An experience in process assessment

Fabiano Cattaneo, Alfonso Fuggetta, Luigi Lavazza
Politecnico di Milano and CEFRIEL

Politecnico di Milano
Dipartimento di Elettronica ed Informazione
Piazza Leonardo da Vinci 32, 20133 Milano (Italy)
Tel.: +39-2-23993540; Fax: +39-2-23993411.
E-mail: {cattaneo, fuggetta}@elet.polimi.it,
lavazza@mailer.cefriel.it

Abstract

This paper presents an experience in software process assessment that has been conducted in a mid-size Italian company. The assessment has been carried out using the CMM (Capability Maturity Model) and taking into account also the indications offered by QIP (Quality Improvement Paradigm).
The paper discusses the results of the assessment and the lessons we have learned from running it. It particular, it argues that it is necessary to broaden the scope of assessment methods, and to evaluate the applicability to software processes of the experiences, methods, and techniques developed in other business domains.

Keywords and phrases:

Software process, software process assessment, process improvement, assessment methods.

1 Introduction

Assessment and improvement of software processes are important and critical issues for any organization which is directly or indirectly relying on software to assembly its products or to deliver its services [9]. This reflects a general trend which puts much emphasis on the quality of the process as an essential means to achieve a higher level of quality in delivered products and services. During the last 15 years, the terms "process", "total quality", "improvement" have become the key buzzwords used by many practitioners and researchers.

To support the achievement of these goals, in past years several assessment methods have been developed and used. In the software process arena, there are at least two approaches that have been conceived and that have received wide attention: the Quality Improvement Paradigm of the University of Maryland [2], and the Capability Maturity Model of the Software Engineering Institute [10]. The first approach emphasizes the importance of establishing a continuous improvement activity, based on a quantitative evaluation of the performances/characteristics of processes and products. The SEI approach provides a series of aids to assess the level of maturity of an organization and to suggest the most suitable improvement initiatives.

This paper presents an experience in using the CMM to support the improvement effort of a medium-size Italian company. It also discusses very briefly some issues related to QIP. The paper emphasizes the weaknesses of these approaches and raises some issues to the software engineering community on the desirable characteristics of process improvement methods. It complements other experiences about the usage of CMM, such as those by Brodman and Johnson, who have analyzed what small business organizations say about the CMM [4].

It must be clearly emphasized that the authors are not related to the SEI or to the University of Maryland. Also, we have not received any specific training on the CMM or the QIP. Our comments are based just on our personal experience, and on the knowledge of the above mentioned methods derived from publicly available documentation.

2 The company

The company where this experience has been carried out is a medium size Italian company (called XYZ in the remainder of this paper for confidentiality reasons), which provides advanced services to a large number of customers (several hundreds) distributed across the Italian territory. These services are offered through a large computer network, where customers' computers are interconnected to concentrators and then to main computer facilities at XYZ's headquarters. The services offered by XYZ are based on distributed software products, that run both on customers' and XYZ's computers. XYZ's services demand for a high

ICSE '95, Seattle, Washington USA
© 1995 ACM 0-89791-708-1/95/0004...$3.50

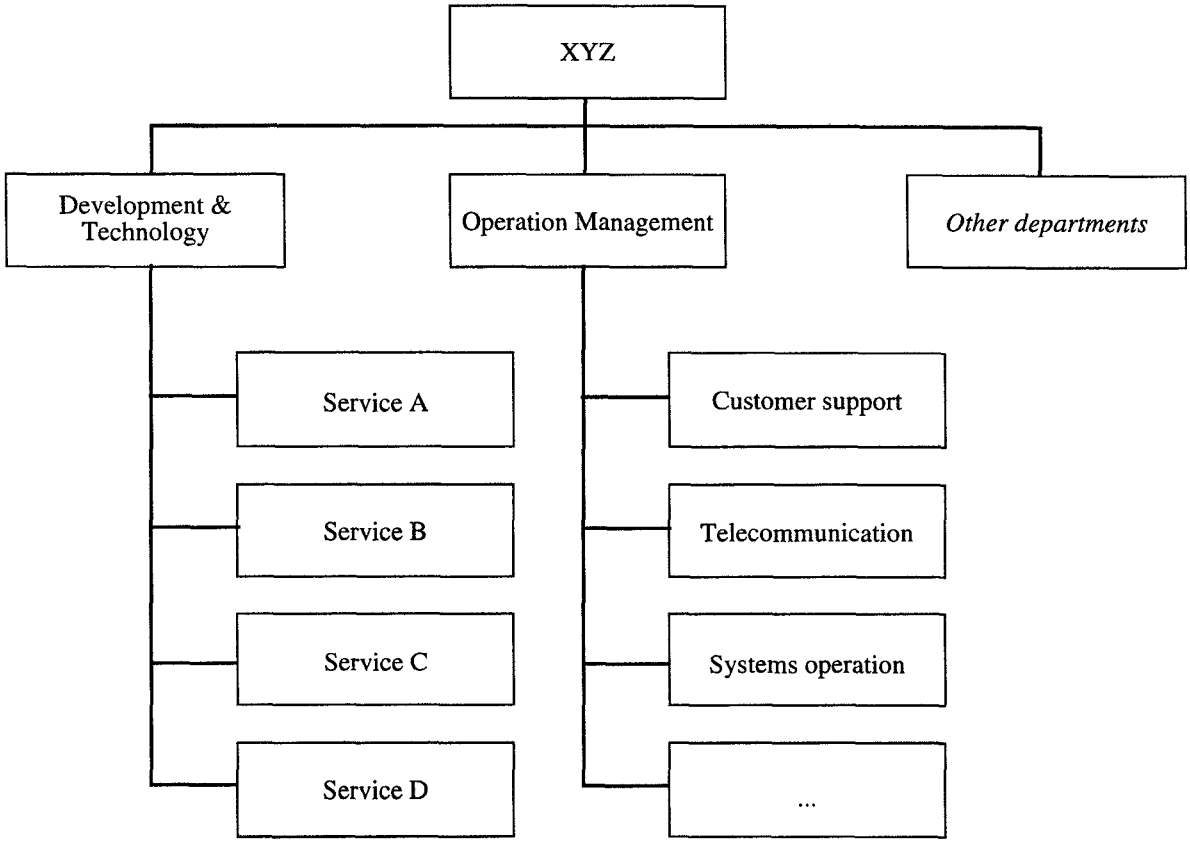

Figure 1: Existing organizational model at XYZ.

level of availability, and are characterized by strong performance and security requirements.

The software products used to provide these services are based on different platforms. For some services, traditional mainframes are used, while in other cases the platform that has been adopted is a combination of Unix servers and Macintosh clients. Other technologies are used to provide the needed networking facilities.

To develop the software products used to deliver these services, XYZ uses internal resources and external software houses. In recent years, however, the internal resources have been more and more used to run and operate software products (i.e., to control and manage the provision of XYZ services), while software development activities have been progressively delegated to external companies.

XYZ is organized in several departments according to a functional structure (see Figure 1). The most important departments are Development & Technology, the internal software house, and Operation Management, in charge of offering the services to customers. All development and maintenance activities are carried out by Development & Technology (possibly using external resources or subcontracting). Development & Technology is organized by services (and consequently by the set of software products used to offer a specific service, with a few projects spanning different services). Operation Management uses the products developed by Development & Technology to provide XYZ's customers with the requested services. It is organized by functions (customer support, telecommunication, systems operation, and so on). Operation Management does not have the possibility of accessing the source code. It completely relies on Development & Technology for any modification of software.

XYZ is experiencing several problems that can be summarized by the following observations:

1. Both the top and mid-level management, and the technical staff are mainly composed of people who do have a strong competence in the application domain, but a

weaker, self-made knowledge of information technology.

2. The requirements of the services to be offered (and consequently of the software products to be developed) are defined in most cases by a council of users, which has a very strong position with respect to XYZ. Requirements are changed quite frequently, both during software development and after the software product has been moved into operation.
3. Requirements and subcontracts management is often insufficient. XYZ receives software products from external software providers, but it does not have a complete knowledge of their structure and architecture. In many cases XYZ completely depends on its providers.
4. The strong requirements concerning service performance and availability put a lot of pressure on the Operation Management department, which must be able to guarantee a very quick response time (in some cases a few minutes) for any problem occurring to any software component in any point of the network.
5. Operation Management directly receives requests of changes from customers. It then negotiates with Development & Technology how to address them. This procedure is viewed as too slow and time-consuming.
6. It is difficult for Development & Technology to get accurate and timely requirements related to the usage and management of software products from Operation Management. On the other hand, Operation Management has no means to control that their requirements are properly understood and taken into account. In many cases, therefore, operators of Operation Management experience several problems in using and operating the systems released by Development & Technology.
7. Configuration management is not carried out according to company-wide policies. There are different platforms, where different and often undocumented policies are used. Even when configuration management is more consolidated (the mainframe area in Development & Technology), there are other problems related to the tools that are used and, in particular, to the policies used to share information and software between Development & Technology and Operation Management (i.e., software developers and main software users).

To address the problems XYZ is facing in producing and using software, XYZ's top management requested CEFRIEL[1] to conduct a software process assessment of the company and to suggest a reasonable improvement strategy. To base the assessment on a consolidated methodology, we decided to adopt the CMM approach. Moreover, we wanted to evaluate the applicability of the QIP as well, and therefore we have also taken into account the indications offered by this method.

SEI claims that CMM can be used to address two different goals [10]:

1. to evaluate contractors/bidders (*Software Capability Evaluation*);
2. to guide and support a process improvement initiative (*Software Process Assessment*).

We have used the CMM to support the accomplishment of the second goal. The next section discusses in detail how we proceeded and the results of this experience.

3 Using the CMM questionnaire

We have followed the procedure indicated by the SEI to conduct a software process assessment based on CMM Ver. 1.1. We used the new questionnaire that was recently released even though the scoring method was not yet available. Indeed, our goal was to identify the good and weak practices of the company, and not to compute a score. Basically, we have organized our assessment activity in the following way:

1. We created an assessment team composed of 3 people from CEFRIEL and 2 people from XYZ. The team studied the official SEI documentation.
2. We had a couple of preliminary meetings with the mid-level management to describe the goals of the assessment and the procedure we would be following. Moreover, we wanted to hear from them their view of the company and their evaluation of XYZ problems. Some of the basic results of our assessment were already quite clear after these preliminary meetings.
3. We selected a set of people with different skills/positions from several projects/groups. Then we asked them to fill up the CMM questionnaire under our direct supervision. Before doing that, we informed them on the goals of the initiative and on the methodology we would be using.
4. We asked additional questions to the interviewed people, based on the first results derived from the initial meetings with the top and mid-level management. We made clear from the very beginning that we would not disclose their identity.
5. We derived our assessment results by considering the evaluation produced by CMM, and our findings based on the discussion with the top management and the interviewed people.

Our experience in using the CMM questionnaire can be summarized as follows:

1. CEFRIEL is a research center operating in Milano. Its activities are advanced R&D, education, and consulting services in the field of Information Technology.

a. Most people found the questionnaire quite *repetitive and verbose*. Many questions are just obvious when correlated to previous answers.

b. Almost all the interviewed people find that many questions were not related to the real problems of the company. For instance, they declared that major problems are in the distribution of human resources among departments, in the organization of the company, and in the company education policy. Most of them knew that they were performing poorly as far as key practices are concerned: for example, they were perfectly aware that they were unsatisfactorily doing configuration management and requirements management. *They wanted to know how to find the resources to do it, and how to introduce these changes in the organization avoiding any turbulence on the customer side.*

c. As members of the assessment team, we found that *it is quite easy to determine the maturity level of the company*, even after a few meetings. That is not the critical part of the work.

In general, the CMM did not allow us to discover those information that were really critical and useful in defining the XYZ improvement strategy. In our opinion, the problems are related both to the methodology suggested by SEI to conduct the assessment (basically, the questionnaire) and the contents of the CMM itself. We believe that the CMM is based on a valid assumption, i.e., improvement of product quality must be based on improvement of process quality. But it then (at least partially) fails to address this goal, by limiting the scope of process assessment and improvement to the engineering and technical component of the process. This is not enough, as we discuss in more detail later on.

4 Results of the assessment

We have identified the following action list to support process improvement at XYZ.

High-priority steps:

a. Reorganization of the company by services, through the merging of Development & Technology with Operation Management (see Figure 2)[2]. The rationale for this choice resides in the characteristics of XYZ's customers and its marketplace (see Section 2). They demand for strong interaction between final users (i.e., users of XYZ's services), software developers, and operators of a specific software product (i.e., service providers). Moreover, it is necessary to integrate operators' requirements into products more consistently, systematically, and from a very early stage of software design. Consequently, for every service (or class of comparable services) we plan to have a group in charge of listening to customers' needs and to deliver them what they require. Each group will take care of service design, software development (including subcontractors management), product maintenance, and service operation.

b. Intensive education programs and personnel requalification. This is needed to support the adoption of more advanced software engineering methods, techniques, and tools.

c. Hiring of new resources and redistribution of existing personnel among departments/areas. Reduction of body shopping and of other external development activities.

Other steps to be accomplished in a second phase are the following ones:

d. Introduction of corporate procedures for configuration management.

e. Introduction of corporate procedures for requirement engineering and management.

The (surprising?) result of our experience is that the top priority steps in the above action list were not derived from the results of the CMM-based assessment.

5 Lessons learned

The key lesson we learned is that while it was reasonably easy to identify the technical areas where improvement is needed (i.e., configuration management and requirements engineering), it is much more difficult to understand how to conduct the *analysis* of a company in order to identify its *organizational and strategical deficiencies*, and how to identify and propose *reasonable changes and improvement actions* accordingly. In the XYZ case, the main problems are the functional organization (software development separated from software operation), the insufficient technical quality of resources, and their scarcity or unsatisfactory distribution among departments. While the CMM would have led us to consider only the internal software house of the company, some of the problems we found are outside or on the boundaries of Development & Technology. If these problems are not solved first, any (in many cases obvious) indication deriving from the CMM is useless. Even more, it can be counterproductive. For instance, a configuration management policy for a functional organization will be quite different from the one you would adopt in an organization by services. Indeed, the introduction of a policy defined according to the existing (and inadequate) organizational model can even make the situation worse. Even an evolution of CMM such as the Bootstrap approach does not introduce substantial changes with respect to the original model, as far as the scope of the assessment is concerned

2. XYZ actually adopted the suggested organization, effective January 1995.

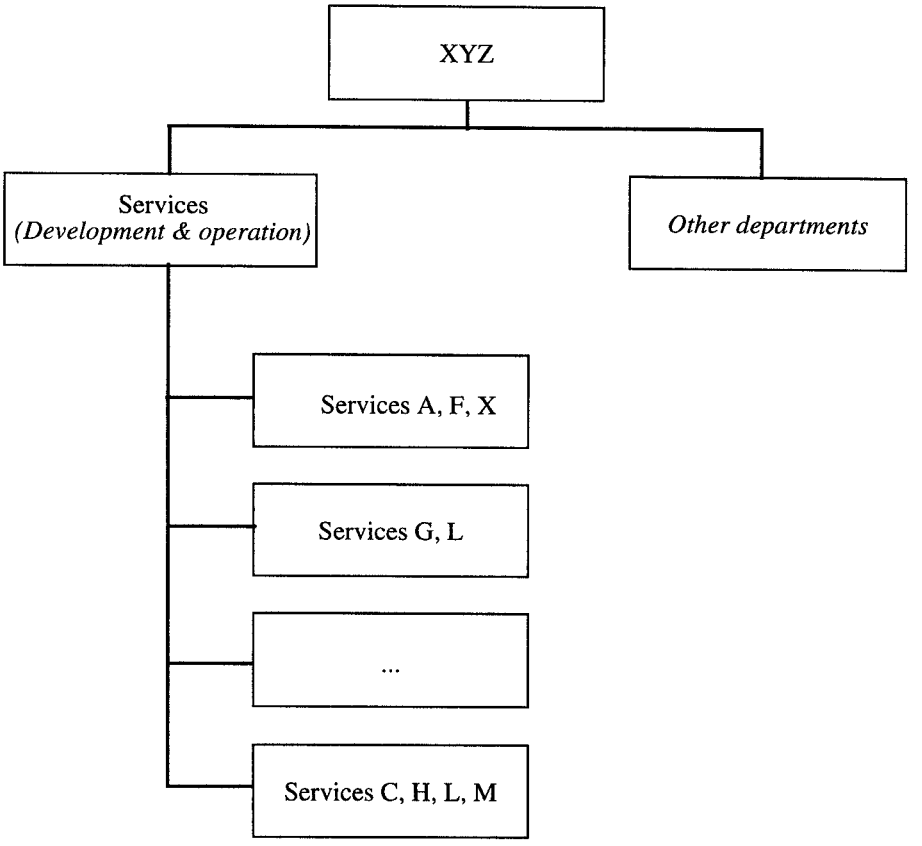

Figure 2: A new model for XYZ.

[7], [8]. In general, we (as a community) "often overlook organizational and social issues" [11].

We also tried to figure out how to apply the QIP approach. We decided that it was absolutely unfeasible to use the QIP in the XYZ situation, since there were not enough resources, competence, and suitable organization structure to adopt such an approach. Our understanding is that the QIP is basically a more detailed specification of CMM level 5, where continuous improvement is institutionalized and based on quantitative evaluation of process performances. To reach that level, the CMM correctly assumes that important practices have already been put in place. They include configuration management (how to measure products characteristics if you do not know your product?) and process definition (how to measure the process if you do not know it, or if you have not defined it, or if you do not follow it?). XYZ was not at that stage.

In conclusion, our experience as users of CMM and QIP is that they do not help in the critical part of an assessment. We need different kinds of aids:

1. It is necessary to identify means to support and facilitate *problems elicitation (analysis of the organization)*. This is one of the most complex part of the job. If we do not understand the real problems that affect the organization performance, any technical innovation is useless or even counterproductive. A somewhat surprising observation is that, considering the state of practice in the general business field, we can discover a series of (very simple) methods and techniques that help an assessment team in identifying the problems of a company. They are based, typically, on some of the so-called "seven management tools" [5]. In particular, affinity diagrams, cause and effect diagrams (also called Ishikawa diagrams), and relationship diagrams[3], although very simple and informal, are the basis of an effective methodology for problem elicitation that is widely used in many sectors since the late 80s.

3. They have nothing to do with Chen's Entity Relationship Diagrams.

2. The experiences gained in the past years in business process reengineering and total quality management have demonstrated that to solve the critical problems of an organization it is necessary to "start over", and *reconsider and redesign the whole organization according to its mission* [6], [1]. The indications offered by the CMM are useful, but do not help in defining a comprehensive and effective improvement strategy. In the XYZ case, the real issue is to move from "functional departments to process teams" [6], each of them dealing "in toto" with a specific service to be offered to XYZ's customers. This is what Hammer calls "reinventing the business" [6]. As a general conclusion, we realized that the real goal for XYZ is to "seek every opportunity to build horizontal linkage across the organization to serve customers" [3]. This kind of issues and indications is not considered at all by CMM.
3. We do not have yet a deep understanding of how to manage *the transition to a new organizational model* or *the introduction of innovations* in an existing process. There has been some contribution in this area (see for instance [12]), but there is still a long way to go before consolidated and effective results are achieved.

6 Conclusions

This paper has briefly presented an experience in assessing and improving a medium size company operating in Italy. The assessment was conducted according to the SEI CMM and by taking into account also the indications offered by the QIP.

SEI claims that CMM can be used to address two different goals [10]:

1. to evaluate contractors/bidders (*Software Capability Evaluation*);
2. to guide and support a process improvement initiative (*Software Process Assessment*).

We did not consider in our work the first goal. As for the second one, we find that the indications offered by CMM are quite obvious and do not take into account important aspects related to the overall company structure and organization. We realized that the real critical issues in XYZ were not covered by CMM. In particular, we consider as top priority the reorganization of the company according to the processes followed to create and deliver XYZ's services. It is the mandatory prerequisite to enable other improvement initiatives, including the ones suggested by the CMM. Unfortunately, CMM did not offer any aids in this kind of analysis/diagnosis.

We therefore argue that we need to rethink this assessment method and learn from what other domains have been doing. In particular, business reengineering and total quality management put a lot of emphasis on the need of rethinking an organization according to its goals, before introducing any specific change and/or technology innovation. Therefore, we believe that the CMM, and any other assessment method, should explicitly take into account important dimensions such as the organization and how it relates to the company mission, its marketplace, and the human resources that are at hand.

Finally, we argue that QIP can be applied only to a very mature organization, while it is less useful or even not applicable at all to immature organization which have not found yet their way towards higher maturity levels. This is consistent with the experiences in business process reengineering, where continuous improvement can be accomplished once the basic, structural problems of the company have been solved. Quoting [1], "... once the 'radical' changes are made, they must install continuous process improvement practices in the business operation to prevent future deterioration and ensure preventive maintenance." We believe that many existing software processes need "radical changes" and major reengineering effort which (unfortunately) cannot be based on quantitative assessment and continuous improvement.

In conclusion, we should define assessment methodology with a broader vision of the problems that afflict software processes. It is not sufficient to consider just engineering issues and it is often unrealistic to foresee a predefined improvement strategy. Moreover, we have to take into account that most software companies are still at a very low maturity level. Experiences in other industrial and business domains could help us in identifying more general and effective methods and techniques.

References

[1] D.C. Andrews, S.K. Stalick. *Business Reengineering: the survival guide.* Yourdon Press Computing Series, 1994.

[2] V. Basili, S. Green. Software Process Evolution at the SEL. In [9].

[3] G. Bounds, L. Yorks, M. Adams, G. Ranney. *Total Quality Management: toward the emerging paradigm.* McGraw-Hill, 1994.

[4] J. Brodman, D. Johnson. What small businesses and small organizations say about the CMM. *Proceedings of the 16th International Conference on Software Engineering* (ICSE 16), IEEE Computer Society, Sorrento (Italy) 1994.

[5] R. Futami. Seven Management Tools for Quality Control. *Proceedings of the International Seminar on TQC*, Juse, Tokyo (1988).

[6] M. Hammer, J. Champy. *Reengineering the corporation: a manifesto for business revolution.* Nicholas Brealy Publishing, London (UK) 1993.

[7] V. Haase, G. Kock, H.J. Kugler, P. Decrinis. Bootstrap: fine-tuning process assessment. In [9].

[8] P. Kuvaia, J. Similä, L. Kranik, A. Bicego, S. Saukkonen, G. Koch. *Software Process Assessment & Improvement: The Bootstrap Approach.* Blackwell Business, 1994.

[9] S. L. Pfleeger, H.D. Rombach, eds. Special issue on measurement-based process improvement. IEEE Software, Vol. 11, No. 4, July 1994.

[10] M. Paulk, B. Curtis, M.B. Chrissis, C.V. Weber. Capability Maturity Model for Software, Ver. 1.1. Technical Report CMU/SEI-93-TR-24, Software Engineering Institute, February 1993.

[11] D. Perry, N. Staudenmayer, L. Votta. People, organizations, and process improvement. In [9].

[12] SEI. Tutorial on Technology Transfer and Organization Change Strategies. 13th International Conference on Software Engineering (ICSE 13), IEEE Computer Society, Austin (USA) 1991.