

European Software Process Improvement **SPI and Assessments**

[Category Index](#)

Rated Newspaper Supported by EU Project



Send rating for this paper

CMM in a Micro Team: A Case Study

Joao Batista
ISCAA/CISUC, Portugal
A. Dias de Figueiredo
CISUC, Portugal

Abstract

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 short resources, clearly placed in the first CMM level of maturity, the Initial Level. One 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. We have concentrated primarily on those parts of the CMM that regard the achievement of the objectives of Key Process Areas (KPA) in Level 2, the Repeatable Level. We have identified some KPAs where improvement could be started, and we have included others as we moved along. 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 number of calls from clients for technical assistance had decreased to 25% of the initial value; the budget requirements had decreased to 28%; and the time devoted to software production had increased by 137%. The pragmatic application of the CMM has thus shown that improvement can indeed be achieved for very small teams. We also came to the conclusion that the model cannot be applied in a straightforward way to such small teams: it must be used judiciously, by constantly adapting it to the environment of the team and by selecting just the key practices that are really relevant to the process. We have also concluded that team management is just as important as process management in this kind of environment.

Introduction

The CMM, or Capability Maturity Model, has been established in response to the so called software crisis of the late 70s, which was a motive of very serious dissatisfaction from the US Department of Defense [1]. This crisis, which has by no means decreased in the software industry as the years passed, 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 bigger problems naturally tended to occur in highly complex systems, for which very large teams had to be assembled. This explains why the CMM considers a team to be small if it is composed of less than 70 people, big if more than 200 people are involved, and medium sized if its dimension falls between those two values [2].

The current software industry is, however, largely made up of very small firms, many even resorting to less than 10 people

for software development. We will refer to those 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 does even exist, 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" [3].

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.

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. Other methods, besides the CMM, that we have not explored include SPR [4], QIP from SEL [5] [6], and SPICE [7].

The CMM [8] [9] [10] recognises a set of five levels of maturity of the software: 1 - Initial; 2 - Repeatable; 3 - Defined; 4 - Managed; and 5 - Optimising.

Each one of those levels expresses a different state of maturity in an organisation 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 organisation 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 attained 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 realised, but not how it is realised.

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 in some measure. Finally, we discuss those results and present some concluding remarks.

The Background

The case study is about a small business in which we have applied CMM during 12 months. The business has four units, or sectors: a) software sector (SS); b) commercial sector (CS); c) hardware sector (HS); and, d) finance sector (FS). The SS was a micro team with less than 10 people, with very short 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 make technical support to 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. Although significant experience had accumulated over the years, there was no internal or external co-ordination. Technical abilities and practices had poor sophistication. No managerial or engineering procedures or policies had been reinforced, and no administrative support existed.

At the technical level, quality was low. No analysis and requisite management were 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 and followed through, and the resources for development were almost exclusively applied in maintaining the existing applications. No one knew exactly what the other people working on.

From the client side, the lack of satisfaction was also very clearly visible. Indeed: a) some of the applications did not carry out all the expected tasks; b) other applications completed the desired tasks, but often in inconsistent manners; and c) support services were neither effective or 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 nor managed. The HS did not follow any adequate quality assurance procedures regarding the machines 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 very fragile.

All those factors contributed to increase the instability and confusion within the SS. In other words, software was developed but no process existed 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 reason permanently about all items, so that the process could continuously evolve.

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 have been reinforced; b) computers and development software have been upgraded; and c) access to technical information and to the Internet have been granted.

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 found out that the software process was clearly in level 1 of the maturity scale. It was also found out that 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 were calling 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's the productivity of programmer Y?".

In possession of 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 questing a more systematic pursuit for 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 interiorised 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 stressed:

software requisites and their changes were always written down and always resulted from a process of analysis;

the times and costs associated to the various activities have been measured and registered. Each element of the team filled up 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 which case they were associated to an application, or they could be management tasks, such as "technical support", in which case they were associated to an application and a client. After some "tuning" time, an application has been developed to register those elements automatically;

the systematic use of metrics led, after some time, to 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 agreed 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, and 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 that 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 found to be the feasible and acceptable compromise;

the SS recognised the need for a rigorous control of all versions, and applications and utilities have been developed to build up a digital repository of some of the elements and to manage the placement and life cycle of those elements that were not available in digital form.

The Results

After the first year of application of the CMM many results could be gathered and organised. 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 with most

weight was technical support to clients. Fig. JBADF.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 decrease to 43% of its initial value.

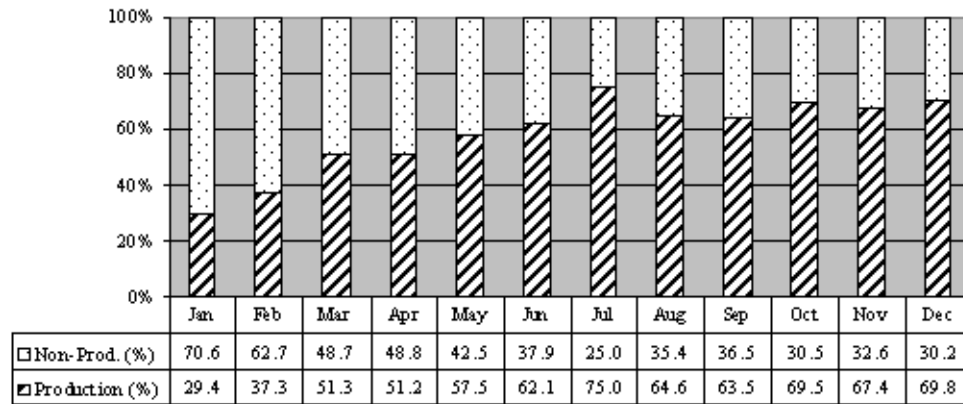


Fig. JBADF.1 - Time spent in the activities of the software sector (SS): time dedicated to non-production activities versus time dedicated to production activities.

Those 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 Fig. JBADF.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

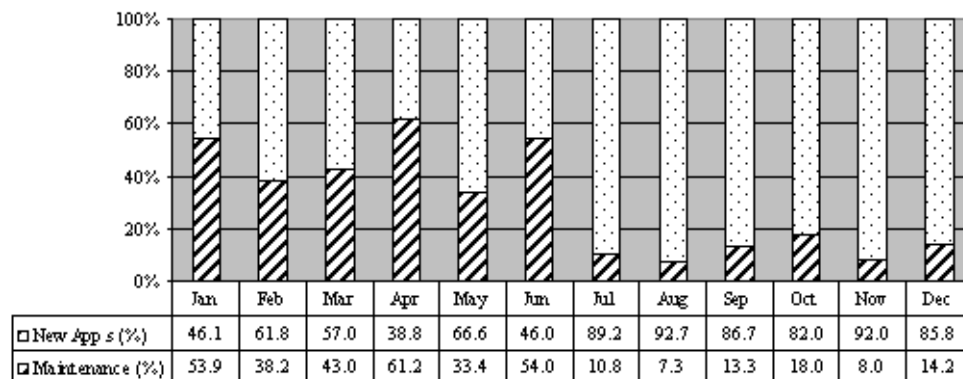


Fig. JBADF.2 - Time spent in the activities of the software sector (SS): time developing new applications versus time maintaining existing applications.

From Fig. JBADF.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 brought in the meantime.

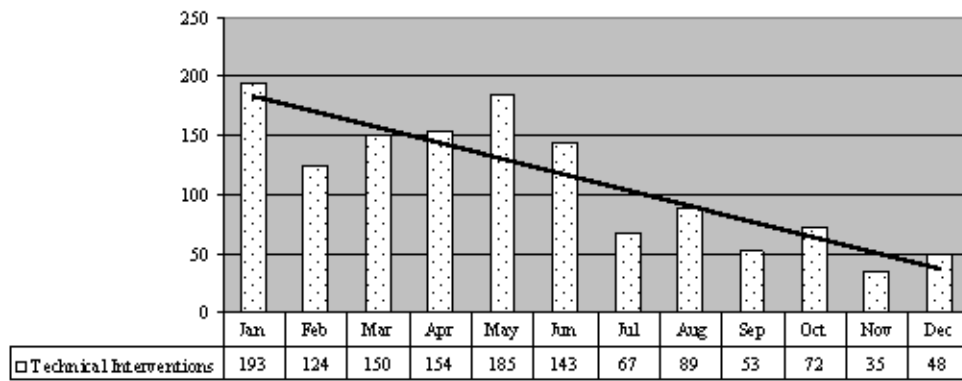


Fig. JBADF.3 - Number of technical support interventions for clients.

Financial Results

To express the financial results that follow an index system is used instead of real values. The total cost for the first month is expressed as index 100 and all the others are compared with it.

From Fig. JBADF.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.

Fig. JBADF.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) those results are consistent with those presented in Fig. JBADF.1.

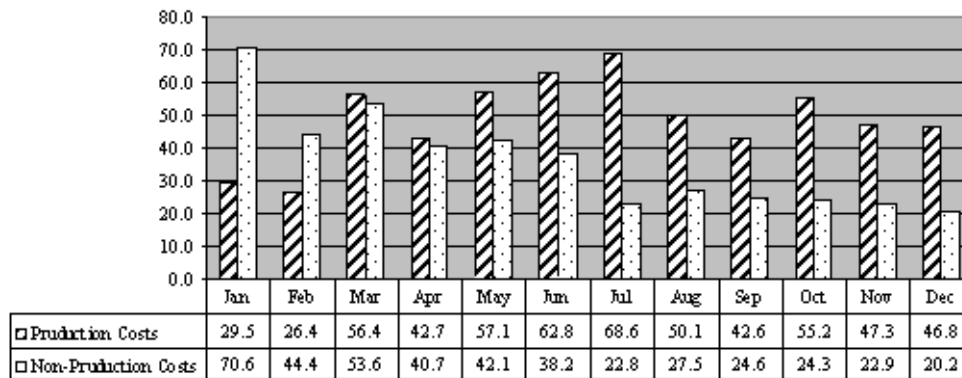


Fig. JBADF.4 - Costs and benefits of the software sector (SS).

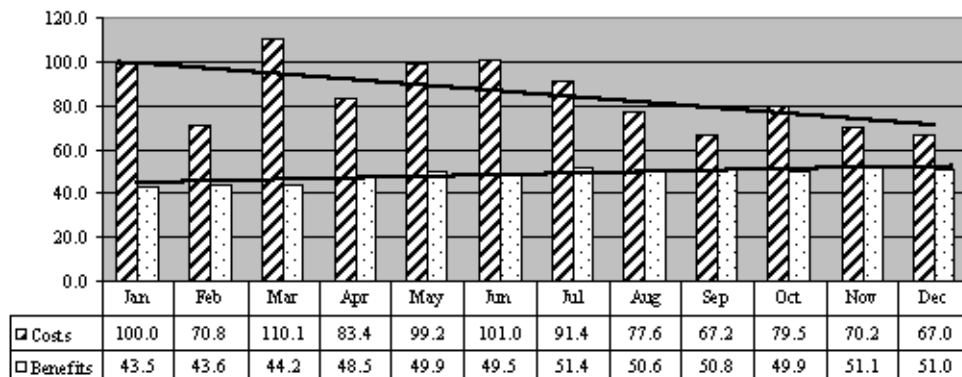


Fig. JBADF.5 - Production activities costs versus non-production activities costs.

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.

Some doubts may arise regarding the interpretation of the financial information in Fig. JBADF.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 has depended strongly from 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 stabilise, or even grow up, and the benefits line will tend to grow 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 lays 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.

The application of the CMM to a micro team has shown that another critical factor must be taken into careful account besides the software process: 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 relevance of the factors regarding human resources has been recognised by other authors [11], and even by the SEI in its publication of CMM V.1.1: "The CMM may also become multi-dimensional to address technology and human resource issues" [8]. This resulted, in 1995, in the publication of the P-CMM - People Capability Maturity Model [12]. Also in 1995, the technological dimension has been addressed in the SE-CMM - Systems Engineering Capability Maturity Model [13]. These are factors that must be taken into further account in future refinements of the micro team approach, so as to make the application of the CMM more consistent with the dimensions of process, technology and human resources.

Acknowledgements

This work has been 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

- [1] Humphrey, W., Characterizing the Software Process: A Maturity Framework. *IEEE Software* 5, 2 (March, 1988), 73-79
- [2] Goldenson, D., and Herbsler, J., *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors that Influence Success*. CMU/SEI-95-TR-009, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1995
- [3] Davis, A., and Leffingwell, D., *Using Requirements Management to Delivery of Higher Quality Applications*. Rational Software Corporation, U.S.A., 1996
- [4] Jones, C., *Assessment and Control of Software Risks*. Prentice-Hall, Englewood Cliffs, New Jersey, U.S.A., 1994
- [5] Basili, V., and Green, S., Software Process Evolution at the SEL. *IEEE Software* 11, 4 (July, 1994), 58
- [6] Basili, V., Zelkowitz, M., McGarry, F., Page, J., Waligora, S., and Pajerski, R., SEL's Software Process-Improvement Program. *IEEE Software* 12, 6 (November, 1995), 83
- [7] Emam, K., Drouin, J.-N., and Melo, W., (editors) *Spice: The Theory and Practice of Software Process Improvement and Capability Determination*. IEEE Computer Society Press, Los Alamitos, California, U.S.A., 1997
- [8] Paulk, M., Curtis, B., Chrissis, M., and Weber, C., *Capability Maturity Model for Software, Version 1.1*. CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1993
- [9] Paulk, M., Weber, C., Garcia, S., Chrissis, M., and Bush, M., *Key Practices of the Capability Maturity Model for Software, Version 1.1*. CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1993
- [10] Humphrey, W., *Introduction to Software Process Improvement*. CMU/SEI-92-TR-7, revised, 1993, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1993
- [11] Bach, J., Enough About Process: What We Need Are Heroes. *IEEE Software* 12, 2 (March, 1995), 96
- [12] Curtis, B., Hefley, W., and Miller, S., *People Capability Maturity Model*. CMU/SEI-95-MM-02, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1995
- [13] Bate, R., Kuhn, D., Wells, C., Armitage, J., Clark, G., Cusick, K., Garcia, S., Hanna, M., Jones, R., Malpass, P., Minnich, I., Pierson, H., Powell, T., and Reichner, A., *A Systems Engineering Capability Maturity Model, Version 1.1*. CMU/SEI-95-MM-03, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, U.S.A., 1995

Joao Lopes Batista is an Adjunct Professor of Informatics at the Higher Institute of Accounting and Administration of Aveiro (ISCAA) since 1987. He obtained his Master's Degree in Informatics from the University of Coimbra in 1993, and he is currently working on his Ph.D. His main research interests concentrate on Strategy and Management for Information Systems, Software Engineering and Object Technology. His past professional activities included lecturing at the University of Aveiro as an invited professor, and he has been a consultant for external companies in the fields of Software Engineering and Management. He is a researcher of the Centre for Informatics and Systems of the University of Coimbra (CISUC) since 1994. He is a member of the ACM, of the IEEE, and of the Portuguese Professional Institution of Engineers. He is the author of several papers, communications and research reports.

António Dias de Figueiredo is a Full Professor of Informatics Engineering at the Faculty of Science and Technology of the University of Coimbra, Portugal, since 1984. He obtained his Ph.D. in Computer Science from the University of

Manchester, U.K., in 1976. His main research interests concentrate on Strategy and Management for Information Systems, Software Engineering, and Information and Communications Technologies in Education. He is the doyen of the Department of Informatics Engineering of the University of Coimbra, which he founded in 1994 and chaired until March 1997. He is the doyen and founder of the Centre for Informatics and Systems of the University of Coimbra (CISUC), the institution for R&D in Informatics of the University of Coimbra. He represents Portugal in the Intergovernmental Informatics Programme of UNESCO, where he acted for two years as Vice-president elected for the Western Europe Region. He has participated in various European projects, both as a partner and as a science advisor, and acted in various occasions as a consultant to the European Commission in matters regarding the definition of strategies for information and communications technologies in education. At the invitation of the NATO Science Committee, Brussels, he integrated, for a period of four years, the NATO Special Programme Panel on Advanced Educational Technology. At the invitation of the European Commission, he contributed to the preparation of the White Book on Education and Training for the XXI Century. He is a member of the Panel for Research and Development in Consortium of the National Innovation Agency, and integrated various international panels for the approval and evaluation of research projects. He is a member of the panel of the IBM Science Prize since its creation in 1989. He is member of the Portuguese Engineering Academy. He is a member of the ACM, of the Portuguese Engineering Academy, and of the Portuguese Professional Institution of Engineers. He is an elected member of the Council for Qualification and Admission of the Portuguese Professional Institution of Engineers. In 1997 he was awarded an *Honoris Causa* by Universidade Aberta, the Portuguese Open University. He is the author of over 120 papers and presented over 140 communications. He has integrated over three dozens organising and science committees of conferences held both in Portugal and abroad.

The **Instituto Superior de Contabilidade e Administração de Aveiro**, ISCAA, (Higher Institute of Accounting and Administration of Aveiro) is a well-established school of Accounting and Administration of the Portuguese network of public polytechnic institutes. It has been created in 1966, and is currently attended by a population of 1400 students taking their graduations in "Accounting and Auditing" and in "Business Administration". In association with the Portuguese Open University it is running a Master's degree in "Accounting and Business Finances". It has collaboration protocols with the universities of Sceaux and Brighton within the Erasmus Program. It has a staff of about 60 professors and lecturers deeply involved in research projects within their areas.

The **Center for Informatics and Systems of the University of Coimbra** (CISUC) is a large Portuguese research institute in the fields of Informatics and Communications, which has been created in 1991 with the aims of carrying out R&D at a pre-competitive level, training highly qualified professionals, co-operating in national and international projects and programs, and promoting the dissemination of results, namely through contracts with national and international companies. It is currently composed of eight research groups in the areas of Software Engineering and Information Systems, Artificial Intelligence, Communications and Telematic Services, Simulation and Technologies in Education and Training, Control Theory, Dependable Systems, Combinatorial Optimization, and Theoretical Fundamentals of Computer Science. Most of its research activities are developed within national and international projects and research networks, with the support of funding programs, such as PRAXIS XXI and ESPRIT, thus contributing to the R&D effort of the European Community. Its scientific results are regularly published in well-established journals and presented in prestigious conferences. The quality of its work is expressed in national and international awards that the members of the CISUC have received since its foundation, their participation in the scientific and editorial boards of conferences and journals, and their membership to government and professional committees responsible for the definition of R&D policies. Interaction with industry and other private and public institutions, national and international, is strengthened through Instituto Pedro Nunes, a non-profit private organization founded in 1991 for assisting economic agents in their efforts towards global competitiveness.

Partners in EuroSPI





SINTEF



Editors

ISCN LTD, I SCN GesmbH, Schieszstattgasse 4/24, 8010 Graz, and Coordination Office, Florence House, 1 Florence Villas, Bray, Ireland, office@iscn.at, office@iscn.com, office@iscn.ie, Editing Done: 19.7.2002