Fraunhofer IESE

Application Scenarios and Guidelines for Quality Model Adaptation

Authors:
Constanza Lampasona
Michael Kläs

IESE-Report No. 102.12/E
Version 1.0
November 2012

A publication by Fraunhofer IESE

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

Fraunhofer IESE is directed by
Prof. Dr. Dieter Rombach
(Executive Director)
Prof. Dr.-Ing. Peter Liggesmeyer
(Scientific Director)
Fraunhofer-Platz 1
67663 Kaiserslautern
Germany

Abstract

This paper deals with the adaptation of software quality models. We have observed that for quality models of realistic size, adaptation is a complex and error-prone task. Therefore, we identified in a small survey the most relevant scenarios for the adaptation of quality models and derived concrete guidelines to support quality managers in dealing with these scenarios.

Keywords: Software quality, quality models, adaptation, tailoring, application scenarios

Table of Contents

1 Challenge and Context

Software quality models are applied in many companies as a tool for describing, assessing, or predicting the quality of software artifacts. A large number of quality models have been proposed [10]. Many of them specify a prescriptive set of quality characteristics or metrics. Even though ISO25010 [7] attempts to provide some standardization through a structure of quality attributes and sub-attributes, most companies use their own quality models [18][19].

Methods for adapting quality models should be efficient and produce consistent models. Adapting quality models rigorously is very important for obtaining models that fit the needs of a concrete application context without the need to build a new model from scratch every time [17]. However, in many cases the adaptation of a quality model is a complex and error-prone task.

In previous work [11], we addressed this challenge by developing a flexible but rigorous approach for adapting quality models under the assumption that they conform to a principal structure provided by a meta-model. The method addresses the need for efficiently adapting quality models in a way that results in consistent and appropriate models. The consistency of the adapted quality model is covered by the definition of elementary adaptation operations and corresponding consistency rules.

One major challenge regarding the definition of such a quality adaptation method is to make it as independent as possible of a particular model and type of adaptation, i.e., to define a set of adaptation rules that will be universally applicable to any model and adaptation scenario. However, to be applied efficiently in practice, the method needs to be accompanied by practical application guidelines, including concrete application scenarios with corresponding step-by-step instructions.

In order to do this, an elaborated list of relevant adaptation scenarios has to be developed. Adaptation scenarios are important because they focus the adaptation process on a specific situation. Different kinds of adaptations are triggered by different reasons, for example, if measurement data from a new tool needs to be included, or if an existing model needs to be customized for a different software development paradigm.

This paper extends the work presented by the authors at different workshops and conferences ([12][14][11]) and focuses on the practical application of the method, i.e., adaptation scenarios and guidelines. In particular, two research questions shall be analyzed:

RQ1: Are specific types of adaptation scenarios more relevant than others?

RQ2: Are adaptations focusing on more concrete concepts in a quality model considered more relevant than changes to more abstract concepts?

In the following, we will provide a short overview of related work in quality model adaptation. Then, we will show how we elicited and validated the application scenarios and present guidelines for practical use.

2 Related Work

Approaches to the adaptation of quality models mainly focus on customizing the quality standard ISO9126, as in [15] and [2], where quality characteristics or domain-specific features are added. Other approaches, such as in [1], [3], and [13], try to refine goals or factors into metrics using the factor-criteria-metric or the goal-question-metric approaches. They again use the characteristics defined in ISO9126 and its successor ISO25010 as a starting point. The quality model adaptation presented in [9] relies exclusively on decisions made by experts.

More advanced approaches are presented in [5] and [16]. Unfortunately, the guidelines proposed by [5] are only rough and difficult to operationalize for adaptation in practice. Plösch et al. tailor a set of rules provided by static code analysis tools based on a set of criteria [16]. The idea consists of reducing a set of rules according to specific criteria. The approach has tool support and seems promising in terms of ensuring efficient adaptation, even though the scope of the model is limited.

Most of these approaches focus on only one adaptation scenario or do not explicitly name any scenario.

3 Elicitation and Validation of Application Scenarios

We propose a series of application scenarios for the adaptation of product quality models. We observed that an obvious scenario where quality models are being adapted or updated is quality model maintenance. We started studying this scenario and afterwards extended it to further uses.

3.1 Quality Model Maintenance Activities

The sustainable use of a quality model in a company requires its systematic maintenance. We can distinguish several types of quality model maintenance and classify them by adapting the definitions provided by the ISO/IEC standard for software maintenance (ISO/IEC 14764) [6]. Unlike for software, in the case of quality models we define *correction* as modification of the model to meet unchanged performance requirements in an unchanged context. Moreover, we define *enhancement* as modification of the model to meet either changed performance requirements or fit it into a changed application context. As a result, we distinguish four types of model maintenance activities:

- *Preventive maintenance* should already be realized during the model's development and introduction in a company, which includes its context-specific adaptation and instantiation.

- *Corrective maintenance* is the reactive modification of the quality model performed after its application in a certain project to improve unsatisfactory performance discovered during the course of the application.

- *Perfective maintenance* is the proactive modification of the quality model performed in the same application context to meet new, improved performance requirements.

- *Adaptive maintenance* is the proactive modification of the quality model performed to keep it usable in a changed (new) or changing (current) project context where it is applied.

3.2 Prioritization by Prospective Users

Based on the quality maintenance activities and on discussions with practitioners, with multiple rounds of reviews and adjustments of scenarios and their descriptions, we formulated a set of typical adaptation scenarios. Subsequently, we surveyed prospective users with experience in dealing with quality models regarding the specified scenarios in order to collect data for the analysis of their importance.

Ten people participated in the survey. We asked the participants – practitioners (4) as well as scientists (6), all experienced in the development of quality models for companies – to rate the specified scenarios on a 7-point Likert scale (1: very unimportant - 7: very important). In addition, we gave the respondents the opportunity to define and rate additional scenarios.

We proposed 16 different scenarios:

Q1.- New data sources: You have a quality model and want to update it in order to use a new or updated data source. For instance, an additional static code analysis tool not considered yet in your model should be used to provide additional information for evaluating software quality.

Q2.- New modeling/implementation languages: You have a quality model and want to use it with a new modeling or implementation language. E.g., your current model is applicable for C, but now you want to assess C++ software.

Q3.- New technologies or paradigms: You have a quality model and want to use it with a new paradigm or technology. For instance, your current model considers only software for single-core microcontrollers, but now you want to also consider multi-core microcontrollers, or you plan to introduce model-based development.

Q4.- New artifacts: You have a quality model and want to use it with new kinds of artifacts. For instance, your current model considers only the source code for assessing quality, but now you want to also consider the design documents.

Q5.- New domains: You have a model and want to use it in another application domain. For instance, a model elaborated for the information system domain should be adapted to be applicable in the service-oriented architecture domain.

Q6.- New/modified quality requirements: You have a quality model and want to integrate new/modified quality requirements. For instance, the adapted model should enforce lower code complexity in order to increase maintainability.

Q7.- New/updated perspectives: You have a quality model and want to integrate a new perspective or update an existing one, that is, you want to consider new quality aspects or change the importance of existing ones. For instance, your current model considers the end-user view, but now you want to assess quality also from the point of view of the operating IT department.

Q8.- Using a subset of data sources: You have a quality model with an elaborate set of measures, but for specific quality assessments, only certain measures can be collected due to limited resources. For instance, only a limited set of tools can be used or priorities have to be set according to cost of data collection, trustworthiness of collected metrics, or relevance of quality aspects.

Q9.- Focusing on specific modeling/implementation languages: Your current quality model addresses several modeling/implementation languages, but you want to limit your assessments to specific ones. For instance, the model covers both Java and C++, but now you want to assess only software written in Java or the Java-based parts of your software.

Q10.- Reduce technology or paradigm scope: Your current quality model addresses multiple paradigms and technologies, but you want to focus your assessments on specific ones. E.g., your current model considers procedural and object-oriented programming, but since the assessed products do not use OO concepts, you want to adapt the model to focus the assessment on procedural programming.

Q11.- Assessing a subset of artifacts: You have a quality model addressing multiple kinds of artifacts, but only a certain kind of artifacts should be assessed. For instance, the model covers both source code and design; however, at the current point in time, only the design documents are available and should be assessed.

Q12.- Focusing on actual domain: You have a model applicable in more than one domain, but for the quality assessments you want to consider the actual domain of the assessed product. For instance, a model applicable for the information and embedded system domains should be adapted to assess embedded systems.

Q13.- Modified project-specific quality requirements: You have a quality model and want to integrate new/modified project or assessment-specific quality requirements. For instance, for the assessed product, a lower coupling level than usual is required, which should be addressed by the adapted quality model.

Q14.- Performing assessments limited to specific perspectives: You have a quality model addressing multiple perspectives but only a certain perspective is important for the current assessment. For instance, the model covers the developer as well as the management point of view, but the current assessment should focus only on the management perspective.

Q15.- Fixing observed assessment problems: You have a quality model that delivered a faulty assessment result. After identifying the root cause of the wrong result, you want to adapt the model in order to correct it. E.g., you adapt the specification of a measure that delivers wrong results or correct a modeling mistake.

Q16.- Calibrating the model for future assessments: You have a quality model that you apply in quality assessments. After a series of assessments, you want to integrate the obtained knowledge into the model to improve the meaningfulness / significance of the assessments. For instance, you adapt the evaluation rules or target values in the model to fit better into your environment.

4 Survey Results and Analysis

For the data analysis, we built groups of similar scenarios. We distinguish between adaptations motivated by changes in the application context (AO, adaptive maintenance at the organizational level), adaptations representing an adjustment for a specific assessment (AP, adaptive maintenance at the project level), and adaptations representing a correction or improvement of the model without any change to its application context (CI, corrective and perfective maintenance).

The scenarios can also be arranged according to how concrete their focus is; from more concrete to more abstract, the focus can be: on data, measures and tools (A1); on language, technology, or artifact (A2); and on domain (A3).

The last group of scenarios is formed by considering whether adaptations are triggered by changing requirements (B1) or changing perspectives (B2). Which scenarios belong to which category is illustrated in Table 1.

		AO: Changes in the application context	**AP: Changes for specific assessments**	**CI: Model correction and improvement**
A1	**Data, measures and tools**	Q1.- New data sources	Q8.- Using a subset of data sources	
A2	**Language**	Q2.- New modeling / implementation languages	Q9.- Focusing on specific modeling / implementation languages	
	Technology	Q3.- New technologies or paradigms	Q10.- Reducing technology or paradigm scope	
	Artifact	Q4.- New artifacts	Q11.- Assessing a subset of artifacts	
A3	**Domain**	Q5.- New domains	Q12.- Focusing on actual domain	
B1	**Requirement**	Q6.- New/modified quality requirements	Q13.- Modified project-specific quality requirements	
B2	**Perspective**	Q7.- New/updated perspectives	Q14.- Performing assessments limited to specific perspectives	
-	**General**			Q15.- Fixing observed assessment problems Q16.- Calibrating the model for future assessments

Table 1: Grouping of adaptation scenarios according to their focus and motivation

Hypotheses

To answer our first research question (Are specific types of adaptation scenarios more relevant than others?), we tested the following hypotheses:

Quality model adaptations at the organizational level, at the project level, and for correcting or improving a model are equally relevant:
$H1_0$: $\tilde{x}(AO) = \tilde{x}(AP) = \tilde{x}(CI)$

Quality model adaptations at the organizational level, at the project level, and for correcting or improving a model are not equally relevant:
$H1_A$: $\exists$ a, b $\in$ {AO, AP, CI}: $\tilde{x}(a) \neq \tilde{x}(b)$

To study our second research question (Are adaptations focusing on more concrete concepts in a quality model (bottom level) considered more relevant than changes to more abstract concepts (top level)?), we tested these hypotheses:

Quality model adaptations are equally relevant for concrete and abstract focuses:
$H2.1_0$: $\tilde{x}(A1) = \tilde{x}(A2) = \tilde{x}(A3)$

Quality model adaptations are not equally relevant for concrete and abstract focuses:
$H2.1_A$: $\exists$ a, b $\in$ {A1, A2, A3}: $\tilde{x}(a) \neq \tilde{x}(b)$

Quality model adaptations are equally relevant for changes in requirements and changes in the perspective:
$H2.2_0$: $\tilde{x}(B1) = \tilde{x}(B2)$:

Quality model adaptations are not equally relevant for changes in requirements and changes in the perspective:
$H2.2_A$: $\tilde{x}(B1) \neq \tilde{x}(B2)$:

Data Analysis

Five of the scenarios (Q1, Q3, Q9, Q13, Q15) were considered very important ($\tilde{x}>6$). The most important scenarios are related to adaptation *per se*, whereas the least important is the scenario motivated by a new domain (Q5) (Table 2). All other scenarios were considered rather important or quite important ($4> \tilde{x}\leq 6$).

We observed a difference in relevance among the groups AO, AP, and CI, as can be seen in Figure 1. It seems that CI is more important than AP and in turn

AP seems to be more important than AO. To test whether our observation is statistically significant, we used a Friedman test followed by Wilcoxon tests.

Friedman test result: There was a statistically significant difference in relevance among the groups AO, AP, and CI, $\chi^2(2) = 10.118$, $p = 0.006$, with which we reject $H1_0$.

	N		Median $\tilde{x}$
	Valid	Missing	
Q1: New data sources	**10**	**0**	**6.50**
Q2: New modeling/implementation languages	10	0	5.00
Q3: New technologies or paradigms	10	0	5.00
Q4: New artifacts	10	0	5.00
Q5: New domains	10	0	2.00
Q6: New/modified quality requirements	9	1	6.00
Q7: New/updated perspectives	10	0	4.50
Q8: Using a subset of data sources	**10**	**0**	**7.00**
Q9: Focusing on specific modeling/implementation languages	**10**	**0**	**6.50**
Q10: Reducing technology or paradigm scope	10	0	6.00
Q11: Assessing a subset of artifacts	10	0	5.50
Q12: Focusing on actual domain	10	0	5.50
Q13: Modified project-specific quality requirements	**10**	**0**	**6.50**
Q14: Performing assessments limited to specific perspectives	10	0	4.50
Q15: Fixing observed assessment problems	**9**	**1**	**7.00**
Q16: Calibrating the model for future assessments	10	0	6.00

Table 2: Median considering each scenario separately (bold: $\tilde{x}>6$, i.e., very important; gray shading: $\tilde{x}\leq 4$, not important).

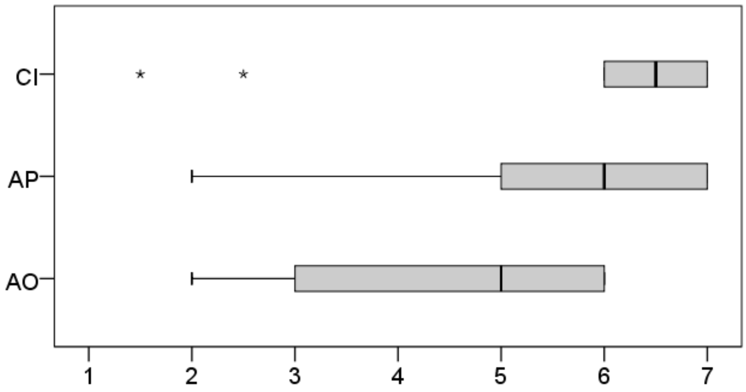

Figure 1: Box plots for AO, AP, and CI.

A *post-hoc analysis with Wilcoxon signed-rank tests* was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$. Median relevance for AO, AP, and CI was 5.0 (2 to 6), 6.0 (2 to 7) and 6.5 (1.5 to 7), respectively. There were no significant differences between AO and AP (Z = -2.232, $p = 0.026$), between AP and CI (Z = -0.570, $p = 0.569$), or between AO and CI (Z = -1.965, $p = 0.049$).

Considering the target concepts, we observed that importance seems to decrease from more concrete changes (A1) to broader and more general changes (A3) (Figure 2). To test whether our observation is statistically significant, we used a Friedman test followed by Wilcoxon tests.

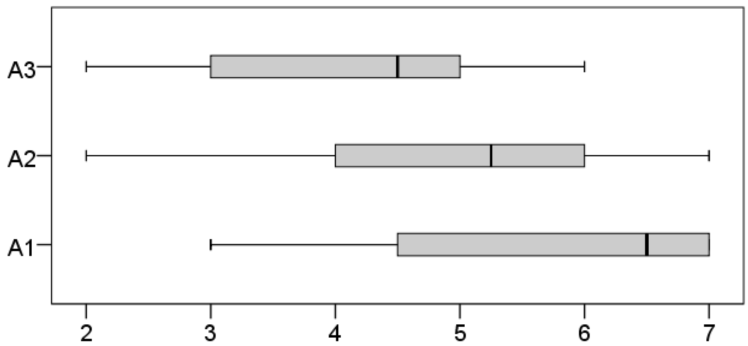

Figure 2: Box plots for A1, A2, and A3.

Friedman test result: There was a statistically significant difference in relevance among the groups A1, A2, and A3, $\chi^2(2) = 9.892$, p = 0.007, with which we reject $H2.1_0$.

A *post-hoc analysis with Wilcoxon signed-rank tests* was conducted with a Bonferroni correction applied, resulting in a significance level set at $p < 0.017$. Median relevance for A1, A2, and A3 was 6.5 (3 to 7), 5.25 (2 to 7) and 4.5 (2 to 6), respectively. There were no significant differences between A1 and A2 (Z = -2.345, $p = 0.019$) or between A2 and A3 (Z = -2.030, $p = 0.42$). However, there was a statistically significant difference between A1 and A3 (Z = -2.458, p = 0.014).

Last, we observed that adaptations triggered by changes in requirements (B1) seem to be more important than those triggered by changes in the perspective (B2). To test whether our observation is statistically significant, we used a Wilcoxon test.

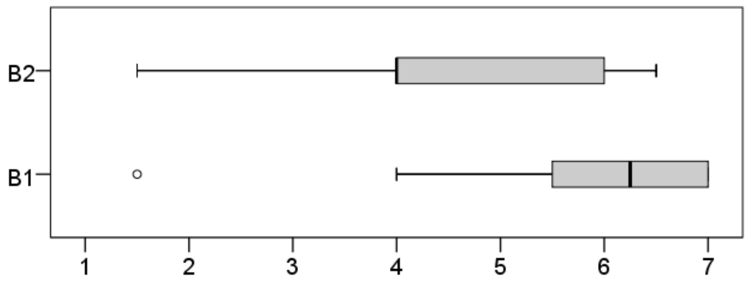

Figure 3: Box plots for B1 and B2.

Wilcoxon test result: Median relevance for B1 and B2 was 6.25 (1.5 to 7) and 4 (1.5 to 6.5), respectively. There was a significant difference between B1 and B2 (Z = -1.975, $p = 0.048$), with which we reject $H2.2_0$.

Interpretation

Adaptations trigged by the need for corrections and improvements as well as changes required to adjust the model for a specific assessment occur more frequently in practice and are therefore considered more relevant.

Increasing major changes caused, e.g., by a change in the domain or an additional perspective addressing a further stakeholder have to be addressed rarely in practice and therefore ranked as less important when compared with more concrete changes (e.g., inclusion of further data sources or requirements). Changes on a higher level may currently lead to the creation of an additional independent quality model.

4.1 Adaptation Guidelines

In [11], we presented a general process for adapting quality models. In this section, we propose guidelines for applying the process according to the scenarios described in the previous chapter.

As illustrated in Figure 4, the general adaptation process has four steps:

1. Specify goal of adapted quality model (QM): takes as input quality needs and context information and delivers the goal of the adapted quality model, which serves as a basis for further adaptation.

2. Identify reference QM: The goal created in step 1 is used to identify a model and adapt it to the needs of the project or organization (reference model).

3. Tailor QM: Once a reference model has been chosen, elements not needed in the final model are discarded. The unnecessary components are eliminated at the beginning in order to reduce the size and thus the complexity of the model. Specific elements to be partly reused and stubs for new elements are marked for detailed inspection and modification in the next step.

4. Iterative changes: The actions performed during the removal of components and the creation of stubs trigger further consistency and adaptation tasks. Iterative changes help bring the model back to a consistent, operational state. Some tasks can be automated (*consistency tasks*). Other tasks will require user interaction, as they are based on user decisions (*adaptation tasks*). Stepwise, elements can be deleted, added, or modified until no further adaptation tasks are required. The extent to which these operations are used depends on the appropriateness of the reference model. At this point, the QM should be piloted to test its suitability for the specified application purpose.

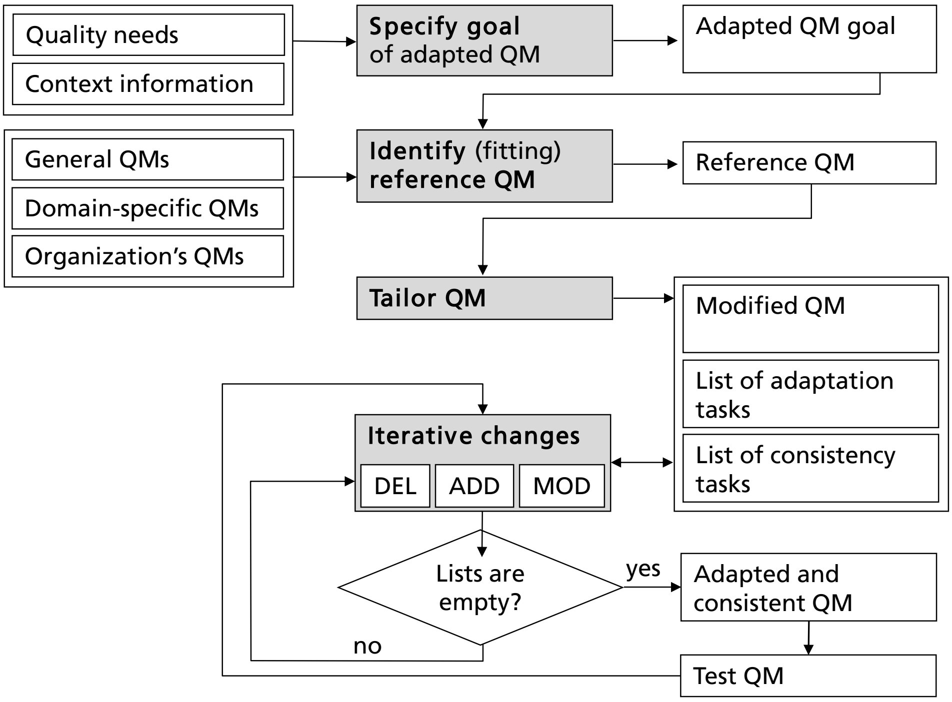

Figure 4: General adaptation process based on [11]

In scenario Q1, the focus is on new data sources. For instance, an additional static code analysis tool not considered yet in the model should be used to provide additional information for evaluating software quality. The adaptation can start with the addition of a stub for the new tool. Then, it is important to connect the tool to the model parts that will use the data.

In scenario Q2, the focus is on new modeling/implementation languages. E.g., your current model is applicable for C, but now you want to assess C++ software. To add a new programming language, several things need to be identified, added, and/or changed. The model is known but its context information has to be updated, that is, C++ has to be added to the model's goal. Then the model parts that consider C and can be reused for C++ have to be marked for tailoring. The parts that are exclusive for C and cannot be reused for C++ have to be deleted. Finally, model elements that are relevant for C++ and not for C need to be added.

In scenario Q3, the focus is on new technologies or paradigms. For instance, your current model considers only software for single-core microcontrollers, but now you want to also consider multi-core microcontrollers, or you plan to introduce model-based development. This case may have consequences for parallelization and deployment measures. For example, measures regarding correct function isolation can be added, which will also lead to the addition of new entities.

In scenario Q4, the focus is on new artifacts. For instance, your current model considers only the source code for assessing quality, but now you want to also consider the design documents. In this case, it is important to identify which existing quality characteristics are expected to be supported by the design documents and whether new quality characteristics need to be added. Then, entities and measures can be derived.

In scenario Q5, the focus is on new domains: For instance, a model elaborated for the information system domain should be adapted to be applicable in the service-oriented architecture domain. In this case, it is necessary to identify whether the model for information systems has elements that also exist in service-oriented architectures. The elements that can be reused are marked for detailed adaptation and new elements relevant for service-oriented architecture need to be added.

In scenario Q6, the focus is on new/modified quality requirements. For instance, the adapted model should enforce lower code complexity in order to increase maintainability. The new quality requirement has to be added and connected to the maintainability element. In this case, it is important to define how the new requirement affects the maintainability evaluation and its trade-off with other requirements that have an influence on maintainability. Furthermore, it needs to be identified for which entities and with which metrics and tools lower code complexity is to be measured.

In scenario Q7, the focus is on new/updated perspectives. For instance, your current model considers the end-user view, but now you want to assess quality also from the point of view of the operating IT department. Here it is again important to identify the differences in perspectives and the model parts that can be reused and customized. Further end-user-specific model elements need to be added.

In scenario Q8, the focus is on using a subset of data sources. For instance, only a limited set of tools can be used or priorities have to be set according to cost of data collection, trustworthiness of collected metrics, or relevance of quality aspects. This is a typical case where the quality model needs to be reduced. The elements to be removed, such as tools that are not available, are deleted. Afterwards, consistency checks have to be performed.

In scenario Q9, the focus is on a specific modeling/implementation language. For instance, the model covers both Java and C++, but now you want to assess only software written in Java or the Java-based parts of your software. In this case, the model already covers the required implementation language and needs to be reduced. Elements that exclusively cover C++ can be discarded. Only quality model components in the reference model that are relevant for the new model are taken over. After sorting out irrelevant information concerning only C++ software, adaptation tasks help to bring the model back to a consistent, operational state.

In scenario Q10, the focus is on reducing the technology or paradigm scope. E.g., the current model considers procedural and object-oriented programming, but since the assessed products do not use OO concepts, the model needs to be adapted to focus the assessment on procedural programming. Again, the adaptation consists of identifying those elements that are not relevant for the model goal (OO concepts in this example), removing them, and bringing the model into a consistent state.

In scenario Q11, the focus is on assessing a subset of artifacts. For instance, the model covers both source code and design; however, at the current point in time, only the design documents are available and should be assessed. In this case, the elements that concern only source code are deleted and the model has to be checked for consistency.

In scenario Q12, the focus is on the actual domain. For instance, a model applicable for the information and embedded systems domains should be adapted to assess embedded systems. Now, the elements that consider embedded systems are kept in the model and all elements that are only relevant for information systems are deleted, and the model has to be brought back to a consistent state.

In scenario Q13, the focus is on modified project-specific quality requirements. For instance, for the assessed product, a lower coupling level than usual is required, which should be addressed by the adapted quality model. This implies that the company or organization probably has a general quality model, which can be applied in similar projects. In this case, the adaptation consists of lowering the threshold for acceptance of the coupling level.

In scenario Q14, the focus is on performing assessments limited to specific perspectives. For instance, the model covers the developer as well as the management point of view, but the current assessment should focus only on the management perspective. This example also focuses on reducing the quality model; in particular, the elements concerning exclusively the developer perspective have to be eliminated and the model is afterwards modified to recover consistency.

In scenario Q15, the focus is on fixing observed assessment problems. E.g., you adapt the specification of a measure that delivers wrong results or correct a modeling mistake. This adaptation is straightforward and consists of checking and correcting those measures that give wrong results.

In scenario Q16, the focus is on calibrating the model for future assessments. For instance, you adapt the evaluation rules or target values in the model to fit better into your environment. This adaptation is also straightforward and consists of checking and adjusting the evaluation rules to the current environment.

5 Conclusions

Adaptation methods should primarily focus on supporting correction and improvement changes as well as changes required to adjust the model for a specific assessment. Adaptations extending the model to address changes in the application context (e.g., new tools, supported languages or artifacts) are considered to be of less importance. However, they should not be neglected when providing a comprehensive adaptation approach because if they occurred they are significantly more complex and time-consuming based upon our experience.

Adaptation methods should support especially those changes occurring on a lower abstraction level. In future, better support of higher abstraction levels may lead to increased relevance of such adaptations due to cost savings compared to building a new quality model from scratch.

Finally, it is very important to know the required model scope and define a goal that compactly describes this scope. The goal helps to find similar models for adaptation. Without a clear goal, the risk of adapting the wrong model increases, which may be more time-consuming than developing a new model.

References

[1] Andersson T and Eriksson IV (1996): Modeling the quality needs of an organization's software. In: Proc. of the 29th Hawaii Int. Conf. on System Sciences Vol. 4: Organizational Systems and Technology, 139.

[2] Andreou AS and Tziakouris M (2007): A quality framework for developing and evaluating original software components. In: Inf. Softw. Technol., 49(2), 122-141.

[3] Bianchi A, Caivano D, and Visaggio G (2002): Quality models reuse: experimentation on field. In: 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment, 535-540.

[4] Calero C, Cachero C, Córdoba J, and 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, 192-201.

[5] Franch X and Carvallo JP (2003): Using quality models in software package selection. In: IEEE Software, 20(1), 34-41.

[6] ISO/IEC 14764 Software Engineering - Software Life Cycle Processes - Maintenance, 2006.

[7] ISO/IEC 25010 Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality model and guide 2009.

[8] ISO/IEC 9126 Software Engineering - Product Quality - Part 1: Quality Model, 2001.

[9] Khaddaj S and Horgan G (2005): A proposed adaptable quality model for software quality assurance. In: Journal of Computer Science, 1(4), 482-487.

[10] Kläs M, Heidrich J, Münch J, Trendowicz A: CQML Scheme: A Classification Scheme for Comprehensive Quality Model Landscapes. In: Proc. of the 35th EUROMICRO Conference Software Engineering and Advanced Applications (SEAA), pp. 243-250.

[11] Kläs M, Lampasona C, and Münch J (2011): Adapting software quality models: Practical challenges, approach, and first empirical results. 37th EUROMICRO Conference on Software Engineering and Advanced Applications, 341-348.

[12] Kläs M, Lampasona C, Trendowicz A, and Münch J (2010): Goal-oriented adaptation of software quality models. In: Proc. of the 3rd Workshop on

Modeling and Assessment of Software Quality SQMB'10.

[13] Lampasona C, Heidrich J, Basili V, and Ocampo A (2012): Software Quality Modeling Experiences at an Oil Company. To appear in: Proc. of the 2012 International Symposium on Empirical Software Engineering and Measurement (ESEM). Lund, Sweden.

[14] Lampasona C and Kläs M (2011): Supporting the adaptation of software quality models – An empirical investigation. In: Proc. of the 4th Workshop on Modeling and Assessment of Software Quality SQMB'11.

[15] McCall JA, Richards PK, and Walters GF (1977): Factors in software quality. Concept and definitions of software quality: Final Technical Report Springfield: NTIS, Reportnr. RADC-TR-77-369 (I, II and III).

[16] Plösch R, Gruber H, Körner C, Pomberger G, and Schiffer S (2010): Adapting quality models for assessments - Concepts and tool support. In: Proc. of the 3rd Workshop on Modeling and Assessment of Software Quality SQMB'10, 12-21.

[17] Wagner S, Broy M, Deißenböck F, Kläs M, Liggesmeyer P, Münch J, Streit J (2010): Softwarequalitätsmodelle. Praxisempfehlungen und Forschungsagenda. In: Informatik Spektrum, Springer, 33(1), pp. 37-44.

[18] Wagner S, Lochmann K, Winter S, Goeb A, Klaes M (2009): Quality Models in Practice. A Preliminary Analysis. In: Proc. of 3rd International Symposium on Empirical Software Engineering and Measurement (ESEM).

[19] Wagner S, Lochmann K, Winter S, Göb A, Kläs M, Nunnenmacher S (2010): Software Quality Models in Practice. Survey Results. Technical Report, https://quamoco.in.tum.de/wordpress/?p=539&lang=en.

…

Document Information

Title:	Application Scenarios and Guidelines for Quality Model Adaptation
Date:	November 2012
Report:	IESE-102.12/E
Status:	Final
Distribution:	Public Unlimited