and extension of the GQM approach for EBIS [23, 20]. Later, the measurements developed for CBR-PEB where successfully analyzed if they are applicable to a similar system, KM-PEB (Knowledge Management Product Experience Base [2]). The initial GQM trials lead to the development of an initial set of feedback channels [4]. Further case studies for a GQM-based evaluation of an EBIS in the SE field are the evaluations of the SFB repository and the EMS of FC-MD [22]. The ESERNET case study [12] (Section 5.2) demonstrates the application of the mature approach as described above. In the SKe project (http://www.ske-projekt.de/) the method is being applied to evaluations of CBR-based EBIS in the IT security domain.

5.1 Project CBR-PEB

CBR-PEB is a CBR-based EBIS that has been developed for supporting CBR system development. 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 (http://demolab.iese.fhg.de:8080/).

This EBIS is based on a number of research efforts: Althoff et al. [1] developed the classification criteria for CBR systems. Bartsch-Spörl, Althoff, and Meissonnier [5] conducted a survey about CBR systems and made these experiences reusable by means

Goal:	*Analyze*	the retrieved information	**Names:**	NN, NN
	for the purpose of	monitoring	**Date:**	97/10/01
	with respect to	economic utility		
	from the viewpoint of	the CBR system developers		
	in the context of	decision support for CBR system development.		

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)** [...]

Fig. 5. Identification of Maintenance Knowledge in the abstraction sheet for Goal 2 "Economic Utility" for CBR-PEB.

of CBR technology, that is, each questionnaire has been represented as a structured case. Finally, an evaluation program was developed to show the usefulness of the system from the viewpoint of its users [23].

Because CBR-PEB was the first application of the GQM method to a CBR-based EBIS, the aim of this case study was to find out if GQM is a feasible approach to evaluating the success of EBIS/CBR systems. The case study was a success regarding this goal and laid the basis for further research in that field.

Since CBR-PEB had already existed for a while before the evaluation and the experts knew the system, CBR-PEB could be considered to be in Phase 2 regarding the 3-Phase Model (Section 3.2). Therefore, CBR-PEB-specific measurements could be developed. These CBR-PEB-specific measurements focused on the usefulness of the retrieved information regarding the economic and technical utility. The abstraction sheet for the economic utility is depicted in Fig. 5.

In an additional analysis of the abstraction sheets for CBR-PEB, we discovered that we also had defined some additional "background" knowledge about certain dependencies, which could be empirically validated through the measurement program [20]. E.g., in the abstraction sheet in Fig. 5, the quality factor "degree of maturity" is related to the variation factor "case origin". This helps us define a strategy for the maintenance of the case base regarding the acquisition of new cases (i.e., a maintenance policy): Depending on what users find more useful regarding the degree of maturity, acquire more cases from "industry" or more from "university". This discovery coined the idea of using such knowledge as empirically validated maintenance knowledge and showed the feasibility of deriving maintenance knowledge from the evaluation program.

For the feedback channels (excerpt from GUI in Fig. 8), the monitoring of the usefulness feedback for CBR-PEB over 5 years showed that approx. 9% of the users gave usefulness feedback. The reason for this rather low number is that CBR-PEB is not integrated in workflow or process in a company. However, industrial experiences for such usefulness feedback mechanisms without integrating the CBR system in a process or workflow are around 5%.

For feedback rates of 5-10%, the usefulness feedback cannot be regarded as statistically valid data that represents the opinions of all users. However, it is useful to point to problems and to trigger maintenance. The triggered maintenance has to consider further investigations to deal with the problem of the insufficient statistical validity of the usefulness feedback for a system without tight integration into a workflow or work process.

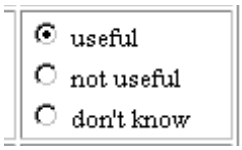


Fig. 8. Usefulness feedback for single cases in CBR-PEB's query result GUI.

5.2 Project ESERNET

ESERNET aims at improving maturity and competitiveness of European software intensive organizations. For this purpose, cases about best SE practices, their benefits as well as their context requirements and boundaries are shared via the ESERNET EBIS (http://www.esernet.org/). Such knowledge enables companies to improve faster and at lower cost.

The ESERNET EBIS as well as CBR-PEB are both publicly available in the web. ESERNET is officially funded by the European Union.

At the point in time when the evaluation program was developed (July 2002), the first release of the ESERNET EBIS was almost finished. Thus, according to the 3-phase model, the ESERNET EBIS was considered to be in Phase 1.

We expected the evaluation of ESERNET [12] to be another case study for Phase 1 of the 3-Phase Model. Furthermore, the existing models for the utility goal should be confirmed with a GQM interview. A more specific goal was to integrate the feedback channels into the method. The hypothesis is that the feedback channels can be assessed based on the current implementation and adjusted using the information on feedback channels in [3]. In particular, we wanted to validate the identification of maintenance policies based on GQM abstraction sheets or GQM plans.

For ESERNET, we derived five maintenance policies from the abstraction sheets and GQM plans for the goals 1, 2, and 4. Two policies have a different version for Phase 1 and Phases 2 & 3. Section 5.2.1 describes this for two of these maintenance policies. The first policy is derived from a quality factors. The second policy is derived from a quality factor and related variation factors as well as baseline hypotheses. Furthermore, Goal 3 provides the maintenance policies related to the improvement actions for the feedback channels. Here, we simply had to select the feasible and relevant improvement actions based on the analysis of the available feedback channels [12]. The remaining ESERNET maintenance policies can be found in [13].

5.2.1 Deriving Maintenance Policies from GQM Abstraction Sheets

The first maintenance policy combines a quality factor, the respective baseline, and several variation factors (see Fig. 9). The policy (Fig. 10) deals with fill-out questionnaires that are not as complete as expected. The variation factors "type of organization" and

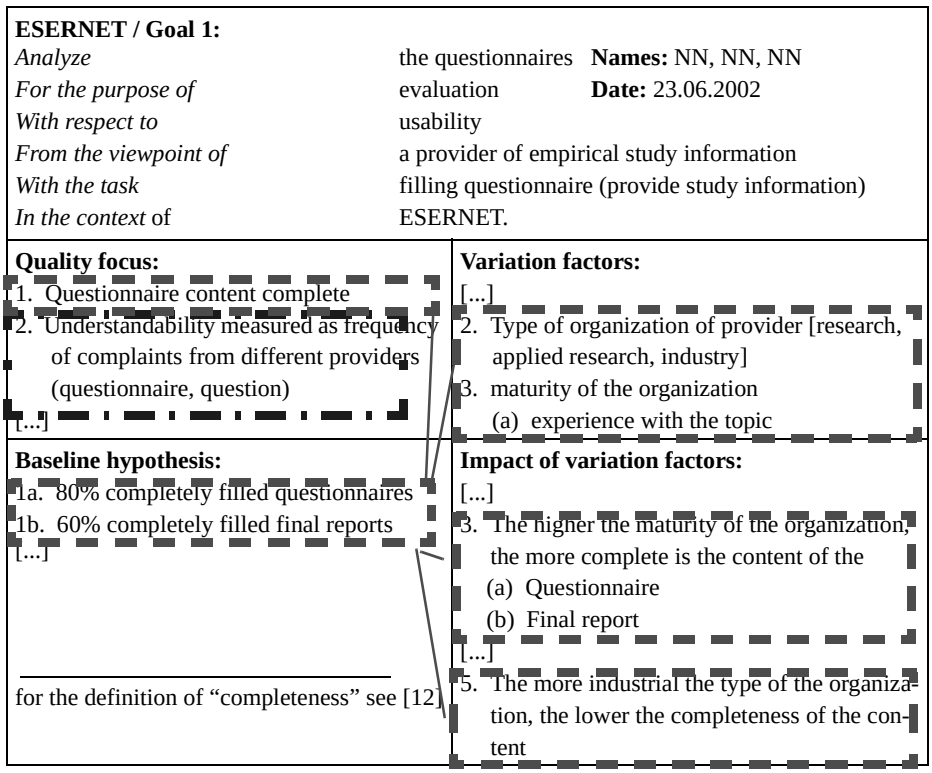


Fig. 9. Abstraction Sheet for Goal 1 for ESERNET focusing on the usability of the web questionnaires. Relevant items and respective relationsships for two maintenance policies are highlighted.

Trigger: new submission of filled-out questionnaire, *[Goal 1, QF1]*
completeness of submission < expectation [type of organization, maturity of organization]

Actions:

- decide if further action is required:
  - criteria: to be learned
- if yes (i.e., further action is required):
  - ask the provider to revise his submission within 2 weeks
  - if the provider has problems with certain questions (see Q-5 "understandability of questions by provider"), clarify the questions with the provider

Expected benefits:

- better completeness of study descriptions

Fig. 10. Example of a maintenance policy derived from a quality factor (quality factor 1 of goal 1), respective baseline hypothesis, and variation factors.

"maturity of organization" select the respective expectation regarding completeness. This expectation will change over time, that is, a good expectation or baseline has to be learned over time. Furthermore, evaluation results regarding understandability problems with questions are considered in the actions taken.

From Quality Focus 2 of Goal 1, which is marked in Fig. 9, a second maintenance policy has been derived (see [13]).

5.2.2 Major Lessons Learned from ESERNET

Evaluation. The quality models for utility [22, 3] and the evaluation method in general [22, 3, 23] were validated another time. Furthermore, we developed a quality model for the perceived utility analyzed from the viewpoint of the maintainers. Furthermore, the following lessons were learned:

- For the evaluation personnel, we have to distinguish between the web usage evaluation and the GQM-based evaluation. For the web usage evaluation, standard IT knowledge is sufficient due to good tool support. For the utility and feedback evaluation, which is still a new field, no standard tools exist yet. Because adaptations and discussions are quite frequent, experienced evaluation personnel or consultants are required to bootstrap and coach such an evaluation.
- The implementation of the data collection procedures has to be checked to ensure that the measurement program is really implemented correctly (e.g., the "mandatory metrics" are complete to ensure that meaningful evaluations are possible [3]) and that not some metrics are discarded. If some are discarded, the measurement specialist has to analyze the potential "damage" to the measurement program by tracing bottom up from the measurement plan through the GQM plans to the abstraction sheets to identify what cannot be analyzed any more and discuss these findings and issues with developers and repository-responsible persons. This check is in particular required when measurement specialist and EBIS developer(s) are not the same person.
- Quality focus "coverage" in Phase 1: The development of a more detailed definition then "number of cases per topic/subject area" is a difficult and time-consuming task. We recommend to simply using the measure "number of cases per topic/subject area" because it is easy to comprehend and the respective data is easy to collect and analyze. To monitor the trend, you have to state the expectations about the number of cases and about the type of trend (e.g., linear).

The stated lessons learned were integrated into the evaluation method [3].

Maintenance. First, we validated the EBIS maintenance method, in particular, deriving maintenance policies from GQM plans and using feedback channels.

Second, we developed a set of generic maintenance policies for SE EBIS for perceived utility for two viewpoints (user and maintainer). Some of the maintenance policies have variants for the different phases or stages. For the viewpoint user, the maintenance policies are similar to the improvement action-cause-indicator tuples from the feedback channels. However, when there are several causes for an indicator, this has to be considered for the maintenance policy.

6 Conclusion

We presented the EMSIG-KA method for the acquisition of the knowledge required for maintenance and evaluation of software engineering experience repositories in particular and experience-based information systems in general. The method allows having the required maintenance and quality knowledge available when the system going into "regular use" or "online". Furthermore, the developed knowledge can be integrated into the CBR system/EBIS following the example of the EMSIG framework [21].

The case studies have demonstrated that the method is effective regarding delivering the maintenance knowledge before the "regular use". The method is also efficient because it allows a quick and inexpensive start with predefined models for evaluation and maintenance. These models are the feedback channels and the quality model for the beginning of the regular use, where the focus is on acceptance of the system.

The maintenance is linked to the evaluation in a systematic and traceable manner. The GQM plans were used as starting point to identify the "when" of maintenance. Based on that, so-called maintenance policies were developed, which describe the "when", "how", and "why" of maintenance. A set of generic maintenance policies for CBR systems/EBIS for measuring utility could be identified.

The maintenance policies in the ESERNET case study were reviewed by the ESERNET project team at IESE to check the meaningfulness before continuous operation. However, only the actual continuous operation will show the actual qualities of the policies. Some of the maintenance policies also require further development during continuous operation or have different variants depending on further maintenance experience gained during the actual maintenance. This shows that learning is also an important issue for CBR maintenance.

Future work will focus on system-specific measurements and maintenance policies. We want to identify respective models that are common for systems of a certain environment, task, domain, or lifecycle model.

Acknowledgement

The work presented has partly been funded by the Information Societies Technology Programme of the European Union (IST-2000-26754). The complete list of partners and members can be found at www.esernet.org.

References

[1] K.-D. Althoff. Evaluating case-based reasoning systems: The Inreca case study. Postdoctoral thesis (Habilitationsschrift), University of Kaiserslautern, 1997.

[2] K.-D. Althoff, W. Müller, M. Nick, and B. Snoek. Km-peb - an online experience base on knowledge management technology. In E. Blanzieri and L. Portinale, editors, *Advances in Case-Based Reasoning:Proceedings of the Fifth European Workshop on Case-Based Reasoning*, Lecture Notes in Computer Science. Springer Verlag, 2000.

[3] K.-D. Althoff and M. Nick. *How To Support Experience Management with Evaluation - Foundations, Evaluation Methods, and Examples for Case-Based Reasoning and Experience Factory*. Springer Verlag, 2003. (to appear).

[4] K.-D. Althoff, M. Nick, and C. Tautz. Improving organizational memories through user feedback. In *Workshop on Learning Software Organisations at SEKE'99*, Kaiserslautern, Germany, June 1999.

[5] B. Bartsch-Spörl, K.-D. Althoff, and A. Meissonnier. Learning from and reasoning about case-based reasoning systems. In *Proceedings of the Fourth German Conference on Knowledge-Based Systems (XPS97)*, Mar. 1997.

[6] V. R. Basili, G. Caldiera, and H. D. Rombach. Goal Question Metric Paradigm. In J. J. Marciniak, editor, *Encyclopedia of Software Engineering*, volume 1, pages 528–532. John Wiley & Sons, 1994.

[7] R. Bergmann. Experience management - foundations, development methodology, and internet-based applications. Postdoctoral thesis, Department of Computer Science, University of Kaiserslautern, 2001.

[8] R. Bergmann, K.-D. Althoff, S. Breen, M. Göker, M. Manago, R. Traphöner, and S. Wess. *Developing Industrial Case Based Reasoning Applications - The INRECA Methodology*. LNAI. Springer Verlag, 2003.

[9] L. C. Briand, C. M. Differding, and H. D. Rombach. Practical guidelines for measurement-based process improvement. *Software Process*, 2(4):253–280, Dec. 1996.

[10] T. CEMP Consortium. Customized establishment of measurement programs. Final report, ESSI Project Nr. 10358, Germany, 1996.

[11] W. S. Cooper. On selecting a measure of retrieval effectiveness. In K. Jones and P. Willet, editors, *Readings in Information Retrieval*, pages 191–204. Morgan Kaufmann Publishers, 1997.

[12] A. Jedlitschka and M. Nick. Repository validation report. Technical Report Project ESERNET (IST-2000-28754) - Deliverable D.4.4, Fraunhofer IESE, Kaiserslautern, Germany, 2002.

[13] A. Jedlitschka and M. Nick. Requirements and design of the ESERNET ESE knowledge repository. Technical Report Project ESERNET (IST-2000-28754) - Deliverable D.4.1 (v1.1), Fraunhofer IESE, Kaiserslautern, Germany, 2003.

[14] D. B. Leake, B. Smyth, D. C. Wilson, and Q. Yang, editors. *Computational Intelligence - Special Issue on Maintaining CBR Systems*, 2001.

[15] D. B. Leake and D. C. Wilson. Categorizing case-base maintenance: Dimensions and directions. In B. Smyth and P. Cunningham, editors, *Advances in Case-Based Reasoning:Proceedings of the Fourth European Workshop on Case-Based Reasoning*, pages 196–207, Berlin, Germany, Sept. 1998. Springer-Verlag.

[16] T. Menzies. Knowledge maintenance: The state of the art. *The Knowledge Engineering Review*, 14(1):1–46, 1998.

[17] M. Nick and K.-D. Althoff. Acquiring and using maintenance knowledge to support authoring for experience bases. In R. Weber and C. G. von Wangenheim, editors, *Proceedings of the Workshop Program at the Fourth International Conference on Case-Based Reasoning (ICCBR'01)*, Washington, DC, 2001. Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence.

[18] M. Nick and K.-D. Althoff. Engineering experience base maintenance knowledge. In K.-D. Althoff, R. Feldmann, and W. Müller, editors, *Advances in Learning Software Organisations - Proceedings of the Third International Workshop on Learning Software Organisations*, number 2176 in Lecture Notes in Computer Science, Heidelberg, Germany, Sept. 2001. Springer Verlag.

[19] M. Nick, K.-D. Althoff, T. Avieny, and B. Decker. How experience management can benefit from relationships among different types of knowledge. In M. Minor and S. Staab, editors, *Proceedings of the German Workshop on Experience Management (GWEM2002)*,

number P-10 in Lecture Notes in Informatics (LNI), Bonn, Germany, Mar. 2002. Gesellschaft für Informatik.

[20] M. Nick, K.-D. Althoff, and C. Tautz. Facilitating the practical evaluation of organizational memories using the goal-question-metric technique. In *Proceedings of the Twelfth Workshop on Knowledge Acquisition, Modeling and Management*, Banff, Alberta, Canada, Oct. 1999.

[21] M. Nick, K.-D. Althoff, and C. Tautz. Systematic maintenance of corporate experience repositories. *Computational Intelligence - Special Issue on Maintaining CBR Systems*, 17(2):364–386, May 2001.

[22] M. Nick and R. Feldmann. Guidelines for evaluation and improvement of reuse and experience repository systems through measurement programs. In *Third European Conference on Software Measurements (FESMA-AEMES 2000)*, Madrid, Spain, Oct. 2000.

[23] M. Nick and C. Tautz. 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, Mar. 1999. LNAI Nr. 1570.

[24] T. Reinartz, I. Iglezakis, and T. Roth-Berghofer. On quality measures for case base maintenance. In E. Blanzieri and L. Portinale, editors, *Advances in Case-Based Reasoning:Proceedings of the Fifth European Workshop on Case-Based Reasoning*, pages 247–259. Springer-Verlag, 2000.

[25] M. M. Richter. Introduction. In M. Lenz, B. Bartsch-Spörl, H.-D. Burkhard, and S. Wess, editors, *Case-Based Reasoning Technologies: From Foundations to Applications*, number 1400 in Lecture Notes in Artificial Intelligence, chapter 1, pages 1–15. Springer-Verlag, Berlin, Germany, 1998.

[26] T. Roth-Berghofer. *Knowledge maintenance of case-based reasoning systems. The SIAM methodology*. PhD thesis, University of Kaiserslautern, Kaiserslautern, Germany, 2002.

[27] C. Tautz. *Customizing Software Engineering Experience Management Systems to Organizational Needs*. PhD thesis, University of Kaiserslautern, Germany, 2001.

[28] R. Weber, D. W. Aha, and I. Becerra-Fernandez. Intelligent lessons learned systems. *International Journal of Expert Systems - Research & Applications*, 20(1), 2001.