Supporting the Adaptation of Software Quality Models – An Empirical Investigation

Constanza Lampasona, Michael Kläs

Fraunhofer Institute for Experimental Software Engineering
Fraunhofer-Platz-1, 69663 Kaiserslautern, Germany
{constanza.lampasona, michael.klaes}@iese.fraunhofer.de

Abstract. Measuring and evaluating software quality are fundamental challenges for software engineers. Assessing software quality is hard to accomplish in practice not only because existing quality models are not easily applicable, but also because adjusting them to the needs of one's own organization and projects requires intensive effort. In this paper, we present a study on applying a flexible but rigorous adaptation process for quality models. The goal-oriented adaptation process is based on the existence of a quality meta-model that provides a structure for adapted models. In order to obtain first empirical insights, we compare adaptations guided by the tool-supported adaptation process with (ad-hoc) adaptations using a tool that can be used to create and edit quality models. In the study, we investigated the *formal consistency*, *appropriateness*, and *efficiency* of exemplary quality model adaptations. One important study result is that the quality models obtained applying the tool-supported process are considerably more *consistently* and *appropriately* adapted than the ones obtained by following an ad-hoc approach. Further, we could observe that model adaptation is significantly more *efficient* when applying the adaptation process.

Keywords. Empirical study, goal-oriented adaptation, tailoring process.

1 Introduction

The measurement and evaluation of quality are fundamental challenges for software engineers. A lot of models exist that concretize the concept of software quality by defining sub-concepts that are parts of software quality. This refinement typically results in tree-like structures of concepts and sub-concepts. The leaves of the tree are (or are supposed to be) measurable concepts that contribute to software quality.

Although existing models can help to systematically concretize the concept of quality, they also show deficiencies. Some are very specific, which limits the scope of their applicability. Others are abstract, which leads to a large overhead when concretizing them in order to make them applicable. Moreover, concretizing a quality model often requires intensive expert effort. This is especially true since quality models are rarely provided together with a detailed process or method for their adaptation.

Building quality models based on adapting a core model for specific domains and specific purposes [14] may reduce these problems by achieving a balance between fixed models and models developed from scratch. Relying on this concept of *balanced* models, we propose a method for adapting quality models, including a general adaptation process based on a structure defined by a quality meta-model [12].

Many existing quality models are adaptations of other models, e.g., [2, 5, 1, 4, 10]; however, these customizations are narrowly focused and difficult to transfer to other contexts. Franch and Carvallo [9] present a general process for building an ISO9126 model. Plösch et al. [16] present a method for adapting operational quality models.

We have formerly discussed a potential solution idea for performing goal-oriented, efficient adaptations and obtaining correctly adapted quality models [13]. In this contribution, we present the general process for adapting quality models, which is a core part of our adaptation method, and a study in which we investigated the impact of the implemented adaptation process with regard to (1) the efficiency of adaptation and the (2) formal consistency and (3) appropriateness of the quality models obtained by applying the process.

In the following sections, we provide an overview of the adaptation process and describe the study design. Then, we present and analyze the study's results. Finally, we summarize our current work and sketch planned research directions.

2 Goal-oriented Adaptation of Software Quality Models

Balanced models are a fundamental concept in the German research project Quamoco (www.quamoco.de) in which our work has been conducted. The project aims at developing a software quality standard with easily operationalizable models to cover different technologies for software development. In order to arrive at an operationalized quality model applicable for a concrete environment in a repeatable way, we developed an adaptation process for quality models.

In this chapter, we describe the scope of quality model adaptation and the general steps needed to achieve it, such as identifying an adequate reference model to be customized and the necessary changes to be performed (what has to be modified, when and how).

2.1 Scoping Quality Model Adaptation

Software quality models may exist and be applied at different levels:

Public level: The models at this level are universally available, they may be intended for general use (e.g., ISO9126 [11], Quamoco base model), or for some specific domain (e.g., IEC 61508 [6] for safety in embedded systems, EN 60601-1-4 [7] for medical device embedded systems). This level is comparable to the broad *industry level* described by Fitzgerald [8]. Most of the models at this level are very generic; they are usually not operational and need to be customized. Using and tailoring these models could be useful for showing adherence to some standard.

Organization level: At this level, quality models focus on satisfying the interests of a specific organization. They can focus on the whole organization, a business unit, or a project portfolio. They are typically more specific than quality models at the public level and intended to provide a common basis for project-specific model tailoring. At this level, public models can be refined for a particular organization and organization models can be further refined for specific parts of the organization.

Project level: At this level, quality models are put into operation; they are applied to specify and assess quality. Here, quality model adaptation should be limited to minor adjustments driven by project-specific quality requirements, without drastically changing the structure of the organization's quality model. This helps to preserve the comparability of quality evaluations across software products. At the project level, organization-wide models are further refined for a particular project.

The reuse of a quality model for adaptation is, in essence, more efficient than creating a new model from scratch for each project. We recommend tailoring models stepwise: for organizations and for projects. This means that the adaptation process proposed here can be used to adapt a public model for an organization and then to refine that organization model for specific project needs.

2.2 Adapting a Quality Model

The main steps for adapting a quality model are: (1) identifying a reference quality model as a basis for the adaptation, (2) sorting out irrelevant content, (3) performing adjustments, and (4) testing the adapted quality model.

2.2.1 Identifying a Reference Quality Model

The first thing to be done is to define the goal of the quality model that should result from the adaptation.

In order to define the goal, the organization/project needs with respect to software quality and context information are used. That is, it is necessary to identify the circumstances under which the quality model will be used. In order to describe the goal in a structured way and not to forget important aspects, we use an adapted GQM goal template, which is typically used to define measurement goals [3]: (1) *Object*, (2) *Purpose*, (3) *Viewpoint*, (4) *Quality Focus*, and (5) *Context*.

To define a goal, these questions must be answered:

1. ***What are the elements that are used to define, measure, or assess the product quality?*** For example, product documentation, source code, requirements, design, build process, test suite, etc. This information is the *object* in the goal.
2. ***For which purpose do I need the quality model?*** Following the classification of application purposes for quality models proposed in [16], the current Quamoco meta-model considers two different *purposes*: specification and evaluation of quality. Specification means that quality is described, but neither quantified nor measured. For the purpose of evaluation, quality is quantified, measured, and compared to defined assessment criteria to check the fulfillment of those criteria.
3. ***From which perspective is quality described or evaluated?*** Are there specific management requirements? What are the agreements with the customer? Must practices established in the organization be considered?
4. ***Which quality attributes of the software product are covered with this model?*** Quality can focus on general properties, such as reliability or maintainability, or specific aspects can be considered, such as globalization, learnability, or training.

5. ***What is the context of the software products to be addressed by the model?*** *Context* may include many different things. Are there things that are mandatory within the organization? Which domain should be covered by the quality model (e.g., railway, medical devices, embedded systems, information systems)? Which methodologies, practices, or technologies should be supported (e.g., component-based software development, agile development, open-source software, custom development, C++, Java, or automatic measurement tools)?

The answers to these questions together describe the goal of the required quality model, which will make it easier to focus on the key elements of the adapted quality model. The documented goal can be used later, for example when the model is inspected, to corroborate that it actually fulfills the stated goal.

Now the goal is used to look for a model and adapt it to the needs of the project or organization. We call this model on which the model adaptation is based, the reference model. Finding the right reference model consists in finding the model whose attributes best fit to the defined goal.

2.2.2 Sorting out Irrelevant Information

Once a reference model is chosen, the actual adaptation can start. First, elements are discarded that are not needed in the final model. Only quality model components in the reference model that are relevant for the new model are taken. In this way, unnecessary components of the quality model are eliminated at the beginning. Such components may be used for specific perspectives, such as the management view on quality, or for quality aspects such as internationalization, which are not of interest in the model. They can also be used for artifacts such as user documentation or design that should not be considered in the model, or for measures that cannot be collected since they are not applicable in the context of the model, e.g., measures for Java code in a model for applications in C. Sometimes, specific elements in the model can be partially reused but need some adjustments. Such elements should stay in the model and be marked for detailed inspection and modification in the next step. The parts selected to remain in the model are the basis for further adjustments.

2.2.3 Further Adjustments

After sorting out irrelevant information, the model obtained might not be consistent or operational anymore. The removal of model components triggers further adaptation tasks. These tasks help to bring the model back to a consistent, operational state. Some adaptation tasks can be automated. Other tasks will require user interaction, as they are based on user decisions.

Accomplishing all adaptation tasks will lead to a consistent model customized to the user's needs. Elements are incrementally deleted, added, or modified in the model until no further adaptation tasks are requested. The extent to which these operations are used depends on the suitability of the reference model. At this point, the quality model has been successfully adapted and can satisfy the defined goal.

New elements can be defined and added to the model or elements from models can be added to the adapted model. That is, individual elements from other models can be taken and reused in the new model.

What should be documented?

Goal of adapted quality model: The goal is a compact manageable description of the quality model. If the appropriateness of the model with respect to the goal is put into question during tailoring, the need to complete the model in order to archive the goal must be documented as well as the fact that the model is not complete.

Deviations from reference model: Operations on elements that are used to be mandatory for the organization and are no longer being considered in the sub-organization or project and the reasons of non-inclusion must be listed. Eventually, an agreement should be signed approving these changes. The manager responsible for the organizational model can take these changes and their justifications as one source for changes when maintaining the organization's quality models.

2.2.4 Testing the Adapted Quality Model

In this step, the adapted quality model needs to be applied to a small sample of test objects, i.e., the adapted model is piloted. This will lead to final acceptance of the model or indicate the need for further adaptations. Performing this step is mandatory when adapting a quality model, but how to perform it in detail is not in the scope of the adaption process.

3 Study Goals, Design, and Performance

The adaptation process as the object of the investigation was implemented as an add-on extending the functionality of an existing quality model editor. In the study, we compared quality model adaptation using only the quality model editor (Editor) and quality model adaptation applying the Quamoco adaptation process implemented by the add-on (Adaptation Assistant).

3.1 Study Goals

In the study, we wanted to investigate the quality of the proposed Quamoco adaptation method; in particular, the following question should be answered: 'Does the implemented adaptation process support its three major goals?'

- *Formal Quality Model Consistency*: Adapted quality models are syntactically correct, i.e., they conform to the structure defined by the quality meta-model.
- *Quality Model Appropriateness*: Adapted quality models are correct and complete with respect to their goals, i.e., they are suitable for use with the Object, Purpose, Viewpoint, Quality Focus and Context.
- *Efficiency of Adaptation*: The adaptation of quality models can be performed in an effort-efficient manner.

3.2 Operationalization

We collected subjective judgments about the achievement of the three major goals associated with the adaptation method by asking the participants closed questions:

Perceived_consistency: Does the participant consider the quality model obtained to be syntactically correct? This is the subjective assessment by the participants of *formal quality model consistency.*

Perceived_appropriateness: Does the participant consider the quality model obtained to be appropriate with respect to its goal? (i.e., the model is complete and correct with respect to its goal). This is the subjective assessment by the participants of *quality model appropriateness.*

Perceived_efficiency: Does the participant think that the adaptation can be performed efficiently? This is the subjective assessment by the participants of e*fficiency of adaptation.*

For these variables, we used a 7-point Likert scale: {1: strongly disagree, 2: disagree, 3: somewhat disagree, 4: neither agree nor disagree, 5: somewhat agree, 6: agree, 7: strongly agree}. Additionally, we allowed the answer "I do not know".

Besides evaluating the goals based on the perception of the participants, we also want to evaluate them in a more objective way. Since it is difficult to objectively determine the degrees to which the three major goals are fulfilled directly, we address them indirectly by identifying the minimum set of model elements that need to be adapted (i.e., added, modified, or deleted) in order to obtain a consistently and appropriately adapted quality model. This allows us to define measures for the *completeness* and *correctness* of the performed adaption and use the measurement results as a more objective indicator for the model's *consistency* and *appropriateness*: a more completely and correctly adapted model is more consistent and appropriate.

Completeness: We say that a quality model is completely adapted if all of its elements are adapted that needed to be adapted to obtain a model that is consistent with the structure described by the meta-model and appropriate for addressing its goal. We measure this concept using two base measures: the total number of elements that should be adapted in the quality model based on the provided adaptation scenario and the number of elements in the quality model that were adapted by the study participant:

$$\text{completeness} = \frac{\text{number of adapted elements that should be adapted}}{\text{number of elements that should be adapted}}$$

Correctness: We say that a quality model is correctly adapted if all of its elements that should be adapted are correctly adapted with respect to the quality model goal. This means that we measure the degree of correctness as the percentage of correctly adapted elements with respect to the quality model goal:

$$\text{correctness} = \frac{\text{number of correctly adapted elements}}{\text{number of elements that should be adapted}}$$

Efficiency: We measured efficiency in a more objective way by relating the number of correctly adapted elements and the time needed for the adaptation:

$$\text{efficiency} = \frac{\text{number of correctly adapted elements}}{\text{time required for adaptation}}$$

3.3 Hypotheses

During the study, we tested the following hypotheses:

H_{Sub1} (Perceived consistency): The participants consider the quality models obtained using the Adaptation Assistant syntactically more correct than the quality models obtained using the Editor:

H_{1Sub}: μ(perceived_consistency(AA)) > μ(p_con(E)), i.e., H_0: μ(p_con(AA)) ≤ μ(p_con(E))

H_{Sub2} (Perceived appropriateness): The participants consider the quality models obtained using the Adaptation Assistant more complete and correct with respect to their goals than the quality models obtained using the Editor:

H_{2Sub}: μ(perceived_appropriateness(AA)) > μ(p_app(E)), i.e., H_0: μ(p_app(AA)) ≤ μ(p_app(E))

H_{Sub3} (Perceived efficiency): The participants consider the adaptation to have been more efficiently performed using the Adaptation Assistant than using the Editor:

H_{3Sub}: μ(perceived_efficiency(AA)) > μ(p_eff(E)), i.e., H_0: μ(p_eff(AA)) ≤ μ(p_eff(E))

H_1 (Completeness): The adapted quality models obtained using the Adaptation Assistant (AA) are more completely adapted than the adapted quality models obtained using the Editor (E):

H_1: μ(completeness(AA)) > μ(completeness(E)) , i.e., H_0: μ(comp(AA)) ≤ μ(comp(E))

H_2 (Correctness): The adapted quality models obtained using the Adaptation Assistant are more correctly adapted than the adapted quality models obtained using the Editor:

H_2: μ(correctness(AA)) > μ(correctness(E)) , i.e., H_0: μ(corr(AA)) ≤ μ(corr(E))

H_3 (Efficiency): Quality model adaptation is more efficiently performed when using the Adaptation Assistant than when using the Editor:

H_3: μ(efficiency(AA)) > μ(efficiency(E)), i.e., H_0: μ(eff(AA)) ≤ μ(eff(E))

3.4 Participants and Context

The target population comprises people working as software quality managers in a company or in similar positions where part of their job is to adapt, set up, or maintain software quality models.

We conducted the study during one of the Quamoco project workshops. The participants were partners of the Quamoco consortium experienced in working with quality models. In addition, they had experience with the Quamoco meta-model and the corresponding *Editor*. They had only rudimental knowledge regarding the Quamoco adaptation process and no experience with the *Adaptation Assistant*.

To prepare the participants for the study, we presented the adaptation process together with examples. After that, we introduced the functionality and use of the Adaptation Assistant as well as an example adaptation using the tool.

3.5 Materials and Procedures

For the study, we provided the participants with the following input:

- Two quality model application goals that should be used by the participants to find the most appropriate reference model.
- Two pools of quality models from which the most appropriate reference model should be selected by the participants on paper and in the adaptation tool.
- Two adaptation scenarios including practical adaptation task descriptions.
- Two example quality models that should be adapted by the participants.

During the study, the participants assumed the role of a software quality manager in the scenarios and were asked to perform the following tasks:

- *Finding the most suitable reference model*: The participants had to select a reference model from a pool of quality models for a defined goal of quality model application. Most suitable means that this model meets most of the concepts required by the quality model goal.
- *Producing an adapted quality model*: The participants had to execute the practical adaptation tasks.

These tasks were performed by the participants twice: once with one scenario and the Editor and once with a second scenario and the Adaptation Assistant. We chose this kind of cross-design with two different adaptation scenarios (Table 1) in order to deal with the low number of participants, but get the design-inherent learning effects low.

After each adaptation, the participants provided their feedback by filling out a questionnaire, which asked them to subjectively rate the formal consistency of the quality model, the appropriateness of the obtained quality model, and the efficiency of the adaptation (for the Adaptation Assistant and for the Editor, respectively). After the execution of both scenarios, the entire work-space of each participant was collected and saved for subsequent analysis.

Table 1: Study Design

	Editor	Adaptation Assistant
Group 1*	Adaptation Scenario A	Adaptation Scenario B
Group 2*	Adaptation Scenario B	Adaptation Scenario A

*Both groups had the same number of randomly assigned participants.

4 Study Results and Interpretation

Descriptive Statistics: Table 2 shows the mean, median, and standard deviations (stdev) for the eight adaptations performed during the study, separated into applications of the Editor (baseline) and the Adaptation Assistant:

Table 2: Study Results

	Editor			Adaptation Assistant		
	mean	median	stdev	**mean**	median	stdev
Completeness (in %)	**15.00**	15.78	6.76	**78.55**	76.52	7.46
Correctness (in %)	**8.93**	9.98	3.87	**70.34**	69.17	8.82
Efficiency (elements/minute)	**0.37**	0.41	0.18	**2.90**	2.84	0.54
Perceived Consistency*	**3.50**	3.50	2.38	**6.00**	6.00	0.82
Perceived Appropriateness*	**2.00**	2.00	0.82	**5.75**	6.00	0.50
Perceived Efficiency*	**1.25**	1.00	0.50	**5.75**	5.50	0.96

*measured using a 7-point Likert scale with 1: strongly disagree, 2: disagree, 3: somewhat disagree, 4: neither agree nor disagree, 5: somewhat agree, 6: agree, 7: strongly agree.

Hypotheses*:* As our sample is not large enough to assume a normal distribution, we applied non-parametric one-sided Wilcoxon signed-rank tests with alpha=0.05.

- H_{Sub1}: perceived_consistency(AA) > p_con(E) was *accepted* with p=0.032,
- H_{Sub2}: perceived_appropriateness(AA) > p_app(E) was *accepted* with p=0.033,
- H_{Sub3}: perceived_efficiency(AA) > p_eff (E) was *accepted* with p=0.034,
- H_1: completeness(AA) > completeness(E) was *accepted* with p=0.034,
- H_2: correctness(AA) > correctness(E) was *accepted* with p=0.034,
- H_3: efficiency(AA) > efficiency(E) was *accepted* with p=0.034.

Threats to validity: The two major threats to the validity of our results are the small sample size and the potential learning effects.

Small sample size: The participants were chosen due to their experience in quality modeling in general and with the quality meta-model as well as with the Editor in particular. Therefore, they are (although a convenience sample) more representative of the target population (i.e., professionals performing quality model adaptations as part of their job) than, for example, students studying computer science. However, these selection criteria led to a small number of participants who executed the scenarios.

Potential learning effects: Although the participants were not requested to follow a particular process for adapting the first model using the Editor and were confronted with two different adaptation scenarios, they may have learned from the first adaptation, which may have positively influenced their performance during the second adaptation using the Adaptation Assistant.

Further threats are that only a limited time frame was available for the study participants to conduct the adaptation tasks and that the attitude of the participants toward the well-known Editor or the Adaptation Assistant may have influenced the result.

Interpretation: Not only could hypotheses H_1 to H_3 be accepted, but the magnitude of the improvement using the tool-supported Quamoco adaptation process also seems to be high when compared to performing the adaption without explicit adaptation support using only the *Editor*. The effect was perceived by the participants and could be measured by analyzing the adapted models. Therefore, although several threats to the study's validity exist, we conclude that the proposed tool-supported adaptation process can increase the efficiency of adaptation tasks and the quality of their results in terms of consistent and appropriate models. Further, the study results indicate that typical quality model adaptations are difficult to handle adequately without a tool-supported adaptation process. The main reason for these results appears to be that even at first glance manageable adaptation tasks result in many subsequent sub-tasks that must be performed in order to assure the completeness and correctness of the adapted model. In part, these sub-tasks are hard to identify without support due to the complexity of a typical quality model and even harder to remember until they can be resolved due to their large number, especially if there is no process providing guidelines through the adaptation.

5 Summary and Future Work

Adapting models is important to get quality models that fit the specific needs of a concrete environment without building each model from scratch. However, the adaptation of a quality model is a complex and therefore error-prone task. Our study indicates that the quality of the adapted model can be significantly improved when using a well-defined and tool-supported adaptation processes such as the one developed in the Quamoco project. Not only were the consistency and appropriateness of the adapted quality model significantly improved, but so was the efficiency of performing the adaptation tasks.

In a next step, the adaptation method including the adaptation process and the rules for identifying the required adjustment tasks should be transferred to an updated quality model structure and get evaluated in an industrial field study in order to ensure its applicability for quality models and adaptation tasks in practice.

Acknowledgments

Parts of this work have been funded by the BMBF project Quamoco (grant no. 01 IS 08 023 C). We gratefully acknowledge Jens Göddel for his contributions.

References

1. Andersson, T.; Eriksson, I. V. (1996): Modeling the quality needs of an organization's software. In: HICSS '96: Proceedings of the 29th Hawaii International Conference on System Sciences Volume 4: Organizational Systems and Technology. Washington, DC, USA: IEEE Computer Society, p. 139.
2. Andreou, A. S.; Tziakouris, M. (2007): A quality framework for developing and evaluating original software components. In: Inf. Softw. Technol., vol. 49, no. 2, pp. 122–141.
3. Basili, V.; Weiss, D. (1984): A methodology for collecting valid software engineering data. In: IEEE Transactions on Software Engineering, vol. 10(3), pp. 728-738.
4. Bianchi, A.; Caivano, D.; Visaggio, G. (2002): Quality models reuse: experimentation on field. In: COMPSAC '02: Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment. Washington, DC, USA: IEEE Computer Society, pp. 535–540.
5. Calero, C.; Cachero, C.; Córdoba, J.; Moraga, M. (2007): PQM vs. BPQM: studying the tailoring of a general quality model to a specific domain. In: Advances in Conceptual Modeling – Foundations and Applications, pp. 192–201.
6. E DIN IEC 61508-3:2006-07: Functional safety of electrical/electronic/programmable electronic safety-related systems.
7. EN 60601-1-4:1999: Medical electrical equipment - Part 1-4: General requirements for safety - Collateral standard: Programmable electrical medical systems.
8. Fitzgerald, B.; Russo, N. L.; O'Kane, T. (2003): Software development method tailoring at Motorola. In: Commun. ACM, vol. 46, no. 4, pp. 64-70.
9. Franch, X.; Carvallo, J. P. (2003): Using quality models in software package selection. In: IEEE Softw., vol. 20, no. 1, pp. 34–41.
10. Horgan, G.; Khaddaj, S. (2009): Use of an adaptable quality model approach in a production support environment. In: Journal of Systems and Software, vol. 82, no. 4, pp. 730–738.
11. ISO/IEC 9126-1:2001: Software Engineering - Product Quality - Part 1: Quality Model.
12. Kläs, M.; Lampasona, C.; Nunnenmacher, S.; Wagner, S.; Herrmannsdörfer, M.; and Lochmann; K. (2010): How-to evaluate meta-models for software quality? In: Abran, A.; Büren, G.; Dumke, R.R.; Cuadrado-Gallego, J.J.; Münch, J.: Applied Software Measurement - Proceedings of the joined International Conferences on Software Measurement (IWSM/MetriKon/Mensura 2010); pp. 443-462.
13. Kläs, M.; Lampasona, C.; Trendowicz, A.; Münch, J. (2009): Goal-oriented adaptation of software quality models. In: Technische Universität München. Institut für Informatik: Tagungsband 3. Workshop zur Software-Qualitätsmodellierung und -bewertung. SQMB' 10. München, 2010; pp. 4-11.
14. Kläs, M.; Münch, J. (2008): Balancing upfront definition and customization of quality models. In: Technische Universität München. Institut für Informatik: Software-Qualitätsmodellierung und -bewertung. SQMB'08 - Workshop-Band. München, 2008; pp. 26-30.
15. Kläs, Michael ; Heidrich, Jens ; Münch, Jürgen ; Trendowicz, Adam:
CQML Scheme: A Classification Scheme for Comprehensive Quality Model Landscape. In: EUROMICRO 2009. Proceedings of the 35th EUROMICRO Conference Software Engineering and Advanced Applications. Los Alamitos: IEEE Computer Society, 2009, 243-250: Ill., Lit.
16. Plösch, R.; Gruber, H.; Körner, C.; Pomberger, G.; Schiffer, S. (2010): Adapting quality models for assessments - Concepts and tool support. In: Proceedings of SQMB 2010 Workshop, held in conjunction with SE 2010 conference, February 22nd 2010, Paderborn, Germany, published as Technical Report TUM-I1001 of the TUM.