Abstract

The article focuses on a problem software companies often face: the need to choose a software quality model that, besides being auditable, ensures the connection of quality goals to the business goals of that company. A good solution seems to be to quit relying exclusively on one quality model and use more models in a synergic way instead. We explain a possibility to handle this problem by using a framework that helps positioning the existing software quality approaches and standards according to the basic elements of software production they address, and gives guidance in the sequence of possible steps to do really efficient software process improvement. We present the elements of the framework and a possibility to use it to implement certain CMMI requirements.

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 produce an official certificate about usage of a certain standard or model. However, introducing an approach based on a standard or model, and institutionalizing it, so that the organization is able to pass an audit, requires a lot of investment from 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.

In our opinion, efficient improvement programs are always based on real needs of companies, will always start from understanding the actual situation. 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? In which direction to move if the company wishes to reach higher software quality?

For answering these questions, we have to understand what software quality means in each particular situation- as there is no universally definable “good quality”. Based on this understanding, the right quality profile has to be built – which means defining the most important objects and characteristics of software production in that particular situation.

In the beginning of the article (2.) we address the complex subject of software quality by describing the main objects we have to deal with and their basic attributes, as well as some of the most popular software quality models. We position these models against the objects of software production and we point out the need to quit exclusively relying on one model in favor of using more quality models in a synergic way.

In chapter 3., we develop further the idea of important objects and attributes software quality is dealing with by describing QMIM, a framework that helps positioning the existing software quality approaches and standards, and gives guidance in the sequence of possible
steps to do really efficient software process improvement. We present the elements of the
framework.
In the end of the article (4.) we show a possibility to use QMIM to implement
Organizational Process Performance, a complex process area of CMMI (Capability Maturity
Model Integration).

2. The complex subject of software quality

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 products, the processes that produce the products and the resources
that execute the processes (these elements appear also eg. in Fenton [1]). We have to
define these objects, to choose the right quality attributes for them and verify the actual
value of these attributes by the means of objective metrics.

The approaches, standards and models used in software industry are extremely
various in their approach used.

The first approaches used (Boehm [1] and McCall [2] models) were connected to
software product quality. The fact that definition of objective quality attributes and
metrics (emphasized by these models) turned out to be rather complicated, surely
influenced the software community to look for approaches “easier” to apply and assess
in business life. The idea that if the processes are “good”, they will produce “good
products”, therefore we need to concentrate on organizing and managing the processes
of software production, became very popular in the nineties. The processes of software
production are being very emphasized by approaches like ISO 9001 (its actual version
being ISO 9001:2000 [3]), CMM [4], SPICE [5], ISO 12207 [6], by project management
methodologies and software development methodologies. A big advantage of the
process oriented approaches is that they are auditable, and the certificate of an audit is
well usable in business.

The problem with process oriented approaches is that, if used just “verbatim”, they
might leave software product quality out of consideration. Therefore, software
professionals needed to combine process oriented approaches with approaches
concentrating on software product characteristics and metrics. For this purpose ISO
9126 family [7] was developed, using the experience of earlier Boehm and McCall
models in a simplified way. Answering the need to make software product more
emphasized in process oriented approaches, new approaches and new versions of elder
approaches appeared (like CMMI [8] or ISO 90003 [9]), that, besides the process-
oriented characteristics, introduce elements of product quality as well.

Nowadays approaches concentrating on different aspects of measurement (like GQM
[10] or function point counting methods) are also known, and models emphasizing the
importance of the human factor (e.g. P-CMM [11], PSP [12], TSP[13]) become more
and more popular, also.

Summarizing the statements made before, we can represent the basic objects of
software production and their important attributes in a framework. If we place the most
popular quality approaches into this framework, we obtain a picture like the one shown in
Figure 1. It is to the discretion of the reader to understand that if we wish to deal
with software quality in its complexity, we have to complete the entire framework
shown in Figure 1.
From this picture we can draw 2 basic conclusions:

- The number of the approaches dealing with software quality is large, many of them overlapping in the coverage of some elements.
- There is no approach that would deal with all important objects of software quality.

The first conclusion tells us that it is extremely important to understand the elements covered by a certain approach.

The second conclusion leads to the idea that as no approach, model or standard covers all the 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), companies will have to choose the right combination of approaches based on their business needs. This means to quit exclusively relying on one certain quality model - in favor of choosing among several approaches, using more approaches in a synergic way, according to the specific business needs of a certain organization. Such a combined use of more quality models or approaches would be based, on the one hand, on a solid theoretical knowledge and understanding of the popular quality models and their interconnections, and, on the other hand, on a huge experience in the way the companies in case execute their development processes and do business.

For the moment, there is no guideline, aid or description in literature that would tell a company just starting to deal with software quality improvement what objects to choose, to what approach to connect them, what would be a “right combination of approaches” in their case and in which way to proceed if they want to do a real efficient software quality improvement.

We faced these problems in our day to day work, and we developed the framework presented in chapter 3 to help solving this problem.
3. QMIM – Quality through Managed Improvement and Measurement

QMIM – Quality through Managed improvement and measurement (described in detail in [14]) is a framework that helps software developing companies to find their way among the many models, standards, methods connected to software quality, and to efficiently use the basic concepts of 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 best fits own needs.

3.1. Elements of QMIM

Basic elements of QMIM would be the objects and attributes shown in Figure 1: product, process, resource on the “objects” dimension, and definition, quality attribute and metric on the “characteristics” dimension. If we agree on the fact that processes in software production can be divided into project management processes and technical processes, we will have the elements product, technical process (TP), project management process (PM) and resource on the “objects” dimension. As resources are dealt with usually within project management processes, we can regroup the objects to have product, TP and PM on the “objects” dimension. However, this is a suggestion only. When using QMIM, we are free to leave all objects separated.

The most important requirement at this moment of “getting started” with QMIM is to understand the important objects and attributes of software production and to define them unambiguously in the specific environment of the user. QMIM description ([14]) gives the basic elements of the framework, together with their suggested definition. When using the model in a concrete environment, one will have to analyze the actual situation and, maybe, re-define some of the elements.

As an example, we present the definition and explanation used by us in QMIM detailed description for PM and TP.

“Project management (PM) is the totality of activities carried out for the successful completion of a project. These activities are concerned with assuring the frame, environment that makes it possible to complete the project within the time and budget planned. Using the available resources in an optimal way, in order to produce the planned result of the project with the desired quality attributes are further goals of project management. Project management is done by people, who have the task to plan, follow and control the project. Literature and personal experience show that project management activities are the most stable ones in software organizations, therefore they can serve as a good starting point in the process of standardization.”

“Technical processes are the processes that are connected to the technical work done in a software project. They can differ from project to project, according to the specificity of the work done. We consider technical process for instance, the process of a system analysis, the definition of the requirements, developing a database system, making a system integration and so on. Technical processes (normally) are integrated in the project management processes. Technical processes are executed by the "technical people" (analyst, programmer, tester…). Technical processes can be aided by certain methodologies (e.g. system development methodologies), which give guidance in the activities to be done, and the sequence of these activities. An organization can use existing methodologies, but can develop own methodologies, as well. “

Similar to the examples presented, all elements (objects and characteristics) of QMIM are defined in [14]. Users can use these definitions or can make their own
definitions (E.g. they need to define what is a “software product” in their environment and what characteristics does is have. By the way, this is one of the most difficult tasks in terms of definitions, but it will help a lot when product quality profiles have to be built in the software development projects.)

3.2. QMIM data model

While executing the activities suggested by the QMIM framework, one will generate all kinds 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, we consider it appropriate to have a static data model / view of QMIM framework, which will guide the company in storing the different data. This is the QMIM data model.

In storing the data, two directions are possible. We might store specific data first (data related to a certain, concrete project, technical process or product), and, after understanding the objects and their interconnections, we might want to record the generic data (which characterizes the overall, managerial and technical processes and the products) as well. An other direction could be to start by recording generic data of the field (related to the object) first understood and modeled in the framework, attaching specific data later. In any case the generic data will give us suggestions, guidelines in forthcoming practice, while the specific data will constitute valuable historical data.

In Figure 2 we show a data diagram for the generic level. The specific level data will be stored in the same structure.

3.3. Dynamic aspects of QMIM

It is important to understand how QMIM is used in an organization. There are many elements that can differ according to the concrete situation: the elements defined, the
degree of detail of the definitions, the interconnections observed, the literature approaches or their elements chosen at a certain moment of time by an organization. The differences may appear due to the specificity of the organization using the model, due to the level of the organization's maturity, due to the degree of change accepted by the management and so on. Organizations change over time (e.g. grow in maturity) and in their different stages the model will be applied differently.

“Maturity” of an organization is a concept that is popular and frequently used in the software development community. We use the term “maturity” as defined by Paulk and Humphrey ([15] [16]). Putting it simple, a company is the more mature, the more of its processes are defined, documented, standardized over the organization, measured and continuously improved. In our QMIM description ([14]) we used CMM-maturity levels to position the companies in terms of their maturity. The actual trend, to use CMMI instead of CMM, can be easily applied in QMIM. In chapter 4. we present a way in which QMIM can help even in implementing CMMI.

3.3.1. QMIM guidelines

QMIM guidelines have been developed to help making the QMIM framework operational. They support the elements of the framework, and make them applicable in an organization. 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 build a company's own methodology to deal with software quality in a unified, balanced way. Using the guidelines, the company can build its own process model(s).

The guidelines are structured into 3 levels. The high-level guidelines give the main ideas about the introduction of a quality improvement program. The middle-level guidelines specify some details of the improvement program, using QMIM terminology. The low-level guidelines give advice about the activities to be performed in order to achieve the improvement.

High level guidelines refer to the importance of identifying the need for a change. Assessment of the current situation is needed, and a platform for the quality management activities has to be identified. Next, the desired change has to be made (at this level we do not insist on the nature of the change). The results of the change have to be seen, therefore adding a low level data gathering program is recommended. The change should be consolidated. Afterwards, the existing situation can be assessed, and a new change can be made. The data collecting program will become more and more complex as a set of changes is implemented.

Middle level guidelines start from the activities described in the high level guidelines. They present some of them into somewhat more detail. They insist on the nature of change to make and on the characteristics of the data gathering program. Regarding the nature of change, these guidelines tell the user to make a conscious decision about the direction of evolution. The decisions made (about the object of change and about the level of detailing the characteristics of that object) mean that the user decides upon the way of completing the QMIM reference framework. It can be completed horizontally or vertically (as shown in Figure 3.). The process of identification of the main objects is a complex one. It depends on the specificity of the company and its level of maturity. Therefore, the maturity of the organization should be assessed in a way that provides well understandable data. We suggest to do the assessment on the basis of CMM(I) model. This way, a number on the scale 1-5 is obtained, reflecting the actual maturity level of the company.
Figure 3: A possible direction of evolution in using QMIM

QMIM framework suggests that the objects and their characteristics can be defined in any organization, at any moment of time. The complexity of the definitions, quality characteristics and metrics will differ depending on the company’s maturity level.

However, our experience has shown that in a low maturity level organization (where CMM(I) maturity level is < 2) it is possible to start quality oriented work by concentrating on project management issues. Having a project management system in place will result in an increase of maturity level, bringing a company to level 2 in CMM(I). At this moment the definition of the technical processes can be done, resulting (most probably) in an ISO 9001-conform quality management system, with a maturity level increased to around 3. Afterwards, products can be defined, further increasing the maturity level. In all cases quality attributes can be defined and metrics can be associated to the defined elements.

Low level guidelines consist in a set of practical advice referring to the activities to be done and their sequence. Taking into account the above mentioned ideas, the low level guidelines can be grouped according to the high level and middle level guidelines they refer to.

In Table 1 we summarize the QMIM guidelines. Hi stands for “High level guideline number i”. Similarly, middle level and low level guidelines are marked by Mj and Lk, respectively.

| H1. Create a platform for the quality management activities. |
| H2. Assess the current situation. |
| L1. Understand SEI’s CMM(I) model. Place your company on the CMM(I) maturity hierarchy. An assessment method, e.g. SCAMPI C ([16]) or Bootstrap (if CMM, [17]) can be used for this purpose. |
| L2. Separate the project management activities from the technical ones. |
| H3. Make a change. |
| M1. Depending on the maturity level of the company, choose a direction for the change. Go to M1.1, M1.2 or M1.3, as indicated by the maturity level. |
| M1.1. If maturity is < 2, concentrate on project management. |
| L3. Investigate the existing project management methods, procedures, standards, way(s) of working at the company. In the following make use of the viable project management elements. |
L4. Identify the needs existing at the company in terms of project management. For the identification a structured methodology can be used (e.g. Pm’ methodology of Artemis International).
L5. Understand the concept of project management model and build the project management model for the company.
L6. On the basis of the needs defined in L4., develop and document the project management system at the company. For the development a structured methodology can be used. Go to H4.

M1.2. If maturity is = 2, concentrate on technical processes.
L7. Define quality characteristics of the project management processes.
L8. Define and start using metrics to measure project management characteristics.
L9. Use measurement results as a feedback for improving the project management system.
L10. Investigate the existing methods, procedures, standards referring to the technical activities at the company. In the following make use of the viable elements identified.
L11. Identify the needs existing at the company in terms of technical processes. Use interview techniques for the identification.
L12. Understand the concept of project type models and build the project type models for the company.
L13. On the basis of the needs identified in L11., develop and document the quality management system referring to all processes at the company. When developing the quality management system, understand and apply ISO 9001 requirements for such a system. The quality management system will refer both to project management and technical processes. Go to H4.

M1.3. If maturity is = 3, concentrate on products.
L14. If results of the assessment made are close to CMM(I) level 3, you can think of an ISO 9001 registration.
L15. Define quality attributes of the technical processes.
L16. Define and start using metrics to measure technical process characteristics. Continue using project management metrics.
L17. Use measurement results as a feedback for improving the quality management system.
L18. Understand the product types existing at the company.
L19. Define and describe the existing product types. This description will complete the existing quality management system.

H4. Use measurement results as a feedback for improving the quality management system. To define a measurement program, a methodology (eg. GQM) can be used.
H5. Consolidate the change.
L22. Continue using all QMIM concepts to refine your models and quality system.
L23. Data referring to the objects, concepts and interactions can be recorded in a database. Recorded data can be used in drawing conclusions and making estimations, predictions.

H6. Go back to H2 (assessment of the actual situation).

Table 1.: A summary of QMIM guidelines

4. Using QMIM to implement CMMI

One direction of evolution regarding QMIM is to make it CMMI-compatible. This means an analysis of the elements existing, some regrouping of the objects and, maybe, redefining some of its elements to ensure that they are in line with the nowadays most popular SPI-approach, CMMI. It is a subject of consideration eg. whether processes (grouped into managerial and technical processes) should be completed to be conform to process grouping (Process areas) used by CMMI: process management, project management, engineering and support process groups. At this moment, QMIM’s “project management” covers project management process area of CMMI, “technical processes” cover engineering processes in CMMI. Configuration management process, identified as a support processes in CMMI is regarded as a technical process in QMIM,
while Process and product quality assurance is belonging to PM in our framework. Measurement and analysis, causal analysis are emphasized in QMIM by suggesting to put in place a measurement program already on a low maturity level, and continuously enhancing and improving it. The most problematic are CMMI’s Process management processes, which do not have an explicit place in QMIM at this moment. However, Organizational innovation and deployment, Organizational process focus and Organizational process definition are emphasized by itself the “spirit” of QMIM. Our framework suggests, for instance, to have a common project management procedure for the whole organization, and to identify technical process type models which will be used by the different project types (this is equivalent to having a standard process and tailoring it according to the characteristics of the projects). According to the above, it is our actual subject under consideration to complete QMIM processes by adding Organizational processes.

In the following we show how QMIM helps already at a low maturity level in understanding and implementing Organizational process performance (OPP) process area of QMIM.

OPP is an advanced (progressive) process area among Process Management Process Areas. In staged representation it belongs to Level 4. OPP derives quantitative objectives for quality and process performance from the organization’s business objectives. It helps to quantitatively manage the project’s defined process to achieve the project’s established quality and process performance objectives. OPP makes it possible to accurately predict results of processes determine which processes need improvement to achieve quality targets, reduce variation in processes. It helps setting goals and managing actual values for product and service quality also. Properly using OPP, the organization will be able to: determine whether processes are behaving consistently or have stable trends (are predictable), have meaningful measures and realistic targets for important processes and products, predict what one can expect from a project (have realistic estimates, tell if a certain goal is likely to be reached or not), identify processes where the performance is within natural bounds that are consistent across process implementation teams, identify any aspects of the processes that can be improved in the organization’s standard set of processes, identify the implementation of a process which performs best... Specific goal of OPP is to establish performance baselines and models. This can be achieved by executing some specific practices: selecting processes, establishing process performance measures, establishing quality and process-performance objectives, establishing process performance baselines and establishing process performance models.

Our experience shows that OPP is a process area difficult to understand and implement. As it belongs to level 4 in staged representation, it is implemented in few places. We consider that its concepts, its “spirit” can and should be implemented already on lower maturity levels, “preparing” this way the understanding and approach needed to do continuous improvement in an organization.

Let us look to the first specific practice of OPP. “Selecting processes” has the meaning that important elements of software production need to be understood, next, the most important elements for that case should be selected based on the concrete business objectives of the organization. The priorities should be negotiated and past experience also has to be taken into account. Looking to QMIM elements (3.1) and QMIM guidelines (3.3) we will discover that these are the basic concepts in our framework, as well. Moreover, QMIM suggests that “important elements” should be
chosen from product, PM and TP, where definitions facilitate the understanding of the elements important for the company.

Next specific practice tells us to think about how the actual value of the important elements selected can be measured, where past experience also has to be taken into account. The measures used will evolve from measurement existing already at lower levels. The organization also has to be able to define what the acceptable values of the important elements are. Here, QMIM not only suggests the usage of a metrics program already on a low maturity level, but it also gives examples of possible quality attributes and measures for the different elements, and presents experience from literature describing “acceptable values” of the elements.

Measuring the actual value of “everything that has been chosen as being important” is strongly supported by QMIM. The database (3.2) built to store both the guidelines, literature experiences and concrete data resulting from actual measurement “prepares the ground” already in a low maturity level organization for having reliable data in order to be able to execute OPP in an efficient way.

5. Conclusions

In the article we pointed out the problem of choosing among quality approaches and integrating more models in a synergic way, faced by many software development companies. As a possible solution to this problem, we presented QMIM, and described its basic elements and characteristics. In the end we showed how QMIM can be used to implement the nowadays popular SPI model, CMMI.

6. References

[6] ISO 12207:
[16] Scampi