SOFTWARE PROCESS: IMPROVEMENT PRACTICE
Softw. Process Improve. Pract. 2000; **5**:45–54 (2000)

Managing the Complexity of SPI in Small Companies

Practice Section

Romana Vajde Horvat*, Ivan Rozman and József Györkös
University of Maribor, Faculty of Electrical Engineering and Computer Science, Institute of Informatics, Smetanova 17, 2000 Maribor, Slovenia

Most known models for SPI (e.g. SEI CMM, ISO standards and other methods derived from those mentioned) are primarily suited for large or medium organizations, but with some tailoring they provide substantial support also for the SPI in small organizations considering their specific characteristics. In the article a case of such tailoring – the PROCESSUS model – is presented. The baseline of the methodology is the integration of the CMM and the ISO 9001 together with the ISO 9000-3. According to the integrated model and the study of different lifecycles, a set of standard procedures (SP) and standard documents (SD) was defined. Each standard procedure provides guidelines on how to perform related activities, who is involved, which documents are supposed to be used/derived within the procedure etc. The set of SP and SD is the essential help for SPI conduction – for the purpose of small companies the optimal use of suggested documents and the disposition of roles was defined. The SP-SD set is also described in the article. Copyright © 2000 John Wiley & Sons Ltd

KEY WORDS: software process; software process improvement (SPI); quality system; small software companies; software process improvement for small companies (SPISC)

1. INTRODUCTION

Regardless of their size, software organizations which wish to succeed on the market should provide software of high quality together with related services, support, communication with customers etc. In other words, even a very small organization which develops rather small applications within small projects involving few resources, should assure that their products and services are of the highest quality. After a decade of SPI prevailing within large and medium size organizations the awareness of the importance of the software process improvement is being intensively propagated also to small companies. The experience gained with SPI projects in large organizations contributed to mature models related to software process improvement and assessment (SEI CMM, SPICE) as described in Paulk *et al.* (1993a,b, 1995) and Rout (1996). Considering the special characteristics of small companies, these models and the knowledge and experience gained can also be successfully used for SPI projects in small companies. Nevertheless, appropriate tailoring and simplifications of existing models should be made, and this is also the subject of some recent studies (see Johnson *et al.* 1997).

*Correspondence to: Romana Vajde Horvat, University of Maribor, Faculty of Electrical Engineering and Computer Science, Institute of Informatics, Smetanova 17, 2000 Maribor, Slovenia. E-mail: romana.vajde@uni-mb.si

2. CHARACTERISTICS OF SMALL COMPANIES

The definition of a small software organization is not clearly set with the software community. It depends a great deal on the different types of companies concerned and also on the software market itself. For example, for a multinational company, its branch company/division with 100 or even more employees is treated as a small company, while it is possible that on the same market, only few domestic independent software companies with over 100 employees could be found. Nevertheless, the large number of really small companies with 5 to 15 employees also plays an important role in the software development area. These companies are flexible, mostly oriented to the development of specific software for known users and not for the broad market, and they are usually specialized in a specific area of work (e.g. software support for insurance, transport, education etc.) which ensures their business success and existence on the market. The characteristics of these companies, such as specific procedures of work and specific relationships between employees, require appropriate management of SPI projects.

Considering the software process improvement issues and the influence of the human factor on SPI projects, small companies can be divided into three types.

2.1. Type A: small branch company of a large software company

The nature of work within large companies requires that procedures of work, roles of personnel and input/output documents for each procedure are defined in order to provide efficient communication between the staff involved in software development. This has already been achieved in the majority of large companies due to the results of SPI projects. Consequently, the working procedures and the expected results of each procedure (like products, documents) have also been introduced within their branch companies. Initial support dealing with starting budget, equipment, training of personnel and their introduction into running projects simplify the establishment of a branch company. This also affects SPI projects which have to be conducted and directed according to instructions, requirements and policy of the global company. Therefore, the number of employees does not influence SPI as much as it affects the other two types of companies.

2.2. Type B: Small Independent Company

An independent company is defined as a company which exists on the market independently of other companies and works according to its own procedures and policy. Based on our experience with cooperating organizations within our projects, ranges were defined as in Table 1.

Further in the paper special attention will be given to small independent companies, although the presented results will be equally applicable in other types of organizations.

2.3. Type C: IT departments within large enterprises

IT departments within enterprises from different branches of industry (e.g. pharmaceutical companies, banking, manufacturing etc.) can be treated as independent companies within the enterprises, because they have many similarities with the companies described in item B. There are, additionally, important specific factors in such departments; their primary customers are other departments within the enterprise and therefore the connections with other departments should be defined according to the global policy of the enterprise.

3. CHALLENGES FOR SPI IN SMALL COMPANIES

Independent of the type of small company, the primary goal of an SPI project is to improve the company's business success and optimize working procedures. In order to succeed, characteristics of companies should be considered and properly

Table 1. Size of independent companies

Number of employees	Size of company
Up to 15	Small independent company
15 to 50	Medium-sized independent company
Over 50	Large independent company

exploited. The SPI in small companies should consider the following factors:

- *High dependency on individuals*. Because of the small number of employees, individuals are usually appointed to develop software for a certain problem domain. They become experts for this area, and because within the immature process they prepare deficient documentation, the success of projects (and even the company itself) depends on individuals.
- *Small number of employees and the disposition of roles and activities to be performed*. This usually results in the same person having to perform a variety of different activities in the software development process. Activities included differ in the nature of required skills for their performance, therefore it is questionable whether activities are performed optimally.
- *Large impact of the psychological issue and the impact of the human factor*. Companies of this type usually start with the enthusiasm of a few individuals who perform a small project for a specific customer, usually faced with difficulties such as starting budget, insufficient equipment etc. The relationships among the personnel in such a company can be characterized by friendship and deep commitment to the company and its goals. Problems usually occur when such a company grows and its organizational structure is not properly established to support its management. Experience with our partners showed that 10 to 15 employees is the critical size at which a firm organizational structure is required. This was also considered in the definition of the ranges of small independent companies. The success of SPI projects depends on the acceptance of its goals and tasks by every single employee, therefore human, social and cultural factors should be considered within SPI plans (see Sommerville and Rodden 1996).
- *Dependence of a small company on a usually very small number of projects*. It is difficult to run many projects in parallel to provide mutual financial or other support in case of crisis in one of the projects. Companies are often forced to start new projects in order to support the existing projects – which eventually leads to new crises.
- *Importance of communication with customers*. Small companies usually develop software for known customers. In this type of development, communication with customers is intensive, especially in the requirements specification and analysis phase and in the testing, delivery, supporting and maintenance phase. Therefore, the improvement of communication with customers is crucial. This is also the area in which feedback of investment and effort for SPI activities can be evident quickly and at the same time it is an important commercial feature.
- *Difficulties with investing in SPI and with appointing personnel for SPI*. The personnel (or more often the person) with the right skills should be chosen from already overburdened employees to perform the SPI activities. The appointment of personnel to other activities affects the running projects which is, apart from the fixed SPI costs (e.g. costs for external consultants, training for SPI etc.) an additional factor in the issue of SPI finance.

4. THE PROCESSUS[1] MODEL FOR SPI IN A SMALL COMPANY

The SPI model for small companies should be simple, easy to understand and should offer concrete guidelines for the definition and enactment of the defined procedures. Still, it should cover all procedures that have to be performed within the software development process. Further in the paper the PROCESSUS SPI model for small companies (SPISC) is described.

4.1. Background

The PROCESSUS SPISC is derived from requirements and characteristics of the two best known models for software process improvement – the ISO model (ISO 9001 standard (1994) together with ISO 9000-3 (1991) Guidelines and the SEI SW-CMM), (see Paulk *et al*. 1993a,b, 1995). These two models were chosen, because software companies are often required to obtain the ISO certificate, and

[1]The PROCESSUS project (assessment and introduction of a quality system) was initiated in 1994 through the cooperation of the research group Laboratory for Informatics and Slovenian local industry (17 organizations of different types from small to large). The project was financially supported by the Ministry of Science and Technology of Slovenia.

in the process of achieving it, they can use all the support and advantages of the SW-CMM. At the same time, they can also achieve the SW-CMM compliance.

In the detailed comparison, the common issues and differences of both models were defined (see Rozman *et al.* (1997) for a detailed description of the comparison results).

Although the SW-CMM provides substantial support and knowledge about SPI, some key process areas (KPAs) and appertaining activities of the original SW-CMM (especially those considering the coordination between different groups involved in software development) are too complex and abstract for use in a small company.

Some adjustments of the original SW-CMM for use in small companies have already been performed, and one of most known is the LOGOS Tailored CMM (see Johnson *et al.* (1997)). Within the LOGOS model the activities, documents and responsibilities of the original model are redefined and simplified, but the framework of the model is the same as the framework of the original SW-CMM.

Since some of the requirements stated in ISO model are covered in other CMM models (SA-CMM, SE-CMM and IPD-CMM; see http://www.sei.cmu.edu/cmm/cmms/cmms.html), the organization should consider also the content of these models. Nevertheless, the knowledge and usage of all CMMs are much too complex for small companies. The characteristics and requirements of these models should be integrated and presented in a simpler way within one (integrated) model. Within the PROCESSUS SPISC model this goal is achieved.

Additionally, the flexibility of the small company enables some activities, which are grouped in the particular KPA, to be positioned within other KPAs, or the sequence of the KPAs can even be changed. The most important issues that influence the mentioned simplifications of the integrated model are as follows:

1. The organizational structure within a small company is rather simple, therefore it can be already defined at the beginning of the SPI project. There are also only few roles which exist in the company, and their responsibility for required activities is defined within each procedure.
2. The types of documentation for definition and description of procedures and the types of documentation required/derived within them should be defined at earlier phases of the SPI project. The consistency of the documentation defined during the SPI is also improved this way.
3. The instructions for implementing procedures in practice should emphasize the technical issue for each procedure – like instructions for which activities should be included considering the knowledge of area covered within the procedure.
4. Because of the small number of employees and the small scale projects, the introduction of the improved working procedures is quick – providing that all personnel are committed to the goals of the SPI project. The result is that some issues, like quality assurance, can be incorporated already from the beginning in other procedures.

The SPI projects in small companies are usually conducted in cooperation with external consultants, who present basic knowledge about quality systems and the software process to the personnel assigned to quality management. This method of working additionally accelerates the introduction of the improved process.

4.2. Framework

The framework of the PROCESSUS SPISC is based on the *process modeling* paradigm. Each procedure is treated as a process, which is defined, established, implemented and maintained. The sequence of process implementation is defined within the model.

Figure 1 presents the framework of the model. In the first circle five steps are defined, which have to be performed during the introduction of each procedure:

- *Analysis*. The aim is to define the current state of activities already performed in the company. It can be performed as an external or internal assessment of specific area or simply the critical issues and problems can be defined.
- *Definition*. According to the results of the analysis and knowledge of the described area, the procedures should be defined in detail. During the definition of the procedure, the

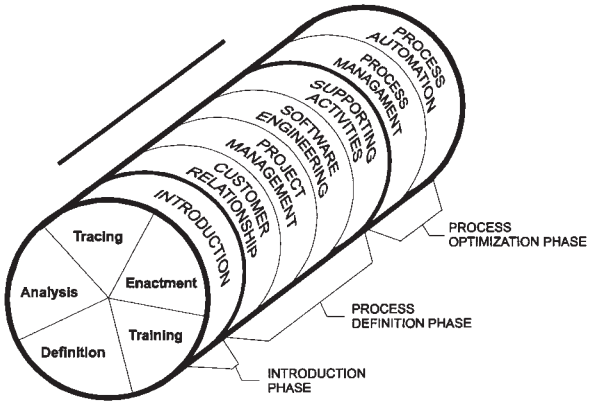

Figure 1. The PROCESSUS model for SPI in a small company

resulting procedure should be discussed with the personnel who perform the activities of the selected procedure.

- *Training*. The defined procedures of work and especially the changes incorporated in the improved procedure should be presented to personnel. Already during the training phase, the real cases from existing projects in the organization should be used to present the applicability of defined procedures.
- *Enactment*. The defined procedures should be first used in some pilot projects, in order to establish the required routine for performing the tasks and activities involved.
- *Tracing*. The applicability, efficiency and acceptance of defined procedures should be traced and analyzed. In a case where major changes are required, the whole circle should be repeated, otherwise the analysis for the next procedure should start.

4.2.1. Introduction phase

The purpose of the introduction phase is to prepare the personnel for the SPI activities and also to define the framework of the organizational structure, according to which all procedures should be defined. The following groups of activities are included in the introduction phase:

- assignment and training of the quality manager (and quality assurance group, if applicable);
- definition of SPI plan;
- definition of organizational structure (usually small companies adopt the simplified matrix project organization);
- definition of process documentation structure;
- introduction of SPI concepts to personnel;
- definition of a few (two or three) simple metrics for evaluation of improvement during the SPI project.

4.2.2. Process definition phase

As it can be seen in Figure 1, the process definition phase is divided into four sub-phases. There are basically two reasons for such a division: first, with the sub-phases, a straight sequence for the introduction of procedures is provided, and secondly, the motivation for progress is increased, because of small steps in process improvement. The sub-phases and the procedures within each sub-phase are defined according to the CMM framework and issues, described at the beginning of this chapter.

1. *Customer relationship management*. Procedures with great influence on the end-user's satisfaction, and consequently on the effectiveness of the organizations, are placed in the first sub-phase of the process definition phase. The following are included:
   - *Contract management*, which covers the assessment of required work, time, effort and financing for performing the required project, the contract review and coordinating activities with the customer.
   - *Requirements management*, which includes the definition and description of requirements and the requirements change management.
   - *Product delivery* deals with the quality of delivery which includes distribution of the product and its placement and testing in the target environment, completeness of all copies and correctness of installation.
   - *Maintenance*, which relates only to the communication with customers (maintenance requests and records of implemented maintenance). Other maintenance activities related to configuration management in maintenance should be introduced after the introduction of configuration management.
2. *Project management*. This includes the activities for preparing the project plan with special emphasis on quality management activities, reviews of input and output of phases within the project, reviews at particular milestones, etc.
3. *Software engineering*. In this phase, the definition and introduction of all procedures for software

engineering (like analysis, design, implementation and testing) are included. Since these procedures are tightly connected to methodologies used, their definition should consider the description of the methodology and reference to detailed manuals. An additional procedure is added at this phase, although it is not a typical software engineering procedure – configuration management (CM). For the CM in a small company, not all requirements stated in the CM theory are relevant. For example, the approvals of the change are often not relevant, because the developer is often also the configuration manager and project leader at the same time. Therefore, no one else in the company can make a more relevant decision about the appropriateness of the change than he/she can. The change management and records of changes for each configuration unit should be emphasized instead.

4. *Supporting activities*. In this sub-phase, those procedures, which are not directly related to the development of a software product, but are needed to assure that the quality of directly related activities is satisfactory, are defined. Procedures included are: *Training; Document control*, and *Included product management*. According to the needs of the company at this stage, some other procedures, for example, Marketing, can also be defined.

4.2.3. Process optimization phase

The main goal of this phase is to provide continuous improvement of the software process. This can be achieved during the reviews of existing procedures and the definition of their improvement. A further step is the automation of those procedures, or parts of them, for which the automation is efficient and sensible. Because of the two reasons described, two sub-phases were defined within the process optimization phase:

1. *Process management*, which includes the definition of procedures: *Metrics, Process management* (Internal reviews and Corrective actions.) These procedures are aimed at defining the efficiency of stated procedures and to introduce the improvement of procedures.
2. Process automation. When procedures are defined and implemented in daily work, the support of their performance is an additional issue, where the efficiency of the process can be improved. The improved procedures introduce some new activities (like the preparation of documentation, detailed testing procedures etc.), which are required within a quality software process, but at the same time they impose an additional burden on personnel. Companies try to solve this problem by developing internal applications for supporting some critical procedures or by using the available groupware or other similar tools. A promising solution for the future is the process-centered software engineering environments (PSEEs), which would enable the modeling of processes and integrated support of all activities performed within the process. Finkelstein *et al.* (1994) and Fuggetta and Wolf (1996) provide the basic knowledge of the process modeling and automation issues.

5. THE PROCESS DOCUMENTATION

The definition of procedures introduced in the company also requires a set of needed documents. Within the PROCESSUS SPISC, the structure of documents, which is suggested in the ISO model, was used. The document types are as follows:

- *Quality manual* is a general document of the quality system (and process) in the company. It includes reference to other documents of process.
- *Standard procedures (SP)* are the documents in which a detailed description of activities performed within the procedure is set out.
- *Standard documents (SD)* are forms, templates and manuals which are used to perform and properly document the inputs/results of certain procedures.

Considering the presented model and the experience gained in cooperation with our partners in SPI projects, the minimal set of process documentation was defined. In Table 2 the set of standard procedures together with standard documents belonging to them is presented. Within the PROCESSUS SPISC model, the instructions for each standard procedure were defined according to the CMM KPA definition example.

Since in a small company the distribution of roles is of great importance, the roles required are

Table 2. The SP-SD set

Standard procedure		Standard document	Roles		
			Implement-ator	Assistant/ Adviser	Quality controller
D1	Contract management	F: Contract review checklist T: Contract	M	PM, QM	QM
D1	Requirements management	F: Requirements change request T: Requirements specification	PM	M, D	QM
D1	Product delivery	F: Acceptance checklist F: Acceptance report	D	PM	QM, M
D1	Maintenance	F: Maintenance request F: Maintenance report	D	PM	QM
D2	Project management	F: WBS and evaluation checklist F: Milestone review T: Project plan	PM	D, M	QM,DC,M
D3	Configuration management	F: Configuration item description F: Project baseline description F: Item change request	D	PM	QM
D3	Analysis	F: Analysis report M: Manual for selected analysis methodology/tool	D	PM	QM, DC
D3	Design	F: Design report M: Manual for selected design methodology/tool	D	PM	QM, DC
D3	Implementation	F: Implementation report M: Manual for programming in selected environment	D	PM	QM, DC
D3	Testing	F: Test plan F: Test cases F: Test report	DC	D, PM	QM
D3	Project documentation management	T: User documentation and technical documentation	D	PM	QM, DC
D4	Document management	F: Document list F: Document description	QM	M, PM	M, PM
D4	Training	F: Training evidence for employees/customers F: Training material template	D	PM	QM
D4	Included product management	F: Purchase report	D	PM, M	QM
D4	Technology introducing/abandoning	F: Technology introducing/abandoning report	D	M, PM	QM
O1	Metrics	F: Metric definition M: Analysis of measurements	QM	M, PM, D	M, PM
O1	Process management	F: Internal review plan F: Corrective action requirement	QM	M, PM, D	M, PM

SD: F, form; T, template; M, manual
Roles: M, manager; PM, project manager; QM, quality manager; D, developer; DC, developer coordinator
Levels: D, Process Definition Phase; D1, customer relationship management; D2, project management; D3, software engineering; D4, supporting activities
O, Process Optimization Phase: O1, process management

defined and their involvement in each standard procedure is defined.

6. PROCESSUS SPISC IN PRACTICE

The model was used for the SPI projects in 17 companies. Seminars, workshops and consulting were conducted according to the framework of the model and using the manuals developed for specific area of process. Some of cooperating organizations improved their whole organization and whole software process (and got the ISO certificate), meanwhile some other organizations were aiming at improving only some parts of their processes. The improvement of their process is assessed using

the questionnaire, which was also developed within PROCESSUS model. During the last two years we intensively cooperate with companies which use the OO technology and help them establish the OO-SW process.

7. CONCLUSION

SPI models for small companies should consider and utilize the characteristics of small companies, therefore their structure should be as simple as possible. At the largest possible extent they should offer guidelines for definition of the process, procedures and the documents used. These characteristics were also stated as the main goal of PROCESSUS SPISC development. The PROCESSUS SPISC model was used for conducting SPI projects in the cooperating small companies, and for these companies the time needed to finish the Process definition phase was approximately 18 months. Using the model in practice has led to further important conclusions:

1. Major improvements of the process can be achieved by improving the technical issue of the process instead of the organizational issue.
2. The influence of human factors on SPI project success is more important than in large companies, because of the important role of each individual in the small company.
3. The documentation of the process and project documentation improve the quality of the process and of the developed products, but at the same time they represent a significant burden to the employees.

Further challenges for the SPISC are therefore related to the support and the automation of procedures or activities within them.

APPENDIX: PRODUCT DELIVERY MANAGEMENT (DM)

Phase 2, Sub-phase 1 – Customer relationship management

The purpose of the product delivery SP is the definition and establishment of procedures which will improve the delivery of products and, consequently, reduce problems and reclamation after product delivery. The DM SP is a simple procedure, which can return investment and effort within short period. The main issues included in this SP are proving the quality of the delivered product, its professional installation and proper training of its end-user.

To avoid possible legal problems after delivery, the customer should also formally accept the delivered product. Before acceptance, acceptance testing should be performed – either by the customer themselves or in a cooperation with the organization. Acceptance testing should prove the correctness of the delivered product. To implement these activities some additional preparation in the organization should be carried out. Activities for the replication of the product should be performed.

Goals

Goal 1: The quality of the software product delivery is improved

The main goal is to reduce potential problems occurring after delivery. These problems can arise because of badly performed installation, poor instructions to the end-user etc. These problems can be avoided if:

- each delivered product (or its copy) is checked for correctness and completeness;
- each delivered product (or its copy) is professionally installed;
- end-users are trained to use the product properly.

Goal 2: The formal acceptance of the software product is performed

When the customer formally accepts the delivered product the software project is usually finished. The maintenance activities follow, either within a guarantee term or as a new work defined in a separate contract. Even when the project is not finished after the acceptance, the initial status of the delivered product should be formally stated – in order to identify maintenance activities.

Commitment to perform

Commitment 1: The delivery group leader coordinates delivery activities between customer and organization

The *delivery group* is a group (or a person) which is formed at the time of delivery of each project

product. After a successfully concluded delivery, the group can take over other responsibilities and tasks. The delivery group usually involves the personnel, who know the product in detail as well as the personnel responsible for the documentation and communication with the customer.

Coordination between customer and organization at the point of delivery is needed in order to:

- define schedule and resources for the delivery;
- define schedule and resources for assistance with acceptance testing;
- define schedule and content of end-user training.

Commitment 2: Customer's management commits that the delivery group will have access to all places and resources needed to install the delivered product

Ability to perform

Ability 1: The delivery group is skilled and trained to perform the delivery activities
Delivery group members should be familiar with all details of the delivered product. They also should have a knowledge of the target environment (existing HW and SW) to perform the installation correctly.

Ability 2: Appropriate resources and financing are available for the performance of delivery activities

Activities performed

Activity 1: The delivery group checks the completeness of the software product (and all its copies) prepared for the delivery
The software project prepared for the delivery is complete if all parts (programs, technical documentation, end-user documentation etc.) that have to be delivered are present in a proper form. When more than one copy of the product is delivered, correctness and completeness of each copy should be checked.

Activity 2: The delivery group installs the product in the end-user environment
The delivery group should perform the installation of the product and check the correctness and the completeness of the installation. If the product is installed by the customer or the end-user themselves, the delivery group should check if information required for the installation is given in the enclosed documentation.

The impact of the installed product to other HW and SW products in the end user's environment should be checked.

Activity 3: The delivery group assists the customer with acceptance test planning
The customer should (with the assistance of the delivery group, if needed) define:

- schedule of acceptance testing;
- procedures for evaluation of the delivered product;
- resources needed to perform acceptance testing;
- acceptance criteria (they should already be defined in the requirements specification and the contract).

Acceptance testing is usually performed in the target environment, therefore the installation of the product should be performed before the acceptance testing.

Activity 4: The delivered product is formally accepted
After the successfully performed acceptance testing, the formal acceptance of the delivered product should be concluded. The customer and the organization sign a document which shows that the delivered product fulfills the requirements stated by the customer and that it has passed the acceptance testing. The formal acceptance of the delivered product is important also in order to determine the initial state of the product, which is needed for the maintenance of the product.

Activity 5: Training of the end-users should be performed
The end-users should be trained to:

- use the functions of the product;
- solve the most common problems that often occur after the delivery of the product (like changes of some system parameters etc.).

Activity 6: Delivery activities should be documented
The records of the delivery activities should be maintained.

Measurements and analysis

Measurement 1: Measurements should be performed and used for the definition of the status of the delivery activities
A typical measurement of the effectiveness of the delivery activities is the number of required interactions after the delivery of the product and reasons for these interactions.

Verifying implementation

Verification 1: Top management and the quality manager should verify the effectiveness of the delivery activities
Verification can be performed periodically or on an event driven basis (e.g. within the specific period after the conclusion of a certain project).

REFERENCES

Finkelstein A., Kramer J., Nuseibeh B. 1994. *Software Process Modelling and Technology*. Research Studies Press Ltd: Taunton, England.

Fuggetta A., Wolf A. 1996. *Software Process*. Trends in Software 4, Wiley: Chichester.

International Organization for Standardization. 1991. ISO 9000–3, Guidelines for the application of ISO 9001 to the development, supply and maintenance of software. International Organization for Standardization, Geneva, Switzerland.

International Organization for Standardization. 1994. ISO 9001, Quality Systems – Model for quality assurance in design/development, production, installation, and servicing. International Organization for Standardization, Geneva, Switzerland.

Johnson D.L., Brodman J.G. 1997. *Tailoring the CMM for small businesses, small organizations, and small projects. Software Process Newsletter* **Winter** (8).

Paulk M.C., Weber C.V., Garcia S., Chrissis M.B., Bush M. 1993a. *Key Practices of the Capability Maturity Model, Version 1.1*. Software Engineering Institute, CMU/SEI-93-TR-25, February.

Paulk M.C., Curtis B., Chrissis M.B., Weber C.V. 1993b. *Capability Maturity Model for Software, Version 1.1*. Software Engineering Institute, CMU/SEI-93-TR-24, February.

Paulk M.C., Weber C.V., Curtis B., Chrissis M.B. 1995. *Capability Maturity Model: Guidelines for Improving the Software Process*. Addison-Wesley: Reading, Massachusetts.

Rout T.P. 1996. *The SPICE project – Past, Present and Future*. SPICE 96 Conference Proceedings, Brighton, 3–5 December: 5–15.

Rozman I., Horvat R.V., Györkös J., Heričko M. 1997. *PROCESSUS – Integration of SEI CMM and ISO quality models*. *Software Quality Journal* **6**: 37–63.

Sommerville I., Rodden J. 1996. *Human, Social and Organizational Influences on the Software Process*. Trends in Software 4, Software Process, Edited by A. Fuggetta and A. Wolf. Wiley: Chichester.

EDITOR'S COMMENT

This paper proposes an improvement model applicable to small software companies (or small groups within larger companies). The requirements set for the model were that it should be simple and have a short implementation cycle time. Using principles from both the Software CMM and ISO 9001, it pleasingly combines essential features of the CMM's staged roadmap and the IDEAL cycle. It concentrates (appropriately, many may think) entirely on improvement and not at all on assessment.

The model was developed by a university group working with industrial organisations. It is a good (but sadly rare) example of how an academic research group can play a key role in producing results of high potential value to industry. Given its limitations in terms of length, this paper has a "tip of the iceberg" feeling about it: it would be really good to know more about the take-up of the proposed model and its effects on those who may have experimented with its use. Too often, good work of this kind has a hard time winning acceptance outside an initial circle of enthusiasts. Within our field, we need better mechanisms for publicising and critically evaluating such innovations.

—CT