SOFTWARE PROCESS: IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. 2000; **5**:55–64 (2000)

Linking Software Process Improvement to Business Strategies: Experiences from Industry

Practice Section

Christophe Debou[1,*] and Annie Kuntzmann-Combelles[2]
[1]*Q-Labs GmbH, Europaallee 9, PRE-Park-Holtzendorff, D-67657, Kaiserslautern, Germany*
[2]*Q-Labs, 28 Villa Baudran, 94742 Arcueil cedex, France*

The major bottleneck to the success of an SPI initiative is the lack of business orientation in the way the program is run. The post-assessment phase or how to translate assessment results into adequate improvement actions is being overdocumented compared to the assessment phase. In this article experiences and lessons learnt from two industry sources in the defense and telecommunication areas are covered. Improvement strategies based on ensuring a continuous link between the program and the business goals of the organization is advocated (e.g. the ami® approach). Copyright © 2000 John Wiley & Sons Ltd

KEY WORDS: ami; measurement-based improvement; business orientation CMM

1. INTRODUCTION

'Software process improvement initiatives have suffered from a lack of business orientation'. This statement is often being repeated at conferences and forums to explain the high percentage of SPI programs that fail. Failing means that the impact on organizational performance is low or non-existent. What the SPI community has documented so far almost exclusively relates to the assessment process and the model to evaluate against. Several assessment methods (e.g. Bootstrap, SPICE, Trillium) have been derived from the original SEI work (Capability Maturity Model (Paulk *et al.* 1993a,b)). Owing to the lack of documentation on the post-assessment phase, assessments are often being performed as a one-shot event without connection to any improvement strategy.

In fact up until the last couple of years, we have missed such improvement strategies. A key step, upon which the whole success of the initiative may depend, is situated at the end of the assessment during the translation of the assessment results into concrete improvement actions, namely improvement action planning. This exercise is much trickier than trying to cover immediately non-existent or deficient CMM practices. In this paper, experiences and lessons learnt from two industry sources on how to ensure the link to business will be covered:

1. Major actions undertaken to eliminate typical SPI bottlenecks within the frame of a corporate-wide initiative in a leading telecommunication company.
2. Application of a goal-oriented and metrics

*Correspondence to: Christophe Debou, Q-Labs GmbH, Europaallee 9, PRE-Park-Holtzendorff, D-67657, Kaiserslautern, Germany.
E-mail: christophe.debou@q-labs.de

driven improvement strategy, i.e. ami® (Pulford *et al*. 1995) in a defense company.

The intent of this article is not to give the full details of improvement programs that are very much specific to an organization and not replicable to others, but to introduce the principles that have been followed and some analysis of difficulties and successes.

2. WHY SPI PROGRAMS TYPICALLY FAIL?

Industry has identified three major bottlenecks to the success of an SPI initiative. The one that accounts for failure the most is the lack of management commitment and understanding. This is typically reflected by an inadequate allocation of resources to the program:

- too few resources or multiply shared resources that are insufficient;
- inappropriate SEPG staff: the same situation occurs as did some years ago with quality assurance engineers, where the staff allocation was based on availability and not skills.

Furthermore, there is still a belief that SPI is a level certification exercise. It seems that the ISO 9001 pitfalls have been forgotten. Hence goals are set to reach a certain level of maturity within *Y* years. This is fine until it is perceived as the ultimate goal, instead of a quantitative improvement goal. Some uneducated customers have also contributed to the situation, auditing suppliers resulting in marks like 1.3 or 2.4.

The second major bottleneck (a consequence of the first) is the late impact of the program on projects both for daily practices and performance. Management starts to lose patience and practitioners lose the momentum gained during the assessment. The time to produce an action plan should be kept to a maximum of 3 months. A typical mistake is to try to plan for the next 2 years whereas a 3–5 month detailed plan in the view of mid- to long-term goals is more appropriate. Furthermore, often the focus of improvement actions is much too wide (trying to cover all KPAs at a time) or just wrong.

The third major bottleneck, that is, the lack of software management skills, is certainly more applicable to industries that have a strong hardware culture (i.e. telecommunication, defense), although even in the software industry, the skill development emphasis has been mainly on technology compared to process.

Let us look at the causes. The number one cause is the general lack of business orientation of the program and the actors, reflected by a missing link with business strategy and goals. SPI actors have been giving too much emphasis to assessment processes/model/questionnaire versus improvement strategies, showing their lack of pragmatism and business understanding. Some extreme examples have shown that the only justification for actions is a KPA profile showing a gap in percentage between the assessment results and an ideal model; this is unacceptable from a community that advocates more professionalism in software engineering. Furthermore, one should remember that level 2 issues are management issues that cannot be solved by people with purely technical skills. Finally, there is a tendency to focus first on organizational-wide processes (skipping level 2) instead of project-related improvements. This typical ISO 9001 behavior has shown an inefficiency to drive self-improvement programs.

As a provocative conclusion, it is time to say loudly that SPI is much more than investigating whether CMM is more appropriate than SPICE. We start to see improvement strategies emerging that tend to improve the understanding in SPI principles: SEL Quality Improvement Paradigm – Experimental Software Engineering (Basili and Rombach 1988, McGarry *et al*. 1994); ami® (Application of Metrics in Industry) Improvement Cycle (Pulford *et al*. 1995, Debou *et al*. 1995); SPICE Guide for Use in Process Improvement (SPICE consortium), and SEI IDEAL (McFeeley 1996).

3. EXPERIENCES FROM A CORPORATE-WIDE SPI INITIATIVE

The top-level management of a leading telecommunication company decided in 1994 to initiate a continuous software process improvement program using the SEI methodology to better address new challenges. The rising complexity of the systems was increasing the risk to deliver lower quality or with unacceptable delay that would imply high financial penalties.

A set of principles based on success stories was devised to drive the initiative:

- The Capability Maturity Model will be the basis for assessing the software development capability and to guide but not dictate the improvements.
- All major development centres have to be assessed before they can produce local action plans (self-assessment supported by external coaches). Training of management and SPI staff has to be performed beforehand.
- Software Engineering Process Groups (SEPG) have to be established in each development centre together with a management steering committee.
- A dedicated budget should be allocated to SPI activities (about 3% of R&D effort).
- All SPI actions within a business division will be co-ordinated by a software process manager reporting to both local technical director and corporate SPI management.
- Business and process performance indicators should be established to continuously evaluate the impact of changes on projects and on the organization.

Although perfectly sound, such principles have not been enough to ensure the success of the initiative. The environment of implementation is very complex (development centers ranging from 50 to several 100 engineers around the world) and just because the top-level management committed to it, it is not enough to make it happen. The three previously listed bottlenecks encountered early in the process have allowed us to tune the strategy.

3.1. Management Commitment = Business Orientation

3.1.1. Obtain Upfront Business Issues/Goals from Top Management Team with Regard to Software
Awareness on the importance of software and the necessity of investing in SPI should be raised. Guided interviews with local management (business, marketing, technical and quality) are organized to understand what are the priority business drivers between quality, cost, schedule or product issues. There has been a high correlation between lasting programs and business management involvement (and understanding) from the beginning. If the organizational readiness is perceived as low, the program should be postponed.

3.1.2. Consider Full R&D Cycle with Critical Interfaces, e.g. Product Management
To get acceptance from everybody in the organization, the whole R&D (and not only software) should be investigated as well as interfaces that usually cause problems.

3.1.3. Do Not Perform Systematically Full Assessment
A full three-week assessment does raise a lot of expectations in the organization. Management is being publically confronted with issues and is put in front on its responsibility to initiate actions. If nothing happens, practitioners will lose trust in management and will not believe any more in such kind of initiatives. Criteria like size of the development organization, initial 'maturity' and readiness in terms of commitment can help in deciding which approach to select: full assessment; dedicated assessment on given domains or key process areas; awareness training, or even software engineering training.

3.1.4. Avoid the 'Bible' Approach ('CMM and Nothing Else')
Such a kind of model should never be taken as granted and needs to be tailored. During an assessment performed in a Scandinavian country, the management was presented with SPI concepts and was asked whether other aspects outside the CMM should be investigated. They mentioned that they were getting complaints from customers because during a new release installation the switch downtime was much too long and perturbing the traffic. A set of interviews was added to the assessment to specifically cover that issue. Flexibility, without bypassing SPI principles, is a key element to gain management trust.

3.1.5. Have Business/Technical Performance Indicators Defined and Collected Up Front
The collection of historical performance data (schedule, cost, quality) should be one of the first improvement actions within a rather stable organization (no major technological or business domain change for the last couple of years) to achieve a baseline for comparison. Both external (visible to customers, like number of complaints

or time to deliver) and internal indicators (e.g. defect detection rate/phases) should be collected. In environments where former projects are irrelevant in terms of contents, size or technology (of development), only fundamental project management metrics are defined together with improvement actions and the on-going lifecycle phase.

3.2. Quick Impact on Projects

3.2.1. Select Business-Critical Projects as SPI Customers

As emphasized previously, the first focus is projects before the whole organization. They are the primary customers. Since the success of such an initiative is mainly linked to business impact, it makes no sense to select as a customer project a non-strategic project generating small revenue. This is typically what is happening because strategic projects cannot be disturbed, implying that improvement is a disturbance! The SEPG should work with the project leaders to identify concrete improvement actions (e.g. risk identification at the initiation of the project) and support the implementation. The generalization to other projects can happen in a second step.

3.2.2. Focused Effort

Typically, working groups on all CMM KPAs are started up. This is not working efficiently. We had more success initiating a task force with specific projects according to the lifecycle phase they are about to start.

3.2.3. Regular Introduction of Sets of Concrete Practices into Projects (Incrementally)

During the action planning, increments containing a set of well-defined practices to be implemented in a 3–4 month timeframe are defined. Each increment is composed of a preparation phase, a piloting phase on a customer project and a deployment phase across the organization. Such increments can be implemented in cascade.

3.2.4. Initial Support by Approved External Consultants

When the software engineering expertise is low, it can be very efficient to bring in an external consultant for a long period (at least 6 months) who will support the SEPG during the preparation and coaching phase, according to previous principles. The consultant's main contribution is to avoid the organization reinventing the wheel. A lot of templates, guidelines etc. are available; they need to be understood in order to be able to support the adaptation and implementation.

3.2.5. Fostered by Corporate Essential Practices

Based on assessment results, the corporate SPI support group had identified a set of 10 essential software management practices that should form the kernel of actions. Examples of practices are use of history data for estimation and top 10 risks identified.

3.3. Skills Improvement

3.3.1 Educate Management on SPI Concepts and Expectations

It is essential to brief up front senior management on the rationale behind SPI, the whole improvement process and timeframe, the costs and the potential benefits in the short, mid or long term.

3.3.2. Inject External Expertise into Projects for Coaching Improvement Actions

When adequately managed (transfer of knowledge to SEPG planned, full-time presence for some months, consultants trained and followed up by corporate) this approach is very efficient. This is particularly suitable for accelerating the impact of the program on the projects where the software engineering skills are low.

3.3.3. Train Project Leaders in Essential Software Management Practices

Such training is not intending to teach project leaders state of the art but to provide them with a set of essential principles, rules illustrated by some selected well-known techniques, that they can straightaway apply at the end of the course. Such a course should cover level 2 areas like requirements engineering, project management, quality assurance and configuration management.

3.3.4. Cross-Business Division Experience Exchange – SPI Workproducts Exchange

In a case of a multi-site multi-national company, a lot of redundant effort is saved through the exchange of experiences, SPI workproducts and

possibly people. Sending a template might be useful but bringing together with the templates the external consultants or the local SEPG accelerates the necessary tailoring and helps in breaking 'never invented here' barriers. However, a complete corporate infrastructure (people, funding, database, www server) should be put in place to allow 'enforcement' of such an exchange.

3.4. 'Two Years Later' – Feedback

After 2 years of an intensive improvement program, a cultural change has been observed, especially in the way management behaves: almost no political deadline any more or 'real' commitments make the big difference with the previous culture and is one of the main success factors. The most visible business consequence is the delivery of the major yearly release on time with half of the defects in the field. Furthermore, business goals are well understood across the organization. The entire approach has been extended to include more intensive product management and an assessment of product management has been conducted in a pilot phase. The entire organization understands the importance of improving capability not only for the software business but for all the disciplines which have interfaces with software. On the other hand, the measurement program in place has demonstrated the efficiency of practices but is still focusing too much on management; development team members require more quantitative visibility on their day to day work in order to be sure that business goals are met. This is a real sign of culture change and the insurance that software process improvement has happened.

4. APPLICATION OF A GOAL-ORIENTED IMPROVEMENT APPROACH: ami®

This case study demonstrates how to use the ami® concepts to build an improvement action plan after a CMM-based appraisal and how to monitor it for success.

The company is a defense market actor in Europe named C (for confidentiality reasons, we are not allowed to give more details). The software development staff is about 70 people, less than 10% of the total number of employees including commercial people, system engineers and hardware design people. Software is one of the growing components of the systems developed.

The software part increase is about 20% each year and the software business will be considered as one of the key factors for success in the next decade. However, software development still provides surprises and is sometimes considered as a group of artists who may either make significant profits or lose a lot of money without a clear explanation of why. Therefore, mastering the software process is one of the major challenges of the near future. Software process improvement might be one of the key success factors of the competition if organizations ensure that the impact of decided actions is able to provide benefits quicker than the external environment evolves. In other words, any SPI initiative is able to win if and only if improvement actions are addressing in a timely manner the strategic marketing objectives of the organization.

The C company performed a professional CMM appraisal in 1994 and defined an action plan by the beginning of 1995. Discussions to convince the SEPG manager that the ami® approach was able to contribute positively to the actions already in place lasted about 6 months; confidence was obtained after we had the opportunity to meet top managers (executive VP, Product strategy manager and QA manager) and discuss strategic goals for SPI. These people were enthusiastic about the idea to document the link between their vision of the business and the practices used to develop software. They had obviously no deep knowledge of software development issues but they explained what the key drivers were for the next 2 to 3 years in their specific business. The consultant's job was to ask the right questions in order to be able to derive improvement goals and make the transformation very clear between business strategy and software process strategy. These questions were prepared based on both the G/Q/M paradigm and on the 1994 assessment findings. Examples of questions follow:

1. Would you please explain which products should be strategic in the next 2 to 3 years? Why?
2. Considering this market strategy, would you say that software business is a key business

(software component a key component)? Why? What is the added value provided by software?
3. What major changes do you expect in the near future? Would they impact the software characteristics and how? Technology, competitors, new customers, new types of users, standards, internationalization, etc.
4. Would you please explain the cost structure for the development of a new product (or new version of an existing product)? Maintenance cost?
5. Would you please explain what type of goals you defined regarding costs and if software is part of them?
6. Would you please tell if you would need a different level of visibility in the costs and why?
7. Would you please explain how you define and measure the quality of the product?
8. What is the cost of non-quality? How is it calculated?
9. What are the most critical factors for your customers?
10. How will you characterize a successful business now? In 2 years?
11. How will you manage trade-off between cost and quality?

Discussions with top managers made very clear that maintenance costs – evolutive and corrective – of the applications were very high and prevented teams doing all the work they were supposed to do in order to satisfy the customers with new features. At least top managers had this feeling but at the same time were unable to produce any quantitative data to support their judgment. Half of the project leaders shared this view and half did not; they had other good explanations of the overload. Therefore, the initial improvement goal was 'to decrease the maintenance cost'.

So far assessment findings had been related to actions but nobody had checked their impact on the business performance. Hence the complete process of defining actions based on the assessment findings was restarted using ami® concepts. The very first task was to train all involved people – from middle managers, project leaders to development teams – to ami® main concepts, i.e. the four phases of the loop, the importance of measurements to be pro-active and the G/Q/M paradigm. This one-day course has later on been extremely useful, helping all the participants to share the same language and understanding.

Furthermore, a certain number of brainstorming meetings was organized. Consultants played the role of a catalyst for the organization. This was a more efficient way of transferring the technology and to help the organization to acquire the capacity to reproduce this process later on. Meetings were scheduled over a 3-month period.

- The first meeting was dedicated to analyzing business goals as expressed by top managers, to put priorities among them and to translate them in terms of project goals or software related goals. Brainstorming and complementary interviews with customer-related teams were performed. The result was a first level of decomposition of primary goals into more technical goals. Available figures were used to consolidate the thinking process. The result was the goal decomposition. The improvement team had to validate this decomposition with all involved people before the second meeting (see Table 1).
- During the second meeting an impact analysis exercise was performed based on the validated sub-goals. Each assessment finding and set of recommendations was carefully discussed to identify which type of recommendations could drastically improve the situation. This type of analysis and judgment is largely based on the wide experience of the assessment team. It is not always easy because there are very few one to one relations. In other words, several weaknesses, if corrected, might have a positive impact on the same goal. As a result, it is extremely difficult to draw the map and this

Table 1.

Reduce cost
1. Manage the stability of requirements
1.1 Reduce the number of change requests
2. Acquire the knowledge of cost decomposition
2.1 Analyze the reliability of initial estimates
2.1.1 Identify the deviations and origin
2.2 Identify how the effort is split along the lifecycle
2.2.1 Define effort/task
2.2.2 Identify rework effort
2.2.3 Identify the availability of resources

map is certainly not applicable in various contexts. So we have to fight against the statement that this job of linking weaknesses to business goals is repeatable from project to project and even from organization to organization. Environmental constraints and cultural aspects are some of the many explanatory variables you can find in the process. Again, here, G/Q/M is extremely helpful in structuring the brainstorming for more efficiency and better quality.

Table 2 gives an overview of one of the interim results obtained during this second meeting. Weaknesses observed during the assessment were formalized based on the top level goals and subgoals. Only the weaknesses which, when corrected would make improvement, were further considered.

Table 2. Main weaknesses per KPA

CMM KPA	Weaknesses
Requirements management	• change requests insufficiently formalized • impact analysis not systematically achieved and poorly documented • decision to implement changes not necessarily taken at the right level • absence of process measurement: e.g. number of changes addressed/quarter, average number of days necessary to make the change, lead time between request and installation of the new release
Project management	• lack of precision of the effort tracking system • no historical data available (project leaders have their own data) • SW development plan exists but is not updated regularly • risk evaluation and management informally performed • corrective actions taken late, most of the time result in an increase of resources hired from other projects or software houses • overall planning (synchronization between projects) poorly achieved

After the second meeting, the SEPG team (i) validated the results obtained and (ii) listed the top few actions, which should bring the biggest benefits in order to improve the business performance quickly. Consultants reviewed their conclusions based on experience and knowledge of the context.

- The third meeting was dedicated identifying actions (the 'how') to implement the recommendations (the 'what') which could really help to match targets. It is clear that the impact analysis and the goal decomposition continuously maintain the traceability with top level objectives. At the same time actions were allocated to specific roles in the organization. It was verified whether these key roles were involved in the achievement of the business goals.
- The fourth meeting concentrated on the definition of metrics to track defined actions. Between meetings 3 and 4, a huge effort was allocated to the accurate definition of each action, budget estimation, definition of mechanisms to pilot it and finally to measure efficiency and adequacy. This has been mainly performed by the organization with some inspections from the consultants in order to verify overall consistency and time frame for results. These progress metrics were aligned with business targets stated by top managers. Wherever they had been able to quantify, quantitative metrics were systematically defined; when not, either quantitative or qualitative indicators were suggested depending on the experience of the software process group.

Some examples of these indicators are given in Figure 1. Type A projects appeared quite unstable; Q3 and Q4 95 recorded heavy deviations for this population mainly due to a major change in technology. This family – decision making systems – moved to OO design at that time. The deviation's origin was twofold: a bad estimate when the projects started because estimation procedures had not been reviewed according to new technology, and some technical difficulties faced because of the lack of experience of the teams.

Type B (embedded systems in a rather stable environment) on the contrary showed a decrease of the deviation rate.

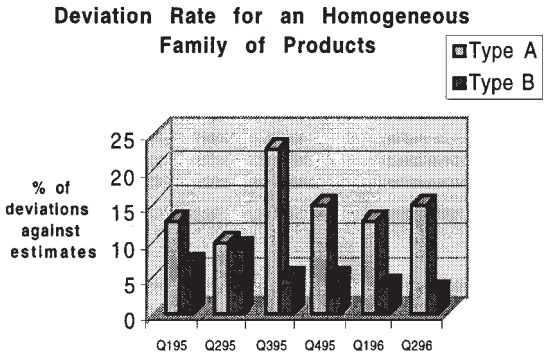

Figure 1. Estimation accuracy

Figure 2 shows a decrease of this ratio over time based on a policy to address change requests at a unique point, i.e. the change control committee, meeting every two weeks.

Figure 3 shows an important decrease of effort

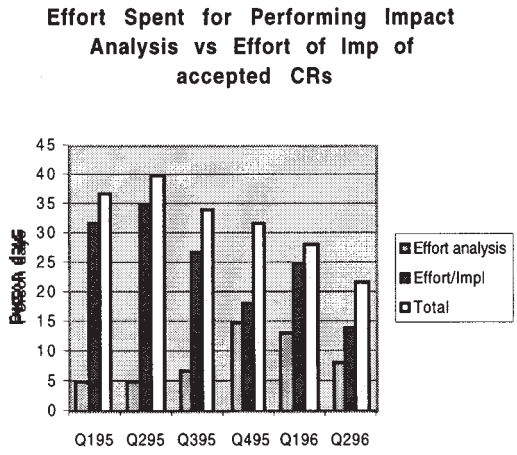

Figure 2. Effort of analysis versus implementation

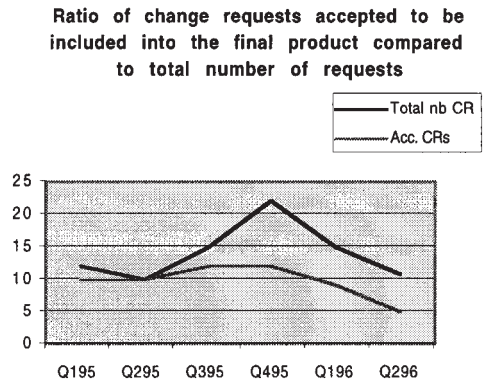

Figure 3. Ratio of accepted change requests versus total number of CRs

over time. A change review board was institutionalized during Q1 96 and its impact is clearly visible on the maintenance effort. The results of the boards were regularly handed over to customers and the reasons for postponing or ignoring some of their requests explained to them based on real and technical data.

It is important to outline that along with meetings 3 and 4 (end of 1995), existing data started to be recorded and exploited by the SEPG; therefore, better understanding of the weaknesses and action impact measurement were achieved concurrently. No new data were defined for collection, but data which were already collected for other organizational reasons were used.

Once the improvement/measurement plan was completed, the metricate stage could start. The plan was first deployed on three pilot projects to validate the mechanisms for data collection and exploitation. The observation period was 3 months with data collected regularly, some collected during reviews at the end of a development phase and some by the end of the project. Together with the mechanisms we wanted to consolidate the budget estimation for the improvement and measurement.

This pilot phase was also judged very positive in as much as it raised motivation among the participants who then communicated their enthusiasm to the rest of the software community. Some minor adaptations were necessary in the measurement plan in order to get a completely adapted collection mechanism and to avoid duplication with the existing global system for cost management.

4.1. Results

Models to be reproduced are the ones keeping a continuous trace of the business goals and top level managers' expectations or the introduction of simple metrics. What could be improved was some CMM coverage, i.e. following more closely the underlined road map. Because there was a potential risk that the company C tried to implement each activity of the CMM and forget business goals, we took exactly the opposite approach; there could be opportunities for a more efficient one.

When the actions have been decided and allocated to groups of people, it was observed that: (i) the motivation was high – people wanted to

reach their goals; (ii) the justification of why to do it and how to do it were better understood by a majority of people because the links with the organization strategy were documented. Because the understanding was better, people responsible for action implementation were stronger in promoting them among the development teams.

The approach was reproduced in another more mature organization with similar benefits. However, because the initial starting point was a higher level of maturity, product line managers were involved in the assessment and in the improvement planning in order to better consider the gap between market strategy and software process improvement. The result was a stronger emphasis on customer relationship weaknesses – inappropriate, not at the right time – which is not well covered in CMM V1.1.

5. CONCLUSION

Those two industrial examples have shown the necessity to link any software process improvement program to business goals and to follow it up closely with indicators related to the initial goals. Those simple principles have so far been seldom followed. One reason lies in the lack of emphasis on those issues from SPI suppliers. It might also be related to the inherent characteristics of engineers to tackle any problem as a technical problem. Most of us have a technical background and will be most confident in techniques and technologies. SEPG working groups are composed of technical people who have been appointed to lower management or mid-management positions according to their technical results, often without training in managerial issues such as communication, team motivation or value added analysis. At level 1–2 of the CMM layer, we are dealing with management-related issues: estimating; planning; controlling; tracking; roles and responsibilities; commitment. When developers discover the maturity model they generally ask questions about software engineering, tools and methods: where do we find guidelines for these aspects if not at level 2? Firefighting at level 1 organizations generally means find a tool or a programming trick or a COT which will solve the problems. Maybe planning risk management does not excite engineers enough?

REFERENCES

Basili V, Rombach H. 1988. The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering* **14**: 758–773

Debou C, Fuchs N, Haux M. 1995. ami: a tailorable framework for software process improvement. Proceedings of the second ISCN Seminar, Vienna, September.

McFeeley B. 1996. IDEAL: *A User's Guide for Software Process Improvement*. CM/SEI-96-HB-001, Software Engineering Institute.

McGarry F, Pajersky R, Page G, Waligora S, Basili V, Zelkowitz M. 1994. *Software process improvement in the NASA Software Engineering Laboratory*. CMU/SEI-94-TR22.

Paulk MC, Curtis B, Chrissis MB. 1993a. *Capability Maturity Model for Software*, version 1.1. CMU/SEI-93-TR-24.

Paulk MC, Weber CV, Garcia SM, Chrissis M. 1993b. *Key Practices of the Capability Maturity Model*, version 1.1. CMU/SEI-93-TR-25.

Pulford K, Kuntzmann-Combelles A, Shirlaw S. 1995. *A Quantitative Approach to Software management – the ami Handbook*. Addison Wesley.

SPICE consortium. 1997. *Guide for Use in Process Improvement*. ISO/IEC JTC1/SC7.

EDITOR'S COMMENT

This paper contains valuable and important practical guidelines from two experienced software process consultants. Its theme is the critical importance of ensuring that assessments are performed not just for their own sake but as part of a well-planned improvement strategy that is linked to business goals. Two case histories, from the telecommunications and defence industries, are introduced to provide illustrations of that theme.

The authors identify three major causes of failure ("bottlenecks") in SPI: lack of management commitment, slow impact on projects, and lack of software management skills. All three are symptoms of a "general lack of business orientation" on the part of key players in SPI initatives, who pay too much attention to assessment at the expense of improvement. The first case history describes how the telecommunications company laid down "a set of principles based on success stories" to drive its initiative, but found it needed to amplify them with additional policies explicitly designed to combat the bottlenecks.

In the second case history, the defence company successfully deployed the **ami** approach (based on the Goal-Question-Metric paradigm) "to document the link between their vision of the business and the practices used to develop software" and to identify "progress metrics [that] were aligned with business targets stated by top managers".

—CT