Production Systems and Information Engineering
Volume 2 (2004), pp. 73-89

SYNERGIC USE OF SOFTWARE QUALITY MODELS

KATALIN BALLA
Technical University Budapest, Dept. of Control Engineering and Information Technology
e-mail: balla@iit.bme.hu

[Received November 2004 and accepted January 2005]

Abstract. The article supports the idea that nowadays using more software quality models in a synergic way and customizing them to fit the specific needs of an organization is the only viable option for doing efficient process improvement in software companies.

The argumentation is done by presenting a theoretical framework in which the quality models can be placed. Next, we describe our personal experience with using this framework in a Hungarian software company, presenting the main results of a 11 years-long case study[1] The article emphasizes the business-driven SPI project of the company, started to enrich the ISO 9001:2000-conform quality system, and the way of consiously using more quality models to do this. In the end of the article we shortly present the huge organizational change the company has undergone, and its consequences on the previously started software process improvement program.

Keywords: software quality management, ISO 9001, CMM, software process improvement (SPI), project management (PM)

1. INTRODUCTION

In the intense international competition software companies are more and more forced to think about proving their capability of delivering good products. One way of having such a proof is to obtain an official certificate about usage of a certain standard or model. However, introducing a quality approach based on a standard or model, and institutionalising it, so that the organisation is able to pass an audit, requires a lot of investment from the software companies, both in terms of money and effort – which a company would not like to waste. Therefore, really business-driven software companies will be willing to do only *really efficient* software process improvement.

[1] The author worked between 1993-2004 as a quality manager at the company where the case study was run. Thanks are due to IQSOFT management for making possible to do research and record the steps of process improvement.

Efficient improvement programs are always based on real needs of companies and will always start from understanding the actual situation of that company. Choosing the right approach, model or standard for the improvement program would be the next step.

The difficult question is: which model to choose to best fit the company's needs in improving software quality? What activities to execute and in which sequence in order to transform the initial – probably almost chaotic – situation, step by step, into some controllable and provable "order"?

We faced the described problem while doing quality management in a Hungarian software company. As a result of the research and practical work done over 11 years, a theoretical framework has been worked out and used. Our experience has shown that the framework is well usable in a "real" environment. Therefore, we consider it worth to present in the following.

1.1. The quality framework

1.1.1. QMIM elements

For answering the questions about a "good approach of software quality", we have to understand what software quality means for the companies, in each particular situation. Quality of software is a very complex subject, and, as such, it is extremely hard to define. If we wish to deal with software quality in its complexity, we have to think about the software *product*s, the *process*es that produce the products and the *resource*s that execute the processes. We have to *define* these objects, to choose the right *quality attributes* for them and verify their actual value by the means of objective *metrics*. In conclusion, a framework capturing the important elements of software quality can be defined (see Figure 1). The framework has been named QMIM (Quality through Managed Improvement and Measurement). It is presented in detail in [1] and [2]. Here we describe its most important element and features.

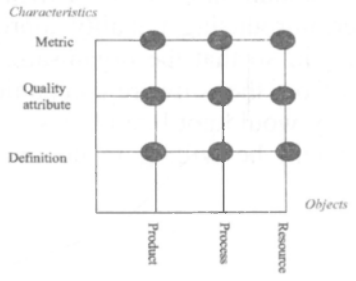

Figure 1: QMIM: the quality framework

Basic idea is that if we wish to deal with software quality in its complexity, *we have to completely fill the QMIM framework.* This means that the software product, the processes producing it and the resources executing the processes have to be equally well understood, their quality attributes need to be defined (we have to understand when do we say that an object "is good"), and the actual value of the attributes need to be measured by objective metrics. Normally, such a detailed quality approach is not possible to follow: it would mean a huge investment to the company both in terms of effort and time, which might not be in line with the actual business needs. However, a long-term vision about quality within a software company requires the understanding of the QMIM elements and the capability to *unambiguously identify them in the actual environment.* With other words: a software company needs to know what its products are, what their characteristics mean and how they can be measured, has to be able to define its processes and ensure the right resource management. Efficient quality management can be done having QMIM elements always in mind and *consciously choosing* the most appropriate ones in each particular situation. This means that *certain elements* of the QMIM framework will be dealt with in more detail at a certain moment, but the company will always be aware of the fact that *all* elements need to be addressed sooner or later.

1.1.2. QMIM static aspects

The processes of a software company can be grouped into project management processes (common for all development projects) and technical processes (bearing the particularities of each development, depending e.g. on technology, methodology etc.). If we agree that resources are managed via project management processes, thus including resources into project management, we have the following objects of software production: project management processes, technical processes and products.

The principles of QMIM help in structuring the quality-related data of a company: the objects, characteristics and metrics are understood and their relationships can be represented. The static aspect of the QMIM framework (see [1]) describes a possible database for storing the data related to quality. See Figure 2 for the structure of the database.

As seen from the figure, the products can be grouped into software systems, that contain items. The items are developed by technical processes that follow a development methodology, described in guidelines. Technical processes are executed within projects. The projects are managed in a way that is described in a PM methodology. The projects use resources. All important elements - software item, technical process, project management process and resource – have associated quality attributes that can be measured by metrics.

The database can be used for storing both generic information (that describes the basic objects and their relationships) and the data of the concrete projects.

To be noted that such a database will not need to be built from scratch, as (almost) all companies have some data-collections that can be incorporated. If the understanding of the structure and relationships is there, the database will always be possible to complete with new elements.

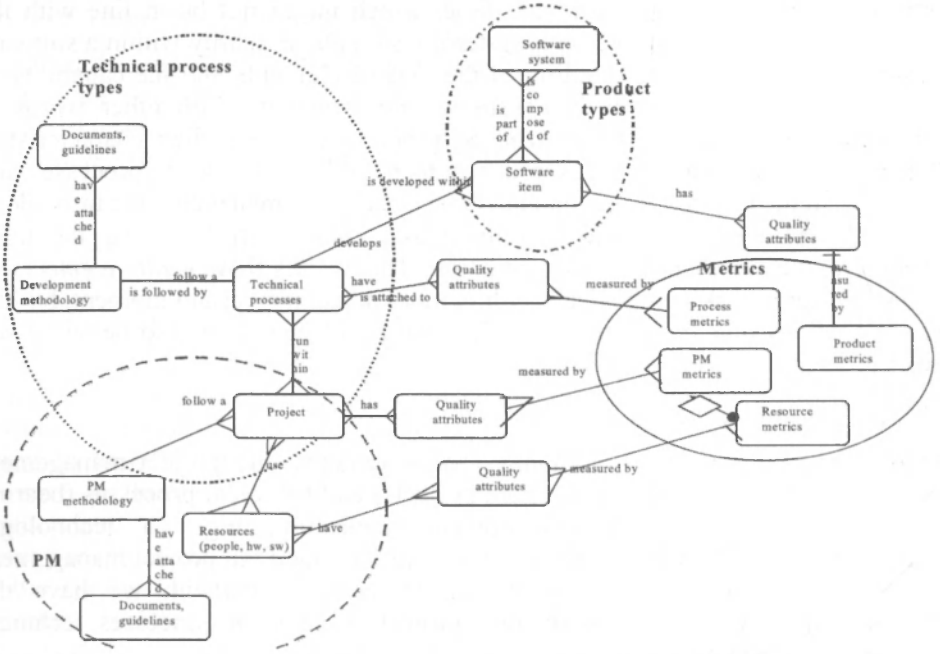

Figure 2: Database structure suggested by QMIM

1.1.3. QMIM guidelines

If a company has the right understating about the quality of its software, it can direct the company-wide processes and process improvement in a way that will permit a step-wise completion of the QMIM framework. There is no strict recommendation about the sequence of "filling" the framework: it can be done both "horizontally" and "vertically" The horizontal approach means that the company starts by obtaining an equally deep understanding of all *objects* of software production, focusing its effort to quality objectives and metrics afterwards. The vertical approach means that the company chooses one particular object, and defines its quality attributes and metrics before starting to deal with the next object.

The two approaches might be mixed: a company can at one time define more elements to different degrees of detail. The choice has to be made based on the

understanding of the actual situation and the importance each element presents to the company.

Our experience shows that some elements are easier to understand than others, e.g. processes are easier understood and measured than products. Experience shows also that project management processes are the easiest to define and measure. Technical processes can follow, and product characteristics and measurement is the element hardest to approach.

QMIM guidelines describe these possibilities in detail (see [1] and [2]).

It would be extremely difficult for software companies to define quality approaches from scratch. At the same time, this would be a useless waste of effort, since many approaches, standards, methods, models have been worked out by the software technology - and quality community. QMIM emphasizes this aspect, and *suggests the usage of the most appropriate method, standard or approach in each case*, giving aid in understanding of what "the most appropriate" means.

The approaches, standards and models in software industry are extremely various in their approach used. Here we present some of the most popular ones.

The early Boehm ([3]) and McCall ([4]) quality models concentrate on software product quality, and so does standard ISO 9126 family ([5]), which gives important guidance in defining and measuring software product characteristics.

The nowadays widely used standard, ISO 9001:2000 ([6]), focuses mainly on the processes. It addresses product quality and measurement also, but without giving guidance for these elements.

Managerial and technical processes are addressed by the popular CMM model ([7]) that addresses product characteristics too (but mainly from a process/organisational aspect), and the Bootstrap methodology ([8]) worked out to asses organisation maturity. The SPICE model / ISO 15504 standard ([9]) is also process oriented, as well as CMMI ([10]), developed to integrate (among others) staged approach of CMM and continuous approach of SPICE. PSP (see ([12]) and TSP (see [13]) are completing the CMM(I) approach, concentrating on individual and team aspects of software development. Project management aspects are addressed by many PM methodologies (e.g. PRINCE [14]). Human resource characteristics are dealt with in e.g. Weinberg's theory ([15]) and People- CMM ([16])

Metrics and measurement methods are addressed by e.g. [11] and Basili's GQM paradigm ([17]). Function point counting methods (see [18], [19], [20]) are dedicated to understand software product size and complexity.

Some of these approaches are widely known, while others are used only by a restricted number of companies. The choice a company makes in terms of software quality model/approach used depends on many factors. For instance, ISO

9001:2000 is nowadays a condition of staying in the market. To this (rather general, therefore not easily applicable for software) approach the companies normally add a model or standard their customers prefer. USA was preferring CMM, Europe used SPICE more, but CMMI will probably solve this problem.

QMIM framework does not impose the usage of one specific model, but is suggests the conscious choice of a model, while understanding how that approach, standard or model is related to the important elements of software production.

However, since we consider that the notions of capability, maturity and CMM(I) levels are widely known and accepted by the software development community, we suggest to make a choice based on the company's actual maturity (in terms of CMM[2]). On the lowest maturity level it seems best to approach software quality by project management process definition, characteristics and measurement. This will probably bring the company to maturity level 2, where technical processes can be defined, the result being a level 3 company maturity. At this level product characteristics can be understood, defined and measured.

QMIM strongly emphasizes the need for measurement already on the lowest maturity level. This aspect is in line with the CMMI – structure, that brought the process of measurement and analysis down to maturity level 2 (while in CMM it appeared explicitly only on level 4).

1.2. Using more models in a synergic way

Studying in detail the models presented shortly in this paper, one will remark that *no approach, model or standard covers all the important aspects of software quality* (although new versions of earlier models are definitely more broad in their scope, in the number of objects they are dealing with). We can state that companies will have to choose the right approach based on their business needs. Understanding the *business needs* in a right way is a rather complex job that claims solid professionalism both in the field of software development's nature and existing quality models and standards. Choosing a wrong approach could do considerable harm to a company, by misleading the efforts from the really important objectives.

One way of avoiding the trap of a badly chosen quality model or approach is to quit exclusively relying on *one* certain quality model in favor of choosing among several approaches, consciously *using more approaches in a synergic way*, according to the specific business needs of a certain organisation.

[2] The maturity levels of an organization, in CMM terms, are: level 1: chaotic, level 2: repeatable, level 3: defined, level 4: managed, level 5: optimizing.

In the following part of this article we describe our experience in using QMIM and point out how its concepts helped in using more quality models in a synergic way.

2. THE CASE STUDY

In this chapter we describe the main results of a 11 years long case study done at a Hungarian software company. We make the presentation having QMIM in mind, and showing at every phase how its concepts were used.

The case study took place at IQSOFT Ltd. / later IQSYS Ltd., one of the main representatives of the software industry in Hungary. The initial company was formed in early 1990 from part of a large state organisation, the Theoretical Laboratory within the Computer Technology Co-ordination Institute (SZKI). The company was medium-sized, having around 100 employees. Three main software activity types could have been defined: software development (mainly in a database environment, using 4GL development tools), software integration, and software implementation. The projects were generally small to medium sized and could differ widely in their characteristics. Research and training were also important activities in IQSOFT's activity profile.

In the following sections we will describe the phases of the case study in more detail.

2.1. The enthusiastic start and seeking new ways

Since 1993, efforts have been made at IQSOFT to develop and introduce an ISO 9001-comform internal quality management system (QMS).

As a preparatory step for building the QMS and to be aware of the good practice existing, in 1994 the overall company and two concrete projects were assessed according to the Bootstrap methodology (see Table 1). Due to an insufficient understanding of the interconnections between CMM and ISO 9001, the company was unable to use the Bootstrap assessment results in an appropriate way.

Experiencing the "failure" of the quality-exercise, the management and the employees became skeptic about the possibility to improve daily practices. However, we came to the idea that a software QMS will not be really operational and useful if we would take into account the ISO 9001-prescriptions only. Research done in parallel confirmed our ideas. We learned about Fenton's basic entities ([11]): products, processes and resources. Before these entities would be understood, precise definitions would be needed (which seldom existed). Next to that we began to understand the relationships with the several software quality attributes. We placed all these elements into a matrix, representing in fact our "chosen quality framework" – a part of the later QMIM (see Figure 1).

We remarked the existence of many other approaches on software quality. Process oriented approaches completing ISO 9001, like CMM, Bootstrap, and SPICE have been studied. We learned about ISO 9126 and approaching software quality by the several product characteristics.

In 1995 IQSOFT won an EU PHARE tender "Technology Development and Quality Management", thus gaining financial support for quality oriented activities. That was the beginning of the so called IQPM^2 project, which used PM^2 methodology of Lucas Management Systems in developing a customized PM system.

The project started in February 1996 and finished in May 1997, and we can state that it was successful: it reached its goal within the planned time and budget limits. Besides the planned ones (developing and introducing a PM system based on the needs of the company) the IQPM^2 project produced a series of side results, which, from the company's long-term perspective, were even more important than the planned ones.

We understood that the processes of a software company could be - in our case: should be - divided into at least two distinct types: *project management processes* and *technical processes*. Projects can be modeled according to both activity types, so project management models and project type models can be situational configured. We noticed that project management activities in IQSOFT were more stable than the technical ones, which justified again their separation. With the standardisation of project management activities –building the (*unique*!) project management model of the company we made the first important step towards bringing order in the company. At that moment we *consciously left the technical activities undefined.* Following this argumentation, *we regrouped Fenton's entities into the following objects: Project Management (PM), Technical Process (TP)*, and *Product. (P)*. Basic reason for this regrouping was the fact that the objects we were talking about were not Fenton's "narrow" objects, but were more business objects. Fenton's "resources" were incorporated into project management, because we considered that all resource-related subjects were addressed within the project management issues. This way, QMIM framework was refined further.

According to a Bootstrap assessment carried out in 1997, the overall organisation had the maturity level 2, while the pilot projects reached 2.50 in CMM. See the results related to some process areas in Table 1.

With the positive experiences of PM^2, IQSOFT's management decided to go again for ISO 9001 registration. On this basis a project was started to obtain registration. It can be called a "new approach" project because it had the scope of obtaining ISO 9001 certification using all former experience of IQSOFT in building a *customised QMS*. The project was declared a top-priority one, having an internal effort of 250 man-days.

Table 1: Results of Bootstrap assessments

Area	1994	1997
SPU	2.25	3
Process description	2	3.25
Process control	1.5	3
Project management	2.75	3.5
Development model	2.5	3
Detailed design& implementation	2.25	4

The quality management system was fully operational beginning with February 1998. The final audit for the registration took place in April 1998, and it was successful. It is important to show the structure of IQSOFT's internal QMS: it followed the recommendations of the ISO standards, but it was built *to fit the specific needs of the company itself.* QMIM concepts were used throughout this project, the company making the conscious decision to concentrate on processes.

The QMS built was used actively, but IQSOFT did not want to stop quality-related activities on ISO 9001 level. Having in mind the QMIM framework, we decided to concentrate on elements not taken into account so far. Literature (theory of measurement, the Quality Improvement Paradigm, GQM, Experience Factory Organisation, e.g. in [11], [17]) shows that *measurement* has to be done to assure that quality of each object of software production is of the requested level. The QMIM framework was completed with the "metric" element, an important item "telling something" about the quality of the objects.

2.2. Broadening the scope of software quality management: ISO 9001:2000

As IQSOFT's first ISO-certificate was valid until April 2001, switching to the new ISO 9001:2000 standard ([6]) with the renewal of the registration was the obviously market- requested step by that moment.

A project was launched to build up an ISO 9001:2000-conform quality management system (QMS). The QMS the company was using for 3 years has undergone some major changes. The quality procedures referred from that moment *to all processes and departments of the company* (marketing, financial processes, human resource management were included). Previously existing procedures have been updated, understanding and following customer needs were emphasized, customer satisfaction-measurement has been started. Quality goals have been formulated, a measurement program to follow their realization has been established. Projects started to develop concrete quality plans.

As a result of the project, the company obtained the ISO 9001:2000 certificate in spring 2001, but the consequences of applying the new standard were more, in

terms of changes in software quality management. This process is presented in the next chapter, and is described in detail in [23].

2.3. Quality life after ISO 9001:2000

It became more obvious - it was explicitly stated - that quality management was not a separate process but rather *an aspect of the management processes*.

Being obliged to fulfill the standard requirement about setting quality goals and establish a metric program to measure them *guided the company towards connecting quality goals to business goals*.

It is interesting to analyze the changes in setting the quality goals of the company. If we look back to connecting quality goals to business goals while having in mind the Balanced IT Scorecard framework (BITS, see [21], [22]), we can notice that the quality goals of IQSOFT have been grouped in fact according to the elements considered important in the BITS. We established quality goals related to: financial issues, customers, people, processes, infrastructure and innovation elements.

To show the links between quality goals and business / strategic objectives of the company, we can use a customized form of a BITS-based representation.

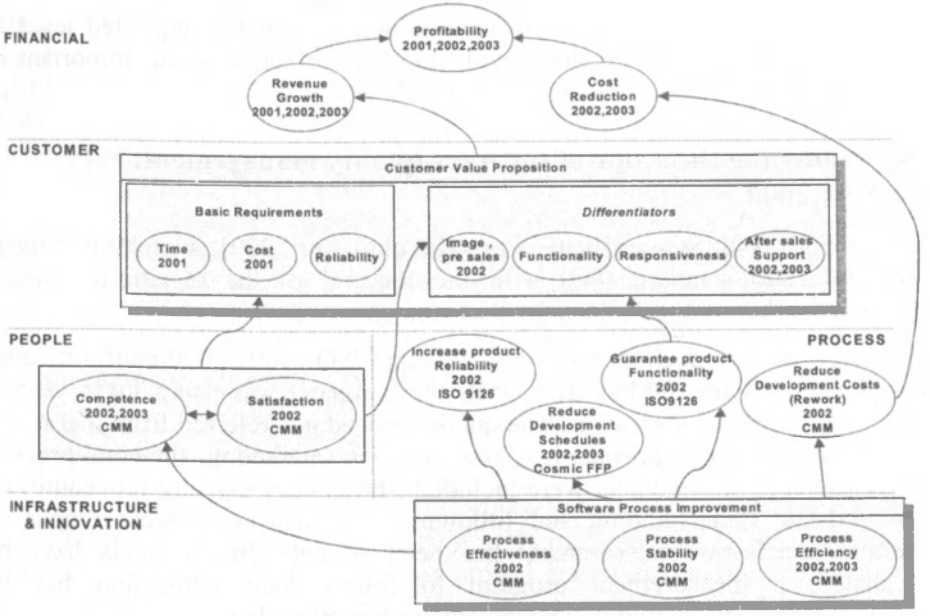

Figure 3: Connecting quality goals to business goals

In Figure 3 we show the elements contributing to successfully execute a strategy, using a representation suggested in [21], marking also the year in which quality goals associated to each element were present in our company. The years are connected also to the moments when the usage of models different from ISO 9001:2000 emerged.

Looking to this picture, some important remarks can be made. First, it is obvious that quality goals in the first year (2001) were rather stereotypical (basically related to financial issues), while in the next years the company started to set quality goals more and more deriving from real business needs. Next, one will notice that *using further software quality models*, besides ISO 9001:2000, *appeared as a business driven quality goal*. Finally, the QMIM framework was at hand to aid the conscious choice for further quality models.

Since in 2001 there was no precise understanding about why and how measurement should be done, only project management- related data gathering was started (planned and actual time, cost and effort of projects were recorded). By 2002, measurement provided some data that, although not sufficiently accurate, guided the attention towards problematic areas of the company's activity.

2.3.1. Product quality issues

The biggest problem was considered to be the huge difference between planned and actual effort of projects, forcing us to face that our estimates were not accurate enough. The wish to make them more accurate resulted in several quality goals for 2002, related basically to *software product quality* and *software process improvement*. In Figure 3 these appear in the elements related to increasing product reliability, guaranteeing functionality and starting software process improvement.

The understanding that IQSOFT was using in fact only one of the possible process oriented approaches – ISO 9001:2000 - towards software quality was there, while other possibilities in choosing appropriate models for different important software-quality–elements were aided by the QMIM framework. This way, the need to use further quality models for further important elements of software production came natural to the company.

In the wish of having more accurate estimates the company came across the *differences between products built for different end-users*.

The need for a better understanding of product types raised, therefore we tried to define the most important *product quality characteristics and metrics* (quality goal for 2002). We formulated general guidelines based on ISO 9126 ([5]), offering a menu of possible quality attributes and metrics, from which every project manager would choose the ones most fitting to his project, and, implicitly, define the quality profile for the type of that product.

On the other hand, the wish to have more accurate estimates drove IQSOFT to trying to connect project effort to the *complexity of the software developed.* Complexity was expected to become an extra element within the criteria used to define product types.

Among the methods for function point counting / software sizing ([18],[19],[20]), Cosmic FFP ([18]) was chosen, as it promised to give good results both in case of business applications and real time applications. This appeared as a product-related and process improvement-related quality goal in 2002. The sizing project was later incorporated into the CMM-based software process improvement project started in September 2002 (see 2.4).

2.4. The CMM –based software process improvement project

The fact that the majority of our problems presented before (estimation, defining and managing product size and quality, increasing process efficiency, managing the knowledge of human resources) could be regarded within the framework of one well known model, the CMM, was continuously promoted by keeping QMIM-principles "alive" The management-level recognition came in 2001, when an informal assessment was performed at the company by the European Software Institute. According to its results, in 2002 we already had the quality goal to run an SPI based on CMM (appearing in Figure 3 within software process improvement). The first, informal assessment resulted in a report and several improvement opportunities, according to which the company started a global process improvement project, planning to get certified according to CMM level 3 by July 2003. The high level results of the assessment are presented in

Table 2. As one can see from the table, the informal assessment at IQSOFT produced results similar to "best case profile" found in such assessments. It seemed a feasible main goal to reach CMM level 3 within a reasonable period. An SPI project was started for this purpose, planned to last 367 days and to use an effort of 800 man-days.

The project activities were possible to group into several groups of tasks. One task necessary throughout the entire life cycle was management of the CMM project. The next big group was developing and introducing the procedures required by CMM. Two basic activity types had to be performed: development and introduction of management procedures and development and introduction of technical procedures. The *management procedures* were concerned with the CMM KPA-s related to this type of activity.

To fulfill SQA KPA, the basic issue was to develop the quality management phased to projects. Within the PM related issues, the already existing planning, tracking and oversight procedures had to be updated to fit CMM requirements. The definition of the estimation procedure was not that easy due to those presented

in the previous chapter, but a procedure has been developed. IQSOFT's resource management, TP (Training Program) and IC (Intergroup Coordination) processes were basically good, but we did not have any peer reviews and the SSM (Software Subcontract Management) procedure had also be substantially updated.

Table 2: CMM assessment results of IQSOFT compared to the "best case profile" of an ISO-certified company

ML	Key process areas	IQSOFT assessment result	ISO best case profile
5	Process change management (PCM)	(Not rated)	Partially satisfied
	Technology change management (TCM)	(Not rated)	Partially satisfied
	Defect prevention (DP)	(Not rated)	Partially satisfied
4	Software quality management (SQM)	Not satisfied	Partially satisfied
	Quantitative process management (QPM)	Not satisfied	Not satisfied
3	Peer reviews (PR)	Not satisfied	Fully satisfied
	Intergroup coordination (IC)	Fully satisfied	Fully satisfied
	Software product engineering (SPE)	Partially satisfied	Fully satisfied
	Integrated software management (ISM)	Not satisfied	Not satisfied
	Training program (TP)	Fully satisfied	Partially satisfied
	Organization process definition (OPD)	Not satisfied	Not satisfied
	Organization process focus (OPF)	Partially satisfied	Not satisfied
2	Software configuration management (SCM)	Fully satisfied	Fully satisfied
	Software quality assurance (SQA)	Partially satisfied	Fully satisfied
	Software subcontract management (SSM)	Not applicable	Partially satisfied
	Software project tracking & oversight (SPTO)	Partially satisfied	Partially satisfied
	Software project planning (SPL)	Partially satisfied	Partially satisfied
	Requirements management (RM)	Fully satisfied	Partially satisfied

Development of a project data base was regarded as an outcome of the previous tasks. We proposed the structure described within QMIM (see Figure 2), and a first draft of the database, the process model of the company has been worked out as presented in Figure 4.

From the figure we can see that the product (Termék) is in the center of the attention, all processes: marketing, sales, managerial processes (Menedzsment folyamatok), technical development processes (Műszaki folyamatok), quality assurance (Minőségbiztosítás), support, subcontractor's work (Alvállalkozók kezelése) are executed around this item. The product development is aided by further processes i.e. human resource management and training (Hr, képzés), system engineering (Rendszergazdai folyamatok), secretarial processes (Titkársági folyamatok), financial processes (Gazdasági folyamatok). Further connections between different processes exist (e.g. quality management is connected to every

other process). The product and the processes all have associated quality attributes and metrics (Minőségi attribútumok és mérőszámok).

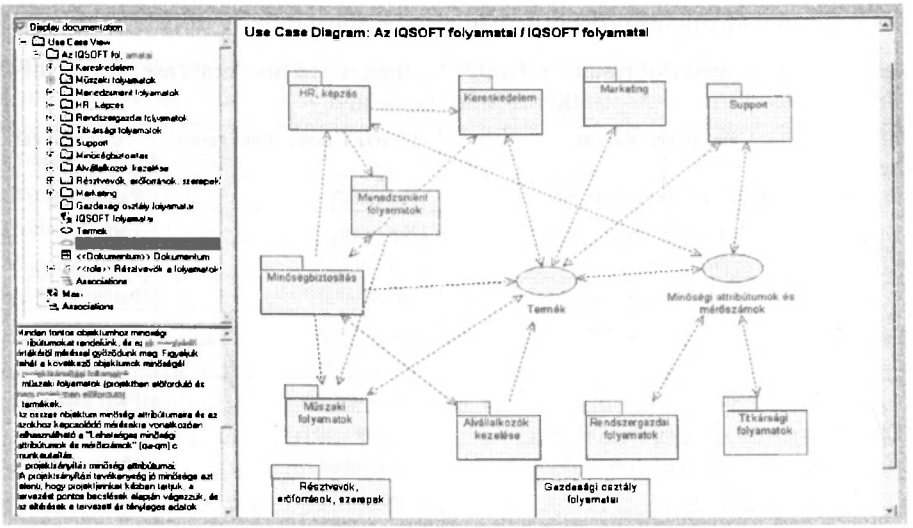

Figure 4: Use case diagram of IQSOFT objects related to quality

The presented Rose model contained all handbooks, templates, guidelines existing at that moment. Figure 4. shows in the bottom left corner the guidelines for determining quality attributes and metrics.

If we compare the structure of the Rose model to the structure of a QMIM-database suggested in the QMIM model (see Figure 2), we can see that they are in fact similar.

In the CMM-based SPI project the part connected to *development and introduction of technical procedures* caused most difficulties to the company. While RM and SCM were rather clear regarding the requirements to be fulfilled, it was not obvious how project types, project life cycle models had to be differentiated.

Based on the opinion that projects were integrating elements of different life cycles and different technologies, technicians planned to decompose the projects into smaller (basic) elements, and to develop a matrix that would show all possible associations between basic tasks, technologies, running environment, development tool, reference project.

The other important difficulty we encountered was connected to *product management*. The need to have well defined product types emerged again. We used our previous sizing results to define product types. The idea was to define a quality profile for every project type, using the previous experience described in 2.3.1.

2.5. 2003: the year of change

General recession has not left the (Hungarian) IT sector untouched. The KFKI group – whom IQSOFT has joined in 1999 – had a decentralized structure, that no longer met the new challenges. The greatest problem of the organizational structure model the group has followed so far was the significant overlaps between the business activities of different member-companies. In 2002 the Holding had 1 consulting company, 4 software development and integration companies, 2 IT infrastructure building and safety-issues –related companies and 2 IT application and service providers, 2 companies of them being involved in more than 1 of the business areas mentioned above.

As a consequence of the IT market regression, the organizational structure of the KFKI Group was greatly simplified in early 2003. The aim was to build one big company in each of the existing business areas. Companies doing software development IQSOFT, CLASSYS, a part of ICON, and a part of ISIS were merged to create the largest Hungarian company in software application development and integration, named IQSYS Ltd. As IQSYS was the successor of IQSOFT in legal terms, its quality management system was built around IQSOFT's former QMS, integrating all the good practices, procedures, methods of the other companies into it. QMIM principles proved to be a good aid towards finding a common language: we regarded PM as being the common framework for all projects, possible to define and implement in short time, while the definition of the technology-specific technical processes was left to the next level. This way, it was possible that the departments use their previously defined processes, company-wide agreement being needed only on PM issues.

Based on these, the QMS has been developed, introduced by mid May 2003. In June 2003 IQSYS was certified by SGS and was entitled to carry on IQSOFT's former ISO 9001:2000 certificate.

The CMM-based SPI project was declared to survive the organizational change, and the quality goals set for 2003 contained this issue. However, despite the initial plans, no SEPG group was established, and software quality made a step back compared to IQSOFT-situation, as its basic scope was to integrate and keep operational the quality systems of the former companies, integrated in the QMS of IQSYS. CMM-based SPIU might continue, after IQSYS stabilized its organizational structure – which was the aim of the company for year 2004.

3. CONCLUSIONS

The first chapter of the article focused on presenting a theoretical framework that can be used to understand the important elements of software quality. The presented QMIM framework has more aspects: the static aspect helps in connecting objects and characteristics of software quality, while the guidelines help using the

model in different organizations. The most popular software quality models, standards, methods can be placed within QMIM. Moreover, the framework helps in choosing the right approach, starting from the concrete needs of the company. The framework emphasizes the idea that using one software quality model can be misleading, while the synergic use of more quality approaches can be the best solution.

In the second chapter we presented the main results of a 11 years long case study, recorded at a Hungarian software company. We described the most positive results obtained by IQSOFT in the field of software quality management, while having QMIM in mind. We emphasized the necessity of consciously choosing the right elements and approaches at a certain moment, taking into account the maturity of the company. Connecting quality goals to business needs is extremely important. The evolution of the quality-related activities followed the principles of the QMIM model: the company started SPI by organizing its project management processes, followed by the technical processes. Issues related to product and human resources has followed as a natural requirement on a certain level of maturity. Using more software quality models at the same time was also a must resulting from concrete business needs.

The QMIM framework that resulted in fact from the SPI and research, is usable in other software companies also.

REFERENCES

[1] BALLA, K.: *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

[2] BALLA, K., BEMELMANS, T., KUSTERS, R., TRIENEKENS, J.J.M.: *The QMIM model.* Software Quality Journal, Volume 9 Number 3, November, 2001, Kluwer Academic Publishers. ISSN 0963-9314, pp. 177-193.

[3] BOEHM, B.W.: *Characteristics of software quality*, North Holland, 1978.

[4] MCCALL, J. A.: *The utility of software metrics in large scale software systems development*, IEEE Second software life cycle management workshop, August 1978.

[5] ISO/IEC 9126. *Information technology Software product evaluation Quality characteristics and guidelines for their use.* See also: ISO IEC 9126-1., FCD 9126-1.2.,. ISO/ IEC 9126-2,. ISO / IEC 9126-3:Jan. 18, 1999.

[6] ISO 9001:2000: *Quality management systems. Requirements*. December 2000.

[7] *The Capability Maturity Model. Guidelines for Improving the Software Process*. SEI Serties in Software Engineering. Addison-Wesley, March 2002. ISBN 0-201-54664-7

[8] KUVAJA, P., SIMILA, J., KRANIK, L., BICEGO, A., SAUKKONEN, S., KOCH, G.: *Software Process Assessment & Improvement. The Bootstrap approach.* ISBN O-631-19663-3, Blackwell, 1994.

[9] ROUT, T. P.: SPICE *A framework for software process assessment.* In: Software Process - Improvement and Practice 1(1), pp. 57-66, August 1995.

[10] CMMI 2001. *By Carnegie Mellon University Capability Maturity Model Integration,* Version 1.1. Continuous representation. Staged representation. December 2001.

[11] FENTON, N.E.: *Software metrics – a rigorous approach.* Chapman&Hall, 1992.

[12] HUMPHREY, W.: *Using a Defined and Measured Personal Software Process.* In: IEEE Software, May 1996, pp. 77-87. http://www.sei.cmu.edu/tsp/psp.html

[13] *Team Software Process.* http://www.sei.cmu.edu/tsp/tsp.html

[14] BENTLEY, C.: *Practical PRINCE2.* ISBN 0 11 702853 3. 2002.

[15] WEINBERG, GERALD M.: *Quality Software Measurement.* Vol.1.: System Thinking. Dorset House Publishing, ill. 1992. Vol.2., 1993. Vol.3.:, 1994.

[16] P-CMM People CMM. http://www.sei.cmu.edu/cmm-p/version2/

[17] BASILI, V.: *SEL's Software Process Improvement Program*, IEEE Software, Nov. 1995.

[18] ISO/IEC 19761:2003 *Software engineering COSMIC-FFP A functional size measurement method*

[19] ISO/IEC 20926:2003 *Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting practices manual*

[20] ISO/IEC 20968:2002 *Software engineering Mk II Function Point Analysis Counting Practices Manual*

[21] REO, D.: *The Balanced IT Scorecard. Quality of Strategy Vs. Strategy Execution.* In: Proceedings of Business Information Technology Management Conference, BITWorld 2001, Cairo, Egypt

[22] REO, D.: *Balanced IT Scorecard.* ESI Commercial Network presentation. 2000.

[23] BALLA, K.: *Software Process Improvement and Organizational Change.* In: 50 jaar informatiesystemen 1978-2028. Liber Amicorum voor Theo Bemelmans. pp. 181-198. March 2004. Edited by Capaciteitsgroep Infomation Systems, Faculteit Technologie Management, Technische Universiteit Eindhoven. ISBN 90-386-1928-6, NUR 982