

GUIDELINES FOR EVALUATION AND IMPROVEMENT OF REUSE AND EXPERIENCE REPOSITORY SYSTEMS THROUGH MEASUREMENT PROGRAMS

Markus M. Nick*, Raimund L. Feldmann**

* Fraunhofer Institute for Experimental Software Engineering (IESE), Kaiserslautern, Germany
nick@iese.fhg.de

** Research Group Software Engineering (AGSE), University of Kaiserslautern, Germany
r.feldmann@computer.org

Abstract. Software development organizations are recognizing the increasing importance of investing in the build-up of core competencies for their competitiveness in software system development. This is supported by reuse and experience repository systems that assist in capturing and reusing all kinds of software artifacts (e. g., code, patterns, frameworks) and processes as well as experiences related to these artifacts and processes. To justify such an investment and guide its improvement, it must be evaluated according to the business case, that is, a measurement program has to be developed that is oriented towards the business goals of such a reuse and experience repository system.

In this paper, we suggest an approach to iteratively build up measurement programs for gaining feedback and, thereby, controlling and improving such a reuse and experience repository system. The focus is placed on guidelines for the evolution of such measurement programs over time, rather than providing directly applicable metrics or questionnaires. In order to illustrate the feasibility of the approach, examples of running measurement programs at different stages of evolutions are given.

1 INTRODUCTION

To master today's needs of competitive software system development, companies have to improve their software products and processes continuously. For this purpose, there are several approaches that complement each other to reach a certain quality standard: First, (repeatable) processes are defined and/or elicited to ensure the development of software products with an assured quality. This is addressed by standards such as CMM [26], ISO9000 [12], or SPICE [23]. Second, the reuse of all kinds of software artifacts (e. g., code, patterns, frameworks) allows to leverage the development of high-quality products by reusing products that already conform to quality assurance standards. Third, capturing and reusing experiences (which enriches processes, development tools, and products with lessons learned during daily project work) enables company-wide learning on-the-job on an (almost) day-to-day basis. All this means that continuous capturing and dissemination of relevant knowledge (in the form of processes, artifacts, and experiences) across projects is required for a software development organization.

The goal of such continuous capturing and dissemination of relevant corporate software engineering knowledge and experience across projects conflicts with the aim of the project teams, that is, to deliver software of a certain quality. Thus, it is necessary to have a separate organizational unit that manages corporate knowledge and experience. Basili and Rombach call such an organizational unit experience factory (EF) [4]. An EF includes as its core element an experience base (EB), which is a repository for all kinds of software artifacts, processes, and experiences (i.e., highly situated¹, context-sensitive knowledge items) that are related to these artifacts and/or processes [2, 4]. The experience factory and experience base idea is related –in general– to the concepts of knowledge management, organizational learning, and organizational memories. Single, specific tasks of an experience factory or experience base can also be assisted by specific knowledge-based systems that (usually) aim at supporting a single, very specific task.

For the purpose of this paper, we refer with the term “reuse and experience repository system” to the following four items: the technical infrastructure (i.e., some software) that is used for storing and retrieving all kinds of knowledge (processes, artifacts, and experiences), the ontology underlying the

1. The situation is typically defined by the process and project the user is currently in.

knowledge and experience, the actual contents of the repository (i.e., the knowledge itself) and the processes for populating, improving, and utilizing the knowledge. Such a repository system can be developed using a methodology such as DISER (see [27]).

It is widely accepted that such a reuse and experience repository (like an experience base) must show a high quality (e.g., [1] or [6]). To ensure this need for high quality, evaluation is required that can guide the development and maintenance of such a repository [3, 6, 14]. Problems with existing evaluation studies [18] show that there is a need for a systematic approach that helps in conducting "good measurements" that are related to the business case and business goals of the repository system. This leads to questions concerning cost/benefit of repository systems, their user friendliness, or maintainability to avoid "data cemeteries".

This paper presents a methodology that facilitates the evaluation and improvement of reuse and experience repository systems by iteratively conducting goal-oriented measurement programs for gaining feedback, and, thereby, controlling and improving such reuse and experience repository systems. Our evaluation approach employs the industrial-strength goal-question-metric (GQM) method to develop quality models in order to smoothly integrate our measurement activities into already existing programs. First results from applying our iteration steps in the context of the SFB-EB [9, 10], the EMS system [24, 29], and the CBR-PEB case study [20, 21] illustrate the feasibility of the suggested GQM plans.

The remainder of this paper is organized as follows: Section 2 reflects related research and discusses existing evaluation models for reuse repositories. Then, we describe the basic concepts of our approach and our quality model (Section 3). Furthermore, we introduce the resulting GQM plan levels for measuring reuse and experience repository systems. In Section 4, we present outcomes of first practical evaluations of our approach as far as they are implemented in the context of our organizations. Section 5 illustrates how evaluation can be exploited to guide the improvement of reuse and experience repository systems. Finally, we summarize our results and conclude with future directions in Section 6.

2 MEASURING REUSE AND EXPERIENCE REPOSITORY SYSTEMS

In literature, a lot of articles can be found that deal with the definition and implementation of measurement programs and suitable metrics for the purpose of evaluation and control (e.g., [7] or [22]). While there are many descriptions that concentrate on measuring product or process quality, like for instance, of code or applied verification processes, only a few metrics or measurement programs are discussed concerning the evaluation of reuse and experience repository systems. However, recently this topic has been addressed, for instance, at the 24th 'Annual Software Engineering Workshop' (SWE24) or at 1999's 'International Workshop on Learning Software Organizations' (LSO'99).

Koennecker et. al. discuss a method for developing an EF organization that incorporates a step called 'analyze how the experience repository is used' [15]. It is reported how they initially measured the growth of the repository in terms of number of documents and number of search requests per week. To create their usage reports they used the access log data of the web server on which the repository was installed. This is similar to the approach described in [9]. Here, a prototype of an EB was implemented using web technologies. Based on the web server log file, a regularly updated usage statistic is plotted to gain fast feedback on how newly-structured or added areas of the repository are accepted by the user. Furthermore, the usage of the repository is compared to other tools installed in the author's software development laboratory.

In [29], the authors describe experimental plans for measuring their experience management system. The mentioned pilot study lists the following goals: a) to evaluate the current set of attributes used to describe the artifacts stored in the repository, and b) to evaluate the retrieval and data entry interfaces. Results of a first evaluation, partly being very specific to the measured repository system, are published in [24].

From these examples it becomes obvious that there are many dimensions that have to be thought of when talking about evaluating and controlling repository systems. However, as with all measurement related activities, one should remember the following statement by Johansson et. al. [13]: 'When an organization starts measuring, its approach to measurement is [often] too comprehensive so too many measures are defined'. Hence, initial measurement programs for evaluating repository system should be kept simple and small. Especially, because gathering measurement-based experience is a

laborious and time-consuming endeavor, as, for instance, stated in [11]. Taking this into account, we introduce our approach to iteratively build up measurement programs for reuse and experience repositories. To improve readability we will use Basili's Experience Base (EB) as a representative to address all repository systems that can be measured with our approach in the remainder of this paper.

3 THE SUGGESTED APPROACH: GQM-BASED EVALUATION

Our approach to EB evaluation consists of three parts: A methodological framework for planning and conducting measurement programs (Section 3.1). A high-level quality model showing how evaluation goals, criteria, and metrics for EBs evolve over time (Section 3.2). And finally, a technique for iteratively refining evaluation goals and criteria (Section 3.3).

3.1 GQM – An Industrial-Strength Method for EB Evaluation

The Goal-Question-Metric (GQM) method is an industrial-strength technology for goal-oriented software engineering measurement, which has been successfully applied in several companies [5, 22]. It systematically facilitates planning and conducting measurements by helping to define and implement operational and measurable software (improvement) goals. In [20] we showed that GQM also meets the requirements for 'good measurements' in Knowledge Engineering [19, 28] and, thus, is suitable for evaluating a knowledge-based system like an EB.

Motivations for goal-oriented measurement with GQM –according to [7]– are to ensure adequacy, consistency, and completeness of a measurement plan¹, deal with the complexity of measurement programs, and stimulate a structured discussion about measurement. 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.

In the following, we summarize how to conduct EB evaluation systematically and where existing methods, quality models, etc., can be plugged in. Note that this cannot give you a complete picture of GQM. For more in-depth information, please refer to [20].

- Based on the business goals of the EB (i. e., the objectives and success criteria identified by the stakeholders during EB build-up), actual measurement goals (so-called GQM goals) are developed. GQM goals differ by the dimensions they address: the object (e. g., technical infrastructure, retrieved information), the context (e. g., competence area as identified during build-up, company or organizational unit), and the focus (e. g., usage, utility, (economic) value, user friendliness).
- For each GQM goal, an interview is conducted with the respective stakeholders. Using so-called "abstraction sheets", evaluation criteria and their dependencies are identified: "Quality factors" characterize 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. A hypothesis about this impact is described under "impact of variation factors".
- Based on such an abstraction sheet, a GQM plan is developed for each goal. This is an operational refinement of a GQM goal (the "G") via questions (the "Q") (which are derived from the evaluation criteria in the abstraction sheets) into measures (the "M") including the analysis models that specify how the measurement data is analyzed to help answer the questions. This GQM plan is the perfect place for plugging in existing metrics and quality models: For example, a question (or evaluation criterion) on the competence coverage of the EB could either use a very simple metric such as the "number of items in a competence area" or an advanced quality model like the competence model from Smyth & McKenna [25].
- From the GQM plans, a measurement plan is derived. This is the actual description of the implementation of the measurement program stating which data is collected when, how, and by whom.

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

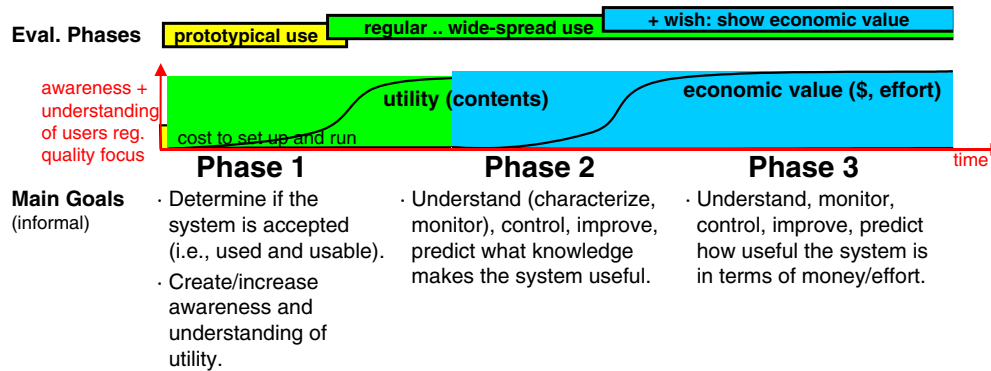


Fig. 1: Quality model for the development of goals (and metrics) over time

Then, the respective data is collected and analyzed.

- In “feedback sessions” with the stakeholders, the analysis results are interpreted and compared to the goals. Based on this, decisions for improvement actions can be made. Improvement actions can be, for example, as follows: Simple metrics are replaced with more advanced models (e. g., replace a metric “number of items in a competence area” with the competence model from Smyth & McKenna [25]). A new competence area is added to the EB; thus, a roll-out of the evaluation program for the new area is necessary, that is, another iteration of the GQM is started.
- To continuously learn about evaluation, quality models as well as lessons learned from the evaluation are collected and stored in an “evaluation experience base”.

The “never ending” iteration of GQM with refinements and roll-outs as well as collection of quality models etc. leads to a learning spiral: We learn more and more about evaluation and, thus, improve the evaluation itself.

3.2 Quality Model for Meaningful Goals and Metrics over Time

Depending on the maturity level of the EB, the measurement program has to be adapted. Hence, the quality models used for evaluating a single EB have to change over time. Fig. 1 illustrates these changes. In our model we distinguish three phases for the evaluation of an EB (oriented at the maturity of the system): Phase 1 consists of the initial definition and implementation of the EB. Furthermore, the prototypical usage in the beginning, where the system is changed quite often according to new/changed requirements and wishes of the users, is included in this phase. Phase 2 is characterized by the regular use of the system. Phase 3 is characterized by the understanding and analysis of the economic viability of the system. These phases are not strictly separated, but rather overlapping. At the end of one phase and at the beginning of the next phase, case studies and experiments help improve the awareness regarding the actual quality of the system.

These phases are reflected in GQM programs for EBs as follows:

- In Phase 1, acceptance can be simply measured by the usage of the system (e. g., percentage/ number of persons that used the system several times). Textual feedback should be considered to get hints on user requirements that are not yet (completely) covered by the system as well as to get a first impression on the utility of the system (incl. its contents).

The focus on acceptance is meaningful because the acceptance of an EB by the intended users is crucial, since 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.

- Phase 2 focuses on the guidance of the development by the evaluation. Ideally this should be conducted as a controlled experiment.

Thus, on the one hand, 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 not disturb improvement activities. Hence, the evaluation must be able to deal with regular maintenance such as, for example, adding, modifying, or deleting experience items. This also must be reflected by respective variation factors.

- Phase 3 is dominated by trying to measure actual costs and benefits to determine the economic value of the EB (e. g., via its return on investment). A detailed theoretical analysis of costs and benefits for experience bases can be found in [21]. The costs can be measured relatively easy right from the start of the EB project (i. e., measuring the effort that is needed to set up, run, and use the system). The actual benefit is much more difficult to measure. Theoretically, this would have to be the subjective value of the information to the user [8]. Because this is difficult to measure practically and economically, we have to use other measures, which are developed as part of the GQM process.

3.3 Iterative Refinement of Evaluation Programs

The quality model from the former section requires a refinement of goals and measurement criteria over time. This refinement happens for several “dimensions”:

- The criteria for one goal are refined. For example, in the beginning (Phase 1) utility is mainly characterized by textual feedback and later characterized and monitored by detailed measurement criteria (Phase 2). This is covered by iterating the GQM process and, thus, part of the “standard” GQM method.
- Certain (high-level) criteria define the quality focus for more detailed goals (see Fig. 2). For

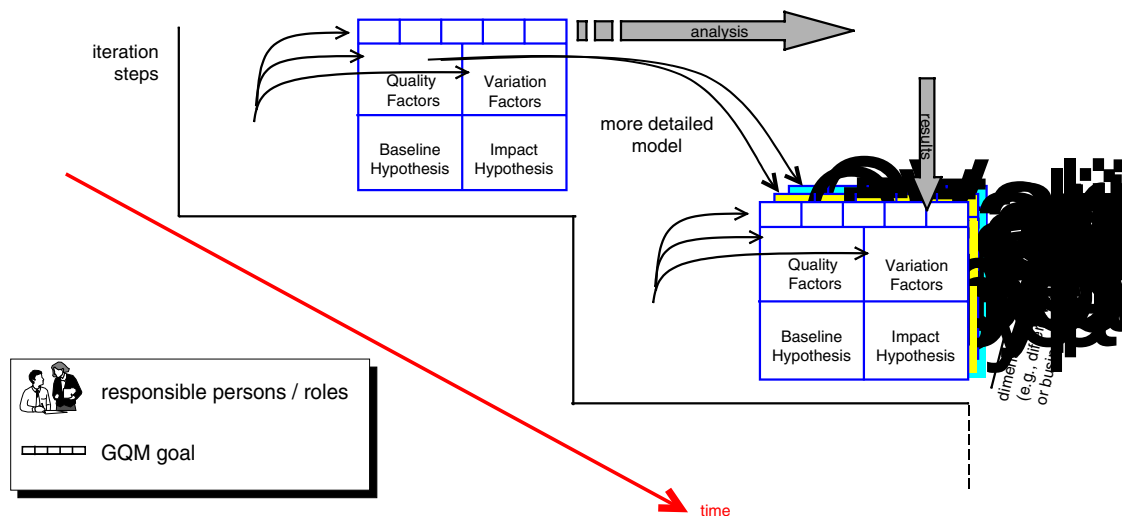


Fig. 2: GQM plan levels: Refinement from an initial measurement program to detailed analysis

example, a high-level goal deals with the quality of the retrieved experience items from the users’ viewpoint and the respective abstraction sheet mentions the (perceived) usefulness as one criterion. This is –later in the evaluation process– used as a separate goal (i. e., usefulness/utility of retrieved experience items from the users’ viewpoint) and described by a separate abstraction sheet with detailed quality criteria.

- The evaluation program is rolled out and refined also by competence areas, business units, and viewpoints. For example, the high-level goals and criteria are defined by the management of the organization whereas the criteria at lower levels are elicited from the respective users of the experience base (i. e., the viewpoint is different).

So far, this iterative refinement model is partially validated by the case studies presented in the next

section as well as by the experience with GQM in general.

4 PRACTICAL EXAMPLES

Although the approach has not yet been applied completely for an EB, the main parts have been validated through several projects. In Section 4.1 we give examples for the evaluation of two reuse repository systems in Phase 1 of our defined model. Section 4.2 reports on results of a deeper analysis of a certain system, namely the CBR-PEB case study. This case study can be compared to the requested deeper analysis of repository systems in Phases 2 or 3.

4.1 Measuring in Phase 1: The EMS/SFB-EB example

This section describes an example for a high-level GQM program performed in the context of two prototype experience repository systems, namely EMS [16, 17] and the SFB-EB [9, 10]. According to our quality model for meaningful goals and metrics over time, both programs belong to Phase 1: 'prototypical use'. The measurement program for EMS (Experience Management System) is being performed at the Fraunhofer Center for Experimental Software Engineering, Maryland (FC-MD¹). EMS was developed by the Experimental Software Engineering Group (ESEG²) at the Department of Computer Science at the University of Maryland in College Park in cooperation with FC-MD. Measurement for the SFB-EB is being performed by the Research Group Software Engineering (AGSE³) at the Department of Computer Science at the University of Kaiserslautern, Germany, in the context of the "Sonderforschungsbereich 501" (SFB 501⁴).

	Goal 1 "Quality of retrieved experience items"	Goal 2 "User satisfaction"	Goal 3 "experience item usage"
Analyze	the retrieved experience items	the EF system	the experience item
for the purpose of	evaluation	improvement	characterization
with respect to	quality	user satisfaction	usage
from the viewpoint of	the users of the EF system	the developers of the EF system	the management of the EF system
in the context of	FC-MD / SFB 501	FC-MD / SFB 501	FC-MD / SFB 501

Fig. 3: GQM goals for EMS and the SFB-EB

This measurement program consists of three goals (Fig. 3). All of them are aiming at a high level since each of the quality foci in the abstraction sheet can be refined into a new GQM goal for a lower-level GQM analysis. The abstraction sheet of the first goal is shown in Fig. 4. As can be seen, the Quality Factor "usefulness" is influenced by Variation Factors 'e' through 'j', 'l', and 'o'.

We want to take parts of this abstraction sheet (Fig. 4) as an example for measures for which data is collected manually. In Fig. 5 you find an example for gathering measurement data referring to Variation Factors 'f' and 'g' using an HTML questionnaire. However, besides the set of measures for which data is collected manually we also deal with measures for which data will be collected automatically by the EF system. In the GQM plan "experience item usage", for instance, there are many measures that affect log data about who uses which experience item, how and when. This automatic gathering of measurement data (also in combination with a regular questionnaire) is important to increase the acceptance of the measurement program by the users. One main goal of this whole program is to minimize the time users have to spend on answering questions, so the questionnaire will be as unobtrusive as possible.

4.2 Measuring in Phase 2: The CBR-PEB Case Study

The CBR-PEB case study was quite typical for an EB in Phase 2. The modeling was relatively stable and the users had some knowledge about the contents of the EB. Thus, it was possible to focus on

1. <http://fc-md.umd.edu>
2. <http://www.cs.umd.edu/projects/SoftEng/ESEG>
3. <http://www.wagse.informatik.uni-kl.de>
4. <http://www.sfb501.uni-kl.de>

Goal: Analyze the retrieved experience item (EI) for the purpose of evaluation with respect to quality from the viewpoint of the users of the EF system in the context of FC-MD / SFB 501.	
Quality factors:	Variation factors:
1 stability 2 usefulness 3 trust 4 correctness	a)number of modifications b)number of (reported) errors for each error class in EI c)age of EI d)how was EI verified e)appropriateness of classification f) appropriateness of classification/documentation (incl. abstraction level) g)quality of description (e. g., grammar, language) h)complete context i) reusability of EI (how generic?) j) relevance of EI k)number of reuse cycles l) type of project the EI came from m)'author' of the EI n)'packager' of EI o)format of EI files
Basel. hyp.:	Impact of variation factors:
	a → 1: more modifications over time, less stability b → 3, 4: more errors, less trust and less correctness c → 1: younger EI (same # modifications), less stable d → 3: certain certifications raise trust e → 2: good class. improves usefulness f → 2: good descr./docu. impr. usefulness g → 2: good descr. improves usefulness h → 2: missing context info, less useful i → 2: more generic EI, more useful j → 2: more relevance, more useful k → 3: more reuse cycles, more trust l → 2: closer related projects more useful m → 3: author might raise/decrease trust n → 3: pack. might raise/decrease trust o → 2: some formats might be unusable

Fig. 4: Abstraction sheet for goal "Quality of retrieved experience items"

How good is the quality of the EI description grammatically and language-wise?

very good
 good
 acceptable
 bad
 very bad

Is the context of the EI complete? If no, please comment.

yes

no

Fig. 5: Partial screenshot of questionnaire (excerpt)

utility and elicit detailed criteria for utility. In this section, first the system is introduced as context for the evaluation and then selected parts and artifacts from the GQM process are presented.

CBR-PEB (Case-Based Reasoning Product Experience Base)¹ is an EB on Case-Based Reasoning (CBR) systems. Users (such as CBR project managers and researchers) can query CBR-PEB for

1. <http://demolab.iese.fhg.de:8080/>

state-of-the-art and -practice studies, feasibility studies, and to receive decision support for CBR system development (i. e., to choose an appropriate application or tool as starting point for their own development activities).

CBR-PEB can be viewed as an EB for a globally distributed company. This means that it is difficult to motivate users to feed cases into the system and to make them enter measurement data because most of the users are not known personally to the EF staff and cannot be trained as actually needed. This has the consequence that the EB should be as easy to use as possible and that the collection of measurement data should be as unobtrusive as possible.

For the evaluation itself, we proceeded as described in Section 3.1. Two representatives of the users (two CBR experts) participated in the GQM interviews and feedback sessions. They already had some knowledge about the contents of the experience base. Thus, we could start the evaluation right away with Phase 2 and elicit detailed criteria for utility.

The GQM goals (see Fig. 6) focus mainly on utility (technical and economic utility) and address the most interesting item for the users: the retrieved information. Only one goal addresses “user friendliness.”

	Goal 1 “Technical Utility”	Goal 2 “Economic Utility”	Goal 3 “User Friendliness”
Analyze	retrieved information	retrieved information	experience base
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.

Fig. 6: The list of formal, ranked, and selected GQM goals for CBR-PEB

The abstraction sheets for goals 1 and 2 (technical and economic utility – see Fig. 8 for the abstraction sheet for economic utility) are typical for Phase 2: Detailed criteria are defined for the utility of the retrieved information (i. e., experience items). These criteria are related to the contents of the experience items for quality and variation factors (e. g., case origin and degree of maturity) and to the experience base as a whole for some variation factors (e. g., number of cases as a simple metric for the competence/coverage of the experience base, amount of background knowledge as a simple metric for the maturity of the domain model).

The criteria for the goal “user friendliness” were more typical for Phase 1 because the web interface of CBR-PEB was still under development when the evaluation was started and the GQM interviews were carried out. This led to a list of non-functional requirements and respective wishes instead of actual evaluation criteria.

To keep the effort for evaluation low on the side of the user, most of the measurement data is collected automatically (e. g., similarity, certain attribute values such as case origin and degree of maturity). The remaining few items are collected using on-line forms (see Fig. 7 for an example).

- 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: %

Fig. 7: Excerpt from on-line questionnaire for a question about the criterion completeness

The CBR-PEB evaluation led to meaningful metrics and measurement data. This was demonstrated by the fact that the users’ representatives could interpret the analyzed data in a feedback session and that they were very satisfied with the results of the evaluation. The evaluation showed them that CBR-PEB performed better than they had expected.

Goal: Analyze the retrieved information for the purpose of monitoring economic utility with respect to the CBR system developers 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. number of cases in the EB 2. amount of background knowledge a. number of attributes (Q-8.1.1) [...] 3. 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. 8: Abstraction sheet for Goal 2 "Economic Utility"

5 GUIDING REPOSITORY SYSTEM IMPROVEMENT BY EVALUATION

Evaluation allows not only to justify the investment in building up and maintaining a reuse and experience repository system, but also provides opportunities for guiding its improvement. The presented GQM-based evaluation approach provides two major opportunities for guiding repository system improvement:

1. The feedback sessions allow to involve stakeholders in making evaluation result-based decisions about changes to the repository system.
2. The abstraction sheets allow to identify useful background knowledge for improvement/maintenance. Such background knowledge can be used as guidelines or rules for supporting decision making about improvement/maintenance (e.g., acquiring or updating experience items) as described in [3]. For example:
For EMS and the SFB-EB, usefulness depends on the file format of the artifacts (Fig. 4, variation hypothesis "o → 2"). Thus, only artifacts of usable formats should be stored in the repository. Artifacts of other formats are either rejected or converted into a usable format.
For CBR-PEB, descriptions of CBR systems with a high degree of maturity are preferred. Especially systems from industry are expected to have such a high maturity (Fig. 8, variation hypothesis 2, quality factor 2). Thus, the CBR-PEB maintenance team should mainly try to acquire descriptions of CBR systems from industry.

To further operationalize the idea of guiding repository improvement/maintenance by evaluation, a framework has been developed, which is described in [3]. One major component of the framework aims at assisting decision making about maintenance, which can use the guidelines or rules as presented above.

6 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we discussed an approach to iteratively conduct goal-oriented measurement programs for evaluating experience and reuse repository systems, like for instance, Basili's Experience Base (EB). Additionally, we outlined how evaluation can be exploited to support maintenance decision making to improve a repository system.

The suggested evaluation approach is built on three parts:

1. The Goal-Question-Metric (GQM) method, which is used as the methodological framework for

planning and conducting such measurement programs.

2. A high-level quality model, consisting of three overlapping phases, for the definition of meaningful goals and metrics depending on the maturity level of the repository.
3. A technique for iterative refinement of evaluation programs based on GQM-based measurement.

Based on the idea that measurement activities should be kept small in the beginning, our approach helps to narrow the scope of an initial evaluation program, but still is flexible enough to get more detailed –if necessary– in later phases. Two case studies (Section 4) exemplify this refinement idea for concrete measurement goals.

However, the examples depicted can only be a starting point for a more detailed evaluation of our approach. First, we have to develop a ‘construction set’ with measurement goals and suitable measures for each of the three phases (regarding the maturity level of the repository). Especially for the last phase we still miss some practical experience. Second, we are planning to integrate a role concept regarding the responsibilities for parts of the measurement program in different phases. Repository developers, for example, should be the driving forces for the definition of quality factors of the measurement program in the first phase (the prototypical use and set-up of the repository), while in later phases, when the quality focus is placed on the repository’s utility and economic value, it is more likely that the stakeholders become the driving force for such measurement activities.

ACKNOWLEDGMENTS

We would like to thank our colleagues from the University of Kaiserslautern and the Fraunhofer Institute for Experimental Software Engineering (IESE) for helpful discussions during the conception of this paper. We also would like to thank the people who participated in the evaluation programs. Carsten Tautz’ review of a previous version of this paper and Sonnhild Namingha’s help in improving the paper’s readability is gratefully acknowledged. Last but not least, our gratitude is extended to Prof. Dr. H. D. Rombach. Note that part of this work has been conducted in the context of the Sonderforschungsbereich 501 “Development of Large Systems with Generic Methods” (SFB 501).

REFERENCES

- [1] A. Abecker, S. Decker, and O. Kühn. Organizational memory (in German). *Informatik-Spektrum*, 21(4):213–214, 1998.
- [2] K.-D. Althoff, A. Birk, S. Hartkopf, W. Müller, M. Nick, D. Surmann, and C. Tautz. Managing software engineering experience for comprehensive reuse. In *Proc. of the 11th Int. Conference on Software Engineering and Knowledge Engineering (SEKE’99)*, Kaiserslautern, Germany, June 1999.
- [3] K.-D. Althoff and M. Nick. Systematic Evaluation and Maintenance of Experience Bases. *Workshop on Flexible Strategies for Maintaining Knowledge Containers (at ECAI’2000)*. Berlin, Germany, 2000.
- [4] V. R. Basili, G. Caldiera, and H. D. Rombach. Experience Factory. In J. J. Marciniak (ed.), *Encyclopedia of Software Engineering*, Volume 1, John Wiley & Sons, 1994.
- [5] V. R. Basili, G. Caldiera, and H. D. Rombach. Goal Question Metric Paradigm. In J. J. Marciniak (ed.), *Encyclopedia of Software Engineering*, Volume 1, John Wiley & Sons, 1994.
- [6] V. R. Benjamins, D. Fensel, C. Pierret-Golbreich, E. Motta, R. Studer, B. Wielinga, and M.-C. Rousset. Making knowledge engineering technology work. In *Proc. of the 9th Conference on Software Engineering and Knowledge Engineering*, pages 18–20, Madrid, Spain, 1997.
- [7] L. C. Briand, C. M. Differding, and H. D. Rombach. Practical guidelines for measurement-based process improvement. In *Software Process*, 2(4):253–280, December 1996.
- [8] W. S. Cooper. On selecting a measure of retrieval effectiveness. In K. Jones and P. Willet (eds.), *Readings in Information Retrieval*, pages 191–204. Morgan Kaufmann Publishers, 1997.
- [9] R. L. Feldmann. Developing a Tailored Reuse Repository Structure -Experience and First Results-. In *Proc. of the SEKE’99 Int. Workshop on Learning Software Organizations (LSO’99)*, Kaiserslautern, Germany, June 16, 1999. Online at: http://www.iese.fhg.de/SEKE99/pdf_files/Proceedings-LSO.pdf.
- [10] R. L. Feldmann, B. Geppert, W. Mahnke, N. Ritter, and F. Rößler. An ORDBMS-based Reuse Repository Supporting the Quality Improvement Paradigm -Exemplified by the SDL-Pattern Approach-. To appear in *Proc. of TOOLS USA 2000 - 34th Int. Conference & Exhibition*, Santa Barbara, USA, July / August 2000.

- [11] F. Houdek and K. Schneider. Software Experience Center: The Evolution of the Experience Factory Concept. In Proc. of the 24th annual Software Engineering Workshop (SWE24), Greenbelt, Maryland, USA, December 1-2, 1999. Online at: <http://sel.gsfc.nasa.gov/sew/1999/program.html>.
- [12] International Organization for Standardization (ISO). ISO 9000 Standards. Geneva, 1994.
- [13] C. Johansson, P. Hall, and M. Coquard. Talk to Paula and Peter - They are Experienced. In Proc. of the SEKE '99 International Workshop on Learning Software Organizations (LSO '99), Kaiserslautern, Germany, June 16, 1999. Online at: http://www.iese.fhg.de/SEKE99/pdf_files/Proceedings-LSO.pdf.
- [14] S. Kirchhoff. Abbildungsqualität von wissensbasierten Systemen: eine Methodologie zur Evaluierung (in German). Verlag Josef Eul, Bergisch Gladbach, Germany, 1994.
- [15] A. Koennecker, R. Jeffery, and G. Low. Lessons Learned from the Failure of an Experience Base Initiative Using Bottom-up Development Paradigm. In Proc. of the 24th annual Software Engineering Workshop (SWE24), Greenbelt, Maryland, USA, December 1-2, 1999. Online at: <http://sel.gsfc.nasa.gov/sew/1999/program.html>.
- [16] M. Mendonça and B. Aydinlioglu. The Experience Management System: principles, design, and implementation. University of Maryland, College Park, MD, USA. Internal report in progress.
- [17] M. Mendonça, M. Zolkowitz, M. Lindval, M. Habetz, M. Frey. The Experience Management System II: principles, requirements, and initial design. University of Maryland/Fraunhofer Center Maryland, College Park, MD, USA. Internal report in progress.
- [18] T. Menzies. Evaluation issues for problem solving methods. In Proc. of the 11th Workshop on Knowledge Acquisition, Modeling and Management. Banff, Canada, 1998.
- [19] T. Menzies. Requirements for good measurements. 1998. Online at: <http://www.cse.unsw.edu.au/timm/pub/eval/cautions.html>.
- [20] M. Nick, K.-D. Althoff, and C. Tautz. Facilitating the practical evaluation of organizational memories using the goal-question-metric technique. In Proc. of the 12th Workshop on Knowledge Acquisition, Modeling and Management, Banff, Alberta, Canada, October 1999.
- [21] M. Nick. Implementation and Evaluation of an Experience Base. Diploma thesis, Fraunhofer IESE, University of Kaiserslautern, 1998.
- [22] H. D. Rombach. Practical benefits of goal-oriented measurement. In Proc. of the Annual Workshop of the Centre for Software Reliability, pages 217–235. Elsevier, September 1990.
- [23] T. Rout. SPICE: A framework for software process assessment. Software-Process-Improvement and Practice, Pilot Issue, pages 57–66, 1995.
- [24] C. Seaman, M. Mendonça, V. R. Basili, and Y.-M. Kim. An Experience Management System for a Software Consulting Organization. In Proc. of the 24th annual Software Engineering Workshop (SWE24), Greenbelt, Maryland, USA, December 1-2, 1999. Online at: <http://sel.gsfc.nasa.gov/sew/1999/program.html>.
- [25] B. Smyth, and E. McKenna. Modelling the competence of case-bases. In B. Smyth and P. Cunningham (eds.), *Advances in Case-Based Reasoning: Proc. of the 4th European Workshop on Case-Based Reasoning*, pages 196–207. Springer-Verlag, Berlin, Germany, September 1998.
- [26] Software Engineering Institute. The Capability Maturity Model: Guidelines for Improving the Software Process. Carnegie Mellon University, Addison-Wesley, 1995.
- [27] C. Tautz and K.-D. Althoff. A case study on engineering ontologies and related processes for sharing software engineering experience. To appear in Proc. of the 12th Conference on Software Engineering and Knowledge Engineering (SEKE 2000), Chicago, USA, July 2000.
- [28] F. van Harmelen. Notes from evaluation working group at Banff'98. 1998. Online at: <http://www.cs.vu.nl/frankh/spool/eval.html>.
- [29] R. Webby, C. Seaman, M. Mendonça, V. R. Basili, and Y.-M. Kim. Implementing an Internet-Enabled Software Experience Factory: Work in Progress. In Proc. of the ICSE 99 Workshop on "Software Engineering over the Internet", Los Angeles, California, USA, May 17, 1999. Online at: <http://sern.cpsc.ucalgary.ca/~maurer/ICSE99WS/ICSE99WS.html>.