This paper describes the case study of an application of the Capability Maturity Model (CMM) to a micro team with less than 10 people and very limited resources, clearly placed in the first CMM level of maturity, the Initial level. A major aim of the case, which extended over a period of one year, was to study the extent to which the CMM could be adapted to very small teams for software process improvement (SPI). We have concentrated primarily on those parts of the CMM that involve the achievement of the objectives of key process areas (KPA) in level 2, the Repeatable level. Throughout the year we gathered data from a wide range of management and engineering tasks. The results obtained after one year have shown a clear improvement of the process. The pragmatic application of the CMM has thus shown that SPI can indeed be achieved by a very small team. We also came to the conclusion that the CMM cannot be applied in a straightforward manner by such a small team. We have also concluded that team management is just as important as process management in this kind of environment. Copyright © 2000 John Wiley & Sons Ltd

KEY WORDS: software process improvement (SPI); Capability Maturity Model (CMM); micro team

INTRODUCTION

The CMM, or Capability Maturity Model, has been established in response to the so-called software crisis of the late 1970s, which was a motive for very serious dissatisfaction from the US Department of Defence (Humphrey 1988). This crisis, which has by no means decreased in the software industry throughout the years, results from the lack of maturity of the software development process. Its most visible consequences are the low quality and reliability of most of present day software products.

The major problems naturally tended to occur in highly complex systems, for which very large teams had to be joined together. This explains why the CMM considers a team to be small if it has less than 70 people, big if more than 200 people are involved, and medium sized if its dimension falls between those two limits (Goldenson and Herbsler 1995).

The current software industry is, however, largely made up of very small firms, many comprising less than 10 people for software development. We will refer to these as micro teams. Micro teams also have very serious problems of maturity in their software development processes. In fact, in many cases no real process exists, often leading to very chaotic modes of operation that affect the whole firm. In the words of Davis, ‘software is
burst of creativity and individual genius rather than teamwork and engineering discipline’ (Davis and Leffingwell 1996).

In this paper, we describe the case study of an application of the CMM to a micro team. The main objective of the case, besides the obvious one of improving the competitiveness of the team, was to study to what extent we could apply, use and adapt the CMM to such a small team.

In our software process improvement (SPI) effort, we have chosen the CMM because: (a) we view it as the most widely known method for software process improvement; (b) it is very well documented; and (c) there is a close relationship between the CMM and ISO 9000. Besides the CMM, there are other SPI methods that we have not explored, including SPR (Jones 1994), QIP from SEL (Basili and Green 1994, Basili et al. 1995), and SPICE (Emam et al. 1997).

The CMM (Paulk et al. 1993a, b, Humphrey 1993) recognizes a set of five levels of maturity of the software process: 1, Initial; 2, Repeatable; 3, Defined; 4, Managed; 5, Optimizing.

Each one of those levels expresses a different state of maturity in an organization devoted to software production. In this scale, level 1 corresponds to the lower state of maturity and level 5 corresponds to the higher state of maturity. We say that the process of an organization is at a given level of maturity, besides level 1, when the objectives of that level have been attained. The objectives are grouped in key process areas (KPA). Thus, for instance, level 2 will have been achieved when all the objectives of the following KPA have been met:

- KPA 1 Requirements management
- KPA 2 Software project planning
- KPA 3 Software project tracking and oversight
- KPA 4 Software subcontract management
- KPA 5 Software quality assurance
- KPA 6 Software configuration management

For each KPA, a number of key practices (KP) are defined. The KPs establish what must be successfully completed, but not how it is to be completed.

In the text that follows, we start by describing the case study and the approach we have taken. We then present results that show that the process has improved to a certain degree. Finally, we discuss those results and present some concluding remarks.

BACKGROUND

The case study is about a small business in which CMM was applied during 12 months. The business has four units, or sectors: (a) software sector (SS); (b) commercial sector (CS); (c) hardware sector (HS); and (d) financial sector (FS). The SS was a micro team with less than 10 people, with very limited resources, and clearly in the first level of maturity, the Initial level.

The mission of the SS is to develop and supply high quality business software products and services. Clients use software that the SS supplies and the SS provides technical support for clients. In return, clients pay a monthly fee.

The SS was very chaotic when the case begun. There was no effective leadership, and motivation was very low. Despite the experience acquired during the years, there was no internal or external co-ordination. Technical abilities and practices had poor sophistication. No managerial or engineering procedures or policies were reinforced, and no administrative support existed.

At the technical level, quality was low. No analysis and requisite management was carried out, version control was poor, no concern existed about reusing code, and no attempts were made to improve quality. Application development was not planned or followed through, and the resources for development were almost exclusively applied in maintaining the existing applications. No one knew exactly what other people were working on.

From the client’s point of view, the lack of satisfaction was also clearly visible. Indeed: (a) some of the applications did not carry out all the required tasks; (b) other applications completed the desired tasks, but often in an inconsistent manner; and (c) support services were neither effective nor efficient.

The other sectors of the business also had their own problems. In particular, the negotiation and sale of software service contracts by the CS were based on insufficient knowledge about the corresponding products and services. This often led to last minute changes and adaptations that had not been planned or managed. The HS did not follow any adequate quality assurance
procedures regarding the equipment and services they supplied, which often resulted in complaints from clients that put the blame on the SS. Not surprisingly, the financial situation of the firm was fragile.

All those factors contributed to increasing the instability and confusion within the SS. In other words, software was developed but no process existed in order to do it, and as no process was identifiable, no repeatability could exist. This clearly put the operation of the SS in CMM level 1, the Initial level.

The changes described in this case started when a new leadership was appointed for the SS. The main objective was to find out how the CMM could be applied, used and adapted to such a team so that the process could clearly be improved. Particular objectives were: (a) to create a repeatable software process; (b) to manage the team so as to keep high levels of motivation and performance; (c) to prepare the SS for technological evolution; and (d) to rationalize permanently about all items, so that the process could evolve continuously.

To achieve these objectives, the action plan for the first year included: (a) the detailed study of the key process areas of level 2, and the clear understanding of their objectives, followed by adaptation and application to the SS; (b) the maintenance and correction of the existing applications and support to the clients that owned them; (c) the creation of new applications, developed using more advanced technology, so as to induce the analysis and management of requisites, code reuse, version control, and other correct practices; (d) the search for new markets; and (e) the permanent review of the action plan.

To put this plan into practice, some more resources were needed: (a) human resources were reinforced; (b) computers and development software were upgraded; and (c) access to technical information and to the Internet were made available.

THE APPROACH

The application of the CMM to the SS took place in three phases. The first phase was dedicated to an evaluation of the maturity of the software process. As described above, it was soon detected that the software process was clearly in level 1 of the maturity scale. Furthermore, the only KPA of level 2 that required no intervention was KPA 4, software subcontract management, simply because the SS did not resort to software subcontracting. All the other five KPAs called for serious intervention, and it soon became clear that it was not practical to intervene simultaneously in all of them.

The second phase of application of the CMM to the SS was concentrated on KPAs 1, 3 and 6. In this way, not only the requisites and configuration of the software products could be controlled, but data could also be collected to describe the behaviour of the SS through the response to questions such as ‘How long did it take to develop program X and how much did it cost?’ or ‘What is the productivity of programmer Y?’.

Given the data that clarified the time and cost associated to each set of applications, it became possible to start establishing development plans. Thus, the third phase could be initiated by concentrating now on KPAs 2 and 5, in search of higher levels of predictability of results for the software process and in the systematic search of higher levels of quality.

From the detailed application of the KPAs to the SS, which is described in the documentation produced for the case, some methodological aspects should be mentioned here:

- The roles established in the CMM have been simplified, given the dimension of the team.
- The norms and procedures have not generally been written down, but rather transmitted and absorbed by the group at the appropriate times.
- All the elements of the SS actively participated in the resolution of any problems and in determining the direction of the SS, both at the technical and management levels.
- The KPAs and their KPs have been evaluated keeping in mind the need to apply the CMM with pragmatism (a) because the team was small, and (b) because the resources were very scarce.

Given their particular relevance for success, some specific aspects of the application of the CMM to our SS should also be enhanced:

- Software requisites and their changes were always written down and always resulted from a process of analysis.
- Time allocations and costs associated to the various activities were measured and registered.
Each element of the team filled out a daily time sheet where the tasks carried out were specified in agreement with a pre-defined table, and the starting and finishing times were registered. Those tasks could be merely technical, such as ‘programming’. In this case, they were associated with an application. Those tasks could also be management-oriented tasks, such as ‘business meeting’. In this last case, they were associated with an application and a client. After some ‘tuning’ time, an application was developed to register those elements automatically.

- The systematic use of metrics led, after some time, to the creation of expectations (though not yet estimates) for the development of the projects. This initial process of ‘expectation management’ will soon become a more rigorous process of: (a) estimation of the size of the software solutions; (b) estimation of the costs involved; and (c) estimation and monitoring of the calendar for each project.

- Monitoring and supervision became a systematic task of the team leader, who made sure that changes and improvements were always in agreement with the people involved and made known to the whole team.

- Development plans for each software project started to be established, though not fully written down. Each project was regularly discussed with the elements of the SS before tasks were distributed and resources assigned to each task. The team leader was responsible for obtaining the required resources in negotiation with the other sectors of the firm.

- The SS accepted the commitment to a permanent search for quality. As the small size of the team made it impossible to set up a quality assurance group, some general practices started to be followed instead. For instance, the systematic revision of a product or sub-product was carried out by people who were not involved in the development of that particular product or sub-product. In this way, the whole SS became responsible for the quality processes. Though this did not grant real independence in the quality management process, it was considered to be a feasible and acceptable compromise.

- The SS recognized the need for a rigorous control of all versions. Therefore, applications and utilities have been developed to build up a digital repository of some of the elements as well as to manage the placement and lifecycle of those elements that were not available in digital form.

**RESULTS**

After the first year of application of the CMM many results could be gathered and organized. The most significant ones are shown.

**Time Results**

It is clearly noticeable that throughout the period of analysis the time dedicated to software production (which corresponds to the development of new applications and the maintenance of existing ones) has increased. Conversely, the time dedicated to other activities, which we described as non-production activities, has decreased. In this category, the activity which carried the most weight was technical support to clients. Figure 1 shows that: (a) the percentage of monthly time dedicated to software production has increased 137% throughout the 12 months; and (b) the percentage of monthly time dedicated to non-production activities has decreased to 43% of its initial value.

These results are reinforced by the comparison between the time dedicated to the maintenance of existing applications and the time spent developing new applications. As can be seen from Figure 2, the percentage of monthly time dedicated to the maintenance of existing applications has decreased to 26% of its initial value, while the time devoted to the production of new applications has increased by 86%.

**Client Results**

From Figure 3 we can see that the number of monthly interventions in response to client calls has been reduced to 25% of its initial value. During the period of our analysis the number of clients increased slightly, but not significantly. Some clients have been lost during the first months, but new ones were gained in the meantime.

**Financial Results**

To express the financial results that follow, an index system is used instead of real values. The
Figure 1. Time spent in the activities of the software sector (SS): time dedicated to non-production activities versus time dedicated to production activities

Figure 2. Time spent in the activities of the software sector (SS): time developing new applications versus time maintaining existing applications

total cost for the first month is expressed as index 100 and all the others are compared with it.

From Figure 4 we can see that: (a) monthly costs decreased to 67% of their initial value; (b) monthly benefits increased 17% of their initial value; and (c) the difference between benefits and costs has decreased to 28% of its initial value, nearing zero and suggesting a trend towards positive values.

Figure 5 illustrates monthly costs discriminating production activities and non-production activities. It shows that: (a) the monthly cost of production activities has increased 59% throughout the 12 months; (b) the monthly cost of non-production activities has decreased to 29% of its initial value; and (c) these results are consistent with those presented in Figure 1.

DISCUSSION AND CONCLUSIONS

The results we have presented are globally consistent between them. The SS holds a high degree of autonomy, and during the 12 months of the case study it has not been subject to any new variables. Thus, the results suggest that the application of the CMM to the SS has contributed significantly to process improvement.
Figure 3. Number of technical support interventions for clients

Figure 4. Costs and benefits of the software sector (SS)

Figure 5. Production activities costs versus non-production activities costs
Some doubts may arise regarding the interpretation of the financial information in Figure 4, since the trend in the reduction of costs is much more evident than the increase in benefits. We believe that this resulted from the rigour and discipline that the use of the CMM has imposed within the SS. On the other hand, the increase in benefits depended strongly on the development of new applications, which took some time, and of the subsequent profitability. We believe that, as time goes by, the trend line for costs will tend to stabilize, or even increase, and the benefit line will tend to rise more sharply.

The application of the CMM to the SS was by no means easy. Given the small dimension of the team and its lack of resources, many simplifications of the method had to be introduced, and only the KPs that were really important for the process have been retained. We could say that the CMM has been applied in a very pragmatic way.

The attempt to move from level 1 to level 2 of the maturity scale has not been completed. At present, the maturity of the SS lies above level 1 but below level 2.

In general, we may conclude that:

- the pragmatic application of the CMM to the SS has led to significant improvements that put its software process above level 1;
- the software process is below level 2, but showing that this level can be achieved;
- the application of the CMM to micro teams is possible and contributes to the improvement of their software process, however, simplifications of the method are required, namely in the structure of the functions and in the formalism of procedures and norms – the costs of trying to fully apply the CMM to micro teams such as the one described would be far too high;
- the results obtained are globally positive and consistent between them, which suggests that they are a consequence of the pragmatic application of the CMM;
- the pragmatic application of the CMM leads to higher levels of quality of the resulting software;
- the CMM is exclusively concerned with the software development process.

Besides the software process, the application of the CMM to a micro team has shown that another critical factor must be taken into account, that is the human resources factor of the team and its management. This has shown to be particularly sensitive, because: (a) the level of maturity of the SS was initially very low; and (b) the small dimension of the SS did not facilitate a global cultural change, unless it was attempted directly through each member.

In summary, the application of the CMM to a micro team must be carried out judiciously by permanently adapting to the environment of the team, using just the KPs that are really important to the process, and keeping in mind that team management is just as important as process management.

The importance of the factors regarding human resources has been recognized by other authors (Bach 1995), and even by the SEI in its publication of CMM V.1.1: ‘The CMM may also become multidimensional to address technology and human resource issues’ (Paulk et al. 1993a). This resulted, in 1995, in the publication of the P-CMM – People Capability Maturity Model (Curtis et al. 1995). Also in 1995, the technological dimension was addressed in the SE-CMM – Systems Engineering Capability Maturity Model (Bate et al. 1995). These are factors that must be taken into further account in future refinements of the micro team approach, to make the application of the CMM more consistent with the dimensions of process, technology and human resources.

ACKNOWLEDGEMENTS

This study was partially supported by the Portuguese Foundation for Science and Technology (FCT) under research contract 326/94. The authors are grateful to ISCAA and CISUC for all the facilities granted.

REFERENCES


Bate R, Kuhn D, Wells C, Armitage J, Clark G, Cusick


