

QUALITY ORGANIZER: A SUPPORT TOOL IN USING MULTIPLE QUALITY APPROACHES

Zádor Dániel KELEMEN^(1,2), Dr. Katalin BALLA^(1,2), Gábor BÓKA^(1,2)

⁽¹⁾ Department of Control Engineering and Information Technology,
Budapest University of Technology and Economics (BUTE),
Magyar tudósok körútja 2, 1117 Budapest, Hungary,
zady@inf.bme.hu, {balla, bokusz}@iit.bme.hu

⁽²⁾ SQI Hungarian Software Quality Consulting Institute,
József krt. 53, 1083. Budapest, Hungary,
{kelemen.daniel, balla.katalin, boka.gabor}@sqi.hu

Abstract: In this paper we describe the main elements of a theoretical quality framework and the first working prototype of a software tool supporting it. The theoretical framework helps to get a better understanding of the complex subject of software quality and to use multiple quality approaches in a synergic way. The supporting software can be used both to understand the company's actual maturity level, and to get a deeper insight into the relationships among existing quality models and standards, thus helping the company in choosing the right quality approach.

Keywords: Software Quality Management, Software Process Improvement, Situation-dependent software process improvement, CMMI

1 Introduction: the problem

Doing professional quality management is a must for software developing companies trying to stay in the market. Besides the fact that quality certificates are sometimes required by business partners, software companies simply cannot afford to work in a chaotic way.

The standards and models used in software industry are extremely various in their approach used. The process of producing software is being still very emphasized (by models like CMMI / SPICE, by ISO 9001:2000, AQAP, ISO 12207, by project management methodologies and software development methodologies). Approaches concentrating on software product characteristics and metrics (like ISO 9126 family) are also being more and more accepted, and the same statement can be made for models and

approaches stressing the importance of the human factor (e.g. P-CMM, PSP, TSP). As no approach, model or standard covers all the aspects of software quality (although new versions models are definitely broader in their scope than the earlier ones) companies will have to choose the right approach based on their business needs.

Companies doing model-based software process improvement always face two difficult questions: which model to choose to best fit the company's needs and in which direction to move for a higher software quality?

Answering these questions claims solid professionalism both in the field of software development's nature and actual state at a company, and existing quality models and standards. As there is no universally definable "good quality", a software company, in order to do efficient software quality management, will have to understand its most important objects in software development and to choose the right quality approach to bring it to a higher level.

This – real-life – problem was the starting point for a research and development done in Hungary. A theoretical framework and a software tool were developed to help companies identifying important objects of software production and position them against popular quality standards and models.

Chapter 2 shortly presents the theoretical framework, while chapter 3 is about the software tool. In the end of this chapter we conclude with feedback obtained on the prototype, and possibilities of further of development.

This paper was written within the project KKK-GVOP-2004-K+F-3-2-2.

2 QMIM: Quality through Managed Improvement and Measurement

QMIM – Quality through Managed Improvement and Measurement (described in detail in [Balla 2001]) is a framework helping to deal with software quality in its complexity. It gives guidance to software companies in identifying the most important elements of their software production and to find their way among the many models, standards, methods connected to software quality.

The framework facilitates understanding, consciously selecting and applying models, standards and methods connected to software quality, and combining them in a way that fits best for own needs. QMIM is not a new approach or method towards software quality, but it acts like a "navigator" to clarify important objects of software quality and their interconnections, as well as to position any existing quality standard, approach or methodology against these objects. The framework has well defined elements, static and dynamic aspects. A representation of the framework can be seen in Figure 1.

This representation shows the basic *objects* of software production: itself the software *product*, the *processes* that develop the product and the *resources* that execute the processes. Each object has some *characteristics*. It has to be *defined* in the actual situation, its *quality attributes* have to be identified and their values have to be measured (*metric*).

QMIM guidelines have been developed to help making the QMIM framework operational. Having QMIM reference framework as a starting point, the guidelines provide aid in populating the quality framework and using it in the way that fits the company's goals in the best way. The guidelines, in fact, help a company in *understanding its actual situation* in terms of important elements of software production (eg. product characteristics, company maturity level or capability level for some important processes). After having

understood the actual situation, QMIM guidelines give aid in selection of the quality improvement approach best fitting the company's own needs. Popular software quality models and standards are positioned against the basic elements of the framework (see some examples in Figure 1). Using QMIM guidelines, one will understand that most important business goal is to define product quality metrics, and measurement process (definition). Next, eg. ISO 9126 will be chosen as an approach for product metrics, and CMMI Measurement and analysis process area (combined with GQM) will help in defining the company's measurement process. Using the guidelines step by step, the company will be able to use more quality approaches in a synergic way, always connected to its business needs.

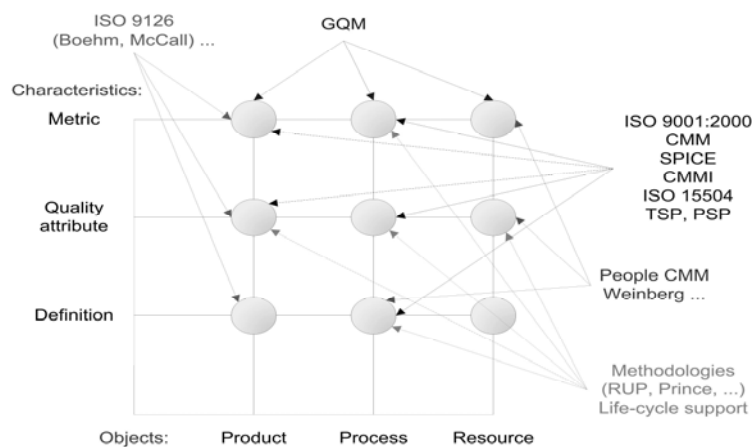


Figure 1 – The QMIM framework

While executing the activities suggested by the QMIM guidelines, one will generate different types of data. The quality manager will describe the project management and the technical processes, will identify different quality attributes and metrics to measure them, will record data regarding evolution of projects, results of measurements etc. Therefore, a static data model / view of QMIM framework was developed: the QMIM data model.

3 QMIM Quality Organizer

The QMIM Quality Organizer is a software tool supporting the QMIM framework, which was developed by SQI - Hungarian Software Quality Institute together with the Technical University of Budapest and Technical University of Eindhoven within a project sponsored by the EU and the Hungarian Ministry of Trade (TST-GVOP-2004-04-0079/3), between 1.12.2004 -1.12.2006.

The software development was started by a rigorous requirements specification. The interesting disputes regarding the functionality of QMIM carried out between industrial and academic are mentioned in [Bóka 2006]. Partners agreed in the following basic functionality of the tool:

- Assessment of the actual situation should be a self assessment, based on a popular quality model, the CMMI. The tool should have a self-assessment module, containing CMMI-requirements for the different processes (see 3.2)
- Quality Organizer should provide a knowledge base containing the most popular quality approaches, standards and models in a structured, easily searchable format (see 3.3).
- The tool should provide the complete description of QMIM guidelines. It should provide connection of different standards and models to the elements of the QMIM theoretical framework (showing eg. in what percentage a standard covers a certain element of the framework).

The software tool was developed using modern technological solutions (see 3.1). After internal testing, it was presented to 3 Hungarian software companies. The feedback obtained is used to further development of the tool (see 3.4).

In the following we present some elements of the project in more detail.

3.1 Some architectural and technological considerations

Due to the number of employees performing the process improvement, the companies demand to gain multiple user access both for the CMMI self-assessment tool and for the knowledge base. One convenient solution is an intranet application. Therefore QMIM Quality Organizer is a client-server intranet web application (composed by a web server and a HTTP browser).

Taking into account that different companies use various software environments, we designed a client-server software which could be integrated in any environment without additional financial investments of the organization. Satisfying this requirement, we developed the software to run on the free LAMP¹ architecture. As an addition, the data management layer was developed in a way permitting the companies to freely choose among many different types of database servers (some of them may already have their default DB solution). This facility was integrated by using the PEAR² DB_Data_Object package. Several up to date technologies were used, like AJAX, Yahoo! User Interface Library (YUI), and YUI-ext.

3.2 The self-assessment tool in brief

The QMIM Quality Organizer offers a CMMI-based self assessment support³, by which the companies could assess [SCAMPI 2001] their maturity or capability level [CMMI 2006] (the function of requirements rating can be seen in Figure 1). Using the tool, the companies can gather evidences connected to each CMMI requirement. As in CMMI projects need to be assessed, basic management of organizational and project data was added to the software.

QMIM Quality Organizer creates an overview of the degree to which CMMI requirements are satisfied, based on objective evidences⁴ and ratings. The overview shown

¹ Further information at: <http://www.onlamp.com/pub/a/onlamp/2001/01/25/lamp.html>

² Further information can be found at <http://pear.php.net/>.

³ This functionality is compliant with the SCAMPI C appraisal method [SCAMPI 2001].

⁴ To prove the conformity to CMMI requirements, different types of evidences need to be collected.

in Figure 1 can be generated for different CMMI maturity and capability levels⁵ (colors symbolize the degree of meeting requirements).

By the end of an assessment, a report could be generated, in which the most important assessment data are included (assessed company, appraisal details, projects included, evidences attached to requirements, requirements satisfaction level etc.)

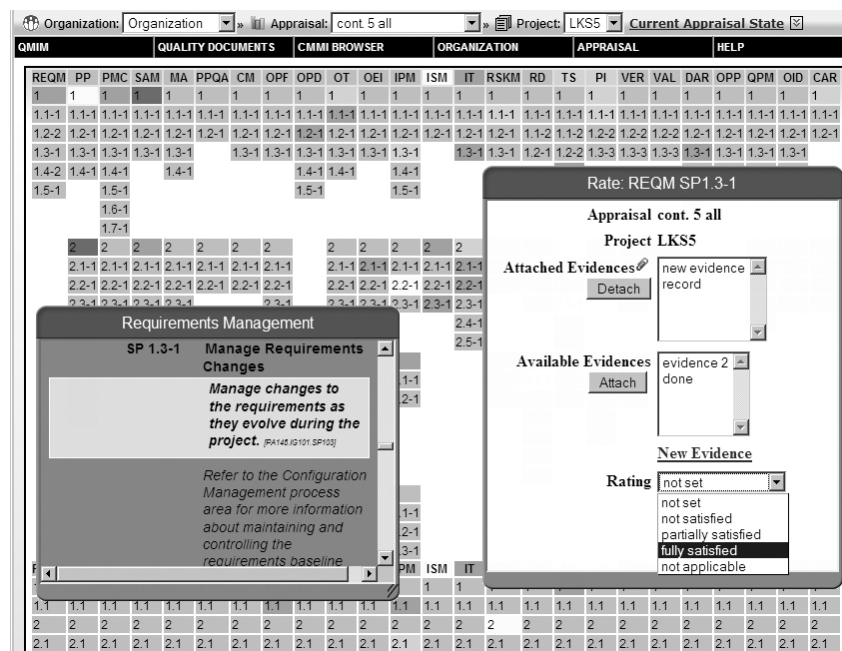


Figure 2 – CMMI self assessment overview (level 5, continuous representation)

3.3 Some features of the knowledge base

Depending on the results given by the assessment tool, a situation-dependent improvement program can be started, concurrently using different quality approaches. Supporting this, the QMIM Quality Organizer offers a knowledge base that stores well known quality standards, best practices, models, definitions and other types of quality approaches.

We categorized the quality elements into: guideline, case study, best practice, tutorial, lifecycle, template, definition, metric, quality attribute, software tool and certificate description. The detailed description of these categories can be found in [Bóka 2006].

Up to now, the knowledge base contains 25 quality elements: the description of: 3 ISO standards, 11 ISO-IEC standards, 9 Hungarian standards, CMMI model and browser, and a detailed description of QMIM framework. These quality documents are continuously upgraded and maintained by the developer SQL. The software shows all these quality approaches in connection to the QMIM framework.

⁵ By using the CMMI model we can choose between two different approaches: staged and continuous. Based on the approach chosen maturity or capability levels can be assessed.

All documents included in the software are converted to a common, searchable format. Additional details of approaches are included to QMMI Quality Organizer, as well as a common search function covering all the documents. Moreover, all definitions found in quality documents have been extracted and added to the database, so the definitions could be checked if needed (see Figure 2).

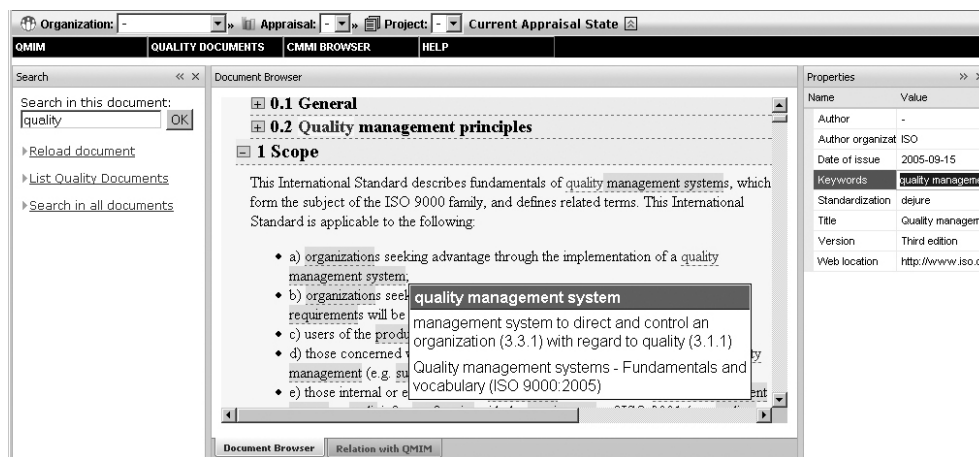


Figure 3 – Browsing a quality element in knowledge base

3.4 Feedback received

Three, differently sized Hungarian software companies were surveyed for feedback on the first working prototype of QMIM Quality Organizer (details on the survey described in [Backer 2006]).

Feedback shows that companies having already dealt with the organizational factors (like management commitment and staff involvement) could benefit from the knowledge-base, overview of relations between standards (with the help of the QMIM model) and the self-appraisal tool provided.

It seems that it is the size of company that influences their choice of Quality Organizer functions. The size might also be related to the degree into which a company has already made progress in the field of quality management. The largest company has the most experience in quality improvement projects. Such a company is more interested in the self-appraisal functions, while the smaller companies are more focused on the knowledge base.

4 Conclusion

Although, the number of companies surveyed is too small to draw global conclusions, there is a clear indication that QMIM Quality Organizer has added value for software engineering companies which are working on continuous software quality improvement.

Further development on QMIM Quality Organizer is needed to build a competitive commercial product. Based on the feedback received, our intention is to continue the project by refining the actual version. By making the GUI more uniform, refining the

overview of self-appraisal and the generated report, as well as including furthermore calculations of maturity levels could be directions for improvement, as well as including further quality elements into the knowledge-base and linking different approaches (eg. including and linking PSP, TSP and GQM, linking CMMI and ISO9001:2000 (we already did research regarding the latter). Another interesting question would be the separation of the two main functionalities to two different products.

5 Acknowledgement

We would like to thank Prof. dr. Rob J. Kusters and Dr.ir. Jos J.M. Trienekens for their indispensable scientific support. We would like to thank as well Peter Schroen, Dirk van Driel and Hans de Backer for their contribution in defining the requirements and getting feedbacks from Hungarian software companies. Their results were used while writing this paper.

6 References

- [BALLA 2001] BALLA K. 2001: The complex quality world. Developing quality management systems for software companies. Ph.D. thesis. Beta Books, Technische Universiteit Eindhoven, 2001. ISBN: 90-386-1003-3
- [BÓKA 2006] Bóka G., Dr. BALLA K., Prof. dr. KUSTERS R., Dr.ir. TRIENEKENS, J. 2006: Towards Tool Support for Situation-Dependent Software Process Improvement. EuroSPI2006 Conference Proceedings, 9.1-9.11.
- [BACKER 2006] BACKER H., 2006: Practical Training Report
- [SCHROEN 2005] SCHROEN P. 2005: Modeling and Managing Software Quality, an elaboration on QMIM. TU/e Internal Research Report. TU Eindhoven, 2005.
- [CMMI 2006] CMMI[®] for Development, Version 1.2 CMU/SEI-2006-TR-008
- [SCAMPI 2001]: Standard CMMI[®] Appraisal Method for Process Improvement (SCAMPISM), Version 1.1: Method Definition Document.