7 SOFTWARE PROCESS IMPROVEMENT: WHAT MANAGEMENT TENDS TO FORGET

Ivan Aaen
Jan Damsgaard
Aalborg University
Denmark

Abstract

It is of great interest and importance for organizations and management alike to clarify mechanisms that can sustain or impede the success of software process improvement (SPI). This paper analyses how best to promote SPI initiatives and increase the likelihood of adoption of SPI innovations. We develop a framework for management to choose appropriate activities to further SPI projects and increase their likelihood of success. As a theoretical device, we apply institutional theory. We distinguish between two dimensions of managing SPI adoption and diffusion. In one dimension, we focus on the exchange of ideas and technical and scientific knowledge between providers and adopters of SPI initiatives within a software producing organization. This exchange can be characterized as driven by two complementary forces: namely, supply push and demand pull. The other dimension captures the mode of how management may engage in managing SPI projects. In this dimension, we distinguish between influential and regulative measures. Six different types of management activities are placed in the two-by-two matrix. Two software-producing organizations that are heavily involved in SPI are analyzed using the framework. The analysis shows that SPI projects are managed mainly through standard setting activities and rigid innovation directives, and that there is plenty of room for further management involvement both on the SPI provision side and the adopting side. One *path ahead to more successful SPI projects involves the usage of a larger spectrum of management activities, for example through subsidies or modifying existing incentive structures.*

Keywords: Software process improvement, management strategies, institutional analysis.

Introduction

Software is increasingly an important part of our everyday life and a driving force in most industries. Simultaneously the mass media abounds with reports of software problems. The year 2000 problem, the crash of the Ariane space shuttle, expensive missile aborts, and enormous financial losses are only a few examples of problems blamed at inadequate software. Hardware advances have for **decades** outpaced advances in software, and software productivity lags behind business and market needs. Furthermore, software is often delayed, or it is of poor quality or functionality, or the costs of the software sometimes exceed the expected by several hundred percent. These observations together constitute what has become known as the software crisis (Pressman 1997). The crisis has led to a quest for silver bullets (Brooks 1987), like high-level languages, programming environments, object-orientation, and program verification, to be fired at the monster. By now it is part of conventional wisdom that we can never mold a silver bullet to kill the monster: The problems at the basis of the software crisis are much too complex to be defeated by any discrete innovation. In line with this, much of the contemporary research on software engineering is integrative by nature, combining several approaches into more comprehensive frameworks.

One prominent example of this is software process improvement (SPI) aiming to increase the maturity—i.e., the richness and consistency—of the software processes in organizations (Humphrey 1989). Some advocate that SPI will decrease the time it takes to develop and maintain software, reduce the software costs, improve the quality of software products, expand the control and predictability of performance, and add to the capacity to produce wealth (Software Productivity Consortium Services Corporation 1993). No wonder many software organizations have turned to SPI as a rescue from insufficient quality, exploding budgets, overworked employees, and blown schedules.

A software process is a set of activities, methods, practices, and transformations that people use to develop and maintain software and the associated products—e.g., project plans, design documents, code, test cases, and user manuals (Paulk, Weber et al. 1993). Thus the process combines people—developers and managers—tools, and procedures. Together this combination affects the cost, schedule, and quality performance of development projects.

SPI is based on any one of several models for software process maturity. The first and still most widespread of these maturity models is CMM—the capability maturity model (Paulk, Curtis et al. 1993; Paulk, Weber et al. 1993) Ever since its first presentation CMM has been extremely influential on software engineering practices around the globe. The model has served as a framework for software process and quality improvement efforts in thousands of software organizations, and the resources expended on CMM-based SPI are in the billions of dollars (Herbsleb et al. 1997).

The CMM defines a framework for how an organization matures its software process through SPI. These levels describe an evolutionary path from ad hoc, chaotic processes to mature, disciplined software processes. Specifically, the CMM defines five levels of process maturity that represent distinct plateaus of organizational performance.

Although SPI is in widespread use, only a limited amount of research on experiences has been published. Most of these studies are case studies where experiences from individual organizations are presented (e.g., Diaz and Sligo 1997; Haley 1996; Hollenbach et al. 1997; Humphrey, Snyder and Willis 1991). Such studies generally report on SPI successes showing dramatic improvements with respect to a number of performance parameters, for example, productivity and cycle time, while studies reporting on SPI failures are rarely found. On the other hand, data is available that indirectly indicate that SPI is a time consuming activity with a high risk of failure.

One such source of data is the semiannual reports of the Software Engineering Institute (SEI). The latest of these reports (Software Engineering Measurement and Analysis Team 1998) states that organizations generally use two years to take the first definitive step up the maturity scale. Of the 782 organizations that have reported assessment data to SEI's database, only 172, or 22% have been assessed more than once, indicating that a majority of the first time assessed organizations do not seem to proceed with a second assessment.

Despite the lucrative promises, SPI thus seems to be less successful than could be expected. There are several reasons for its modest success. In many cases, SPI has been very costly to implement and the skills to design and operate SPI projects have been scarce, or management has not adequately supported the SPI efforts. Accordingly, expected benefits of introducing SPI have not been realized.

It is therefore of great interest and importance for organizations and management alike to clarify mechanisms that can sustain or impede the success of SPI projects. Most literature points to management as a crucial component in facilitating successful SPI projects (Paulk 1996). In this paper, we develop a framework for management to choose appropriate activities to further SPI projects and increase their likelihood of success. As a theoretical device, we shall apply institutional theory (Damsgaard 1996; Damsgaard and Lyytinen 1996; Damsgaard and Scheepers 1998, King et al. 1994; Tornatzky and Klein 1982).

The remainder of this paper is organized as follows. First we describe software organizations and SPI and its nature as an innovation. Thereafter we identify different stakeholders that can be active in the innovation and adoption process and discuss the significance of management in shaping their deliberations. This is achieved by presenting a framework in which management involvement in SPI projects can be analyzed. Generally there are six types of activities that management might consider to "lubricate" the course of SPI projects. We use the framework to analyze two empirically based SPI studies. We then synthesize what is known about management involvement. Each form of strategy is discussed in terms of its malleability and likely success. Finally, we draw some conclusions and suggest areas of further research.

Software Organizations

We consider an organization a microcosmos bounded by some area (geographical, industry, or market segment) and consisting of a number of actors ordered in some hierarchical and functional manner—e.g., management, production, support functions, and technology development (Damsgaard, Rogaczewski and Lyytinen 1994). Each individual and group is regarded as self-directed in the pursuit of some selfish ends (Perrow 1986).

Software producing organizations produce either for the market directly or for one or more customers in some hierarchical manner. Software organizations usually employ a well-educated and highly specialized work force, and they are relatively young organizations with an often informal, flat, and organic structure.

The Dyadic Nature of SPI Projects

As with other types of organizations, a software organization cannot readily apply innovations that are available in the market. Innovations have to be adapted to local contingencies and adoption requires learning (Attewell 1992). The "bringing in" of innovations often rests with a specific department within the organization. This department is involved in designing and controlling organizational products and processes (Minztberg 1983; Porter 1985), and also in SPI projects. It is often a standing entity (i.e., an institution within the organization) but it may also be a more dynamic constellation with altering actors drawn from different parts of the organization. The main objective of the department can be grouped into a number of activities, such as standardization of the work flow, training of the work force, developing new technologies, creating new innovations, or designing organizational structures or processes.

This method department usually does not possess any formal authority, it greatly relies on power dependencies, persuasion, and management support to implement its ideas. In other words, it must sell its innovations to potential adopters (Markus 1983). This creates a supplier and demand relationship between the method department on the one side, and adopters on the other.

The diffusion of innovations always takes place in an environment consisting of entities like management, production, method departments, and sales and service departments. These entities affect and change interactions, and thereby they may constrain or enforce innovation. One notable feature is the role of management, which retains the formal power to intervene and thereby alter the interactions between these two semi-interdependent entities and other stakeholders as well.

Drivers of Innovation

The exchange of ideas and technical and scientific knowledge between suppliers and adopters within software producing organizations can be characterized as driven by two complementary forces: namely, supply push and demand pull. Push is created by the method department, which identifies and/or produces some innovation. The method department then seeks to disperse solutions to the software producing units within the organization through different activities such as supplying scientific and technical knowledge, provision of capital for trials and development of prototypes or pilot studies, and general support for adoption. The aim is to encourage organizational actors to acquire an innovation as a problem-solver or technological fix, as it is advertised and represented through the organizational distribution channels—formal liaison channels and peer networks (Rogers 1983).

The second force takes its departure locally and it focuses on the software development process. The process has certain needs, which creates a pull for innovations (scientific and technical knowledge, technologies, skills, competence, methods, etc.). This triggers the demand for innovations. The human actors involved in the process articulate the demand. Furthermore, the method department will choose to work on potential adopters' perceived problems to increase the likelihood of adoption, which creates a dyadic relationship between the two. At the same time, software producing units will only adopt an innovation insofar that the pros outweigh the cons.

The two forces both offer plausible paths to innovation and adoption, but none of them alone are necessary and sufficient to ensure innovation and adoption of new technologies and practices and all successful adoptions so far have been complex interplays of both drivers. Therefore, any strategy aimed to promote innovative activity or new practices must address both forces. How management may change interactions to alter the cause of an SPI project is addressed next.

A Framework for Strategies

Management is prominently a powerful entity in most SPI projects. Although the 1980s are known as a period of decentralization and front-line empowerment, there is little doubt that management still plays a critical role in sustaining and fostering environments for innovative activity. Management activity can take several forms where tight control is but one.

How Management Intervenes

Strategies to promote innovation and diffusion can be characterized as either influential or regulatory in nature (Damsgaard and Lyytinen 1996; King et al. 1994). Influence is exerted via education and enculturation processes of individuals (Schein 1992), and by the provision of more resources to those activities deemed right and withholding resources from "inappropriate" activities. In short, influential measures are aimed at changing the behavior of those under the management' sway of power without direct reference to force or exercise of command. Regulatory actions, on the other hand, are expected directly to affect the behaviors of entities under the management's jurisdiction. This can take the form of directives, sanctions, regulations, or physical intervention. Regulative measures allow possibly conflicting decentralized decisions to be compatible without the necessity for either units or individuals to comprehend the whole system (Boyer 1988).

When we combine the two modes of management involvement with the two types of driving forces, we obtain a classification of management measures in four entries as exhibited in Table 1. In each entry are listed a number of activities, which may be used by management to change the direction and speed of SPI projects.

Table 1. Classification of Management Measures to Promote Innovation and Adoption

	Supply push	Demand pull
Influence	Knowledge building Knowledge deployment Subsidy Innovation directive I	Mobilization Knowledge deployment Subsidy II
Regulation	III Innovation directive Knowledge deployment Standard setting	IV Subsidy Innovation directive Standard setting

In general, management can initiate activities to (1) develop new technical knowledge or (2) disperse new knowledge to decentralized actors who might benefit from such knowledge. When departments or individuals defray from becoming involved in product or process innovation due to limited resources, knowledge, skills, or conflict of interest, management can consider (3) subsidizing an involvement in innovation processes. In addition, (4) management can direct its own staff or departments directly controlled by it to develop or use innovations. (5) Opinion can also be a target for management intervention to create a demand-pull for innovations. Finally, (6) standardization can be used as a powerful yet crucial tool to achieve a required level of coordination and interdependence between dispersed organizational units.

These activities are not necessarily mutually exclusive but rather they exhibit conceptual and strategic differences in approaches. The six activities are adopted from King et al. They are located in four entries in Table 1. In the following sections, these activities are reformulated in the context of software process improvement.

Knowledge Building

Knowledge building aims at providing scientific and technical knowledge to develop and sustain innovation and diffusion. The basic form of knowledge building is management funded or subsidized R&D programs that help build the base of knowledge necessary for innovative activity. Such programs can be targeted to create technical and scientific knowledge or solutions ready to be used in the organization. This measure clearly falls in entry I in Table 1.

In an SPI, context knowledge building activities may take place in a method department, but more specifically they may be part of special improvement groups' work or even part of the work in technical subgroups. To give an example, an improvement activity aiming at project planning may involve the main improvement group or a technical subgroup working on estimation techniques.

Knowledge Deployment

The goal in knowledge deployment is to disperse new knowledge to individuals and departments so that they are able to use solutions. This may happen through the provision of documentation, data, or libraries of technical or scientific facts, through education, or through the dissemination of knowledgeable individuals (job rotation or bringing new knowledgeable individuals into the organization).

The most common and obvious form of knowledge deployment is to educate the work force (potential adopters) so they can readily apply new technologies or processes (entry II). Another form is to hire knowledgeable individuals to assist the organization with some innovative activity. There are even examples of organizations that have purchased another organization to take possession of valuable knowledge or skills. This is especially true for organizations that wish to acquire a new critical technology or wish to expand their abilities within some specific area (entry I).

It is also common for management to undertake initiatives to educate the work force in general (entry I). The work force is then relieved of their normal duties and instead they attend some education program (e.g., a software academy). Yet another common form is to mandate a certain education level as a requirement of entry for all individuals that wish to work for the organization (entry III).

In SPI, such activities are commonplace. SPI projects may also choose to recruit personnel from "user" departments to form technical groups suggesting improvements within specific areas. In this way, the future procedures and techniques are developed by the very practitioners who will use them afterward. Hence participants in SPI initiatives will often be the first to use new procedures and techniques and act as superusers, promoters, and technology carriers.

Subsidy

The management may subsidize activities that are critical for the diffusion of an innovation. These can be targeted to any or all groups of actors that are involved in the diffusion process. Through subsidies, management can lower the knowledge barriers (Attewell 1992) and they can relieve the burden of some parties from carrying the otherwise unbearable costs of their involvement. This also makes the decision to adopt innovations more lucrative at the early stages of diffusion, when risks are higher, due to scarcity of experience and skills.

Subsidy for the development of prototypes or proof of concepts are both obvious forms of this kind of activity (entry I). Another is to purchase products or services that are envisioned to produce innovative activity (entry II).

Subsidy may also be targeted in a less direct but more radical form. Here the management mandates some innovation to be used at all times to sponsor some activity (entry IV).

In SPI projects pilot, projects or early adopter projects using new technologies may be subsidized via internal consultants, via allocations of extra time or other resources, or via incentive schemes (wages, bonuses, promotions, citations, prestige, travel).

Innovation Directive

Norms that regulate the production or use of innovations in the organization are here called innovation directives. A common way of using innovation directives is to command organizational units to produce, use, or facilitate production and/or adoption of some innovation. One form is to direct the method department to develop a new innovation—e.g., procedures for using a case tool, a software development method, or a test environment (entry I). Another form of using this tactic is to require decentralized units to meet certain requirements or goals that mandate innovative activity (entry III). On the demand side, management imposes organizational units to use certain products or processes whenever possible (entry IV).

In an SPI project, management could mandate that, for example, configuration management procedures and tools be developed and used throughout software development and maintenance.

Mobilization

Mobilization seeks to articulate a particular point of view in relation to innovation that is in accordance with larger organizational goals. This is achieved by making organizational actors perceive an innovation in the "right" way (entry II). Through mobilization of bias, management can, for example, make literature available through publicly accessible databases, intranets, or through organizing workshops and seminars.

Management can also favor a certain point of view by simply paying attention to it and thereby increasing the "demand pull." This can be done through edifying organizational visions around future technological solutions, giving favorable interpretations of the possibilities of an innovation, and so forth.

SPI projects may be pronounced strategic by top management. Achieving a higher maturity level may be motivated by concerns for competitiveness, for positioning the organization technologically at the leading edge, for gaining access to important market segments, or the like.

Standard Setting

Standard setting formalizes practices and limits the space of choice options for organizational actors. Standard setting falls in our classification in the regulation entries since all involved parties have to accept the contract. Setting standards can be targeted both at the demand side (entry IV) and the supply side (entry III). On the demand side, imposing one standard way of doing things creates organization-wide coherence. This triggers learning and also means that organizational actors can readily be used throughout the organization without adjustment or reeducation (entry IV). On the other hand, too rigid standards or standards just for the sake of organizational coherence may prove counter productive, especially if the standard is set against existing practices, culture, or traditions.

On the supply side setting a standard helps the method department focus their resources on support for the standard, which makes their efforts more effective. However in the same vein as the demand side, imposing a standard against existing practices or with too narrow a scope may prove counterproductive (entry III).

In an SPI context, standards may originate from outside of the organization or from the inside. External standards may be used to ensure that procedures and techniques comply with customer demands and/or to set an absolute goal for the SPI project. Internal standards may evolve from best practices in the organization to become organization-wide standards. The enforcement of internal standards may vary depending on specific project conditions.

Field Study

The field study presented in this paper is part of the PF-project (ProcesForbedrings-projektet) at the Center for Information Technology (CIT) in Denmark. The project is a collaboration between software organizations and researchers aimed at applying and further developing ideas related to software process improvement within a Danish industrial context. Systematic activities have been initiated in each of the four participating software organizations. These activities seek to improve software processes based on assessments of the existing capabilities combined with applications of state-of-the-art techniques to professional software development. The collaboration focuses on providing insights into the practical problems and opportunities related to software process improvement. It is not part of the objective to provide an overall picture of Danish software processes and initiatives aimed at their improvement. Started in 1997, the project will run for three years with participants from Danish software-developing organizations, from DELTA Software Engineering—a Danish self-governing research foundation—and from Aalborg University (AAU) and the Technical University of Denmark (DTU). At each organization an action research group was formed consisting of researchers from DELTA and AAU/DTU, and developers/managers from the organization.

Our field study is based on action research. As such we were both engaged in directly helping the companies involved in SPI projects and simultaneously acting as analytical academics. The field study, therefore, was not a well-planned exercise, but instead involved participation and learning, personal involvement, and frequent informal interaction. The field study data was consequently comprised by personal notes taken during meetings, formal minutes of the meetings between the company representatives and the researchers, internal meetings in the companies, open-ended discussions, e-mail correspondence, and other material (such as annual reports and internal manuals and technical reports).

In this section, we present and analyze three cases of SPI initiatives. Two of the cases originate within the same organization whereas the third case is a separate organization. We describe each case and classify all management interventions using the framework developed earlier in this paper. We have chosen not to reveal the true identity of the companies.

Case 1: Comintus Corporation/Phase 1

Comintus Corporation is a large, multinational enterprise specializing in communication systems. A staff of more than 1,000 employees makes the Danish branch large by local standards. The branch is composed of divisions and software development projects within each division are often engaged in projects across several countries.

The Comintus Corporation headquarters requires that all branches participate in substantial SPI activities and specifies a number of minimum goals for the software processes at every branch to be achieved within a given time frame. The Danish branch was engaged in several SPI activities, most of which originated from corporate headquarters. Every year, the corporate level defines new objectives and pronounces new areas of special action.

In the mid-1990s, Comintus Corporation Denmark was engaged in three SPI programs running in parallel. The Primus program aimed at improving CMM maturity in the branch heading for CMM level 2, while the Secundus program with an overall focus on quality and cycle time was addressing a number of special action areas defined at the corporate level. Finally, the Tertius program aimed at consolidating the effects of ongoing process improvement efforts and identifying, planning, and preparing future improvement areas. The third program had several external researchers—among them one of the authors—as participants in a major action research project on software process improvement. The objectives for these three SPI programs were uncoordinated and not prioritized.

The approach used is very broad. The maturity program addressed at least six key process areas simultaneously, and six new special action areas are addressed at the corporate level. Together the current special action areas and those under consolidation amount to some 10 to 15 special action areas. These improvement efforts targeted all projects in every business area. On top of this, corporate initiatives take place in some of the largest projects in the branch. These projects are cooperative, involving branches in several countries in and outside Europe.

The improvement areas address multiple objectives: better quality, more predictable cycle time, reduced cycle time, and CMM level 2. A number of more general objectives supplemented these objectives: profitability, customer satisfaction, employee satisfaction, and corporate image.

The high number of objectives for the improvement efforts indicates that there was no unifying vision at the management level. Some of the improvement programs and efforts were initiated by corporate headquarters, some by branch management, and some by corporate specialist functions. Few efforts are made to prioritize or conciliate objectives, and the status of the objectives is weakened by a firm management attention to products at the expense of processes—*by the end of the day, running code is king*. As a consequence, a large part of the branch at most management levels devoted little attention to the improvement effort of the day.

The Danish branch is a matrix organization with the method group as a staff unit. This specialist unit supports all SPI activities in the branch. In order to build up knowledge about CMM, members of the unit have participated in numerous CMM seminars and courses in and outside the corporation. Furthermore, a new employee formerly employed in another corporation at CMM level 4 has been hired to join the group in the hope that Comintus Corporation Denmark can benefit from his experience.

The unit has little formal power and relies on support from division and project managers in its SPI efforts. The division and project managers are the real process owners of the branch. Project managers and developers did not take directly part in the improvement efforts.

The SPI organization is relatively weak. Most participants engage in more than one SPI activity. Members of the SPI group are often transferred to other tasks and new members are introduced. Roles, authority, and responsibilities are not clearly allocated. The matrix organization leads to massive time expenditures for meetings and information dissemination. At the same time, the matrix organization results in ambiguities with respect to responsibility and authority.

The members of the SPI group primarily define operating procedures and documentation. Little time is assigned to negotiate procedures and support implementation of changes in procedures with project staff. The lack of project staff taking part in defining specific improvements causes problems with respect to selecting appropriate actions as well as with the internal marketing of improvement solutions to project managers and developers.

Branch management decided to delegate responsibility for the SPI activities, but responsibility was not followed by authority. Division managers retained control over the software processes used in their respective projects to the extent that control was not held by international project sponsors. The scope of the SPI efforts was underestimated and no strategic action plan for the coordinated SPI efforts was ever made. Management showed little interest in the SPI programs and rarely asked for progress reports. Branch management did not motivate division managers to take action on SPI initiatives. Project members in the divisions were not informed about the importance of achieving the established SPI goals.

Most of the SPI efforts took place in the method group. The plan was to disseminate approved processes to the practitioners when the processes were ready and the projects needed them. This general approach resulted in a sense of expert domination, alienation, and disregard in the projects as their specific problems were met with ambitious and general solutions.

The corporation pursued an aggressive strategy where new improvement initiatives were launched before present improvements had been implemented; changes were not followed by stabilization. Furthermore, the corporation had numerous initiatives under way simultaneously. Different parts of the corporation drove these initiatives and perhaps as a consequence they were not integrated into a comprehensive improvement plan.

No incentive structure supported the efforts, and the developers and project managers were presented with contradictory success criteria for the software processes and the outcomes of the projects. Moreover, the branch management did little to monitor improvements or to take action when progress was too little or too slow.

In the fall of 1997, the branch management realized that they would likely fail to meet corporate requirements of achieving CMM level 2 at the turn of the year. The branch management decided to take stock of the situation and see if it could launch a simpler and stronger improvement initiative. For this purpose an assessment was scheduled, and this assessment resulted in marked changes in the improvement efforts as described in Case 2.

Institutional Analysis

Knowledge building was not a salient activity in this case, and most knowledge building took place within an expert group detached from practice. The corporation as well as the Danish branch possessed ample knowledge to achieve process improvement. The problem was to get this knowledge transferred from the method group to development projects. Two knowledge deployment activities were used for this purpose: One was to educate a number of employees—mainly from the methods group—in process improvement objectives and assessment activities with a strong focus on CMM (entry II in Table 1). The other was to assign a newly employed staff member with CMM level 4 experience from another corporation to the Primus program (entry I).

Subsidization was not used much in this case, whereas innovation directives played a major role. From corporate headquarters, the branch received numerous and clear directives for process improvements defining improvement areas and goals (entry III). The pressure from headquarters to reach CMM level 2 made the branch search for ways to achieve the required result (entry IV). The realization that the branch would not likely get to level 2 via the activities described forced branch management to change strategy. We describe this change in strategy in Case 2.

Mobilization activities were not used substantially (entry II). Branch management was concerned that product quality was not satisfactory due to a major intake of new employees, but it was not made clear to project managers and developers how product quality was related to process quality and how process improvements were vitally important to the branch. As a consequence, projects would generally focus attention on project deliverables and resources rather than on process improvements.

Standardization was a major concern. Many initiatives coexisted, each addressing standardization issues within a specific area (entry IV). At times these areas were separate, but at other times the same area was addressed by several initiatives. The many initiatives led to a lack of focus and at the same time some of the efforts, aimed at corporation- or branch-wide standardization, were at odds with existing practices (entry III).

Case 2: Comintus Corporation/Phase 2

Partly due to management concern, partly to pressure from the Tertius program, an assessment was made in order to evaluate the effectiveness of the process improvement programs and more specifically to appraise if Comintus Corporation Denmark would be at CMM level 2 at the planned date. Two consultants from Implicator Consulting (a subsidiary of Comintus Corporation) performed the assessment. The consultants spent three days interviewing management, project managers, and developers. Key people from the process improvement programs were interviewed as well.

The assessment showed that the Danish branch would not reach the goal unless substantial changes were made to the improvement programs. It was concluded that goals had not been prioritized, that too many different activities had been directed toward the same goals, and that there had been too little focus on implementation concerns. In addition, the consultants found that management was not visibly committed to the improvement efforts.

The assessment resulted in a decision to give the Primus program top priority for a period of six months. Aiming at CMM level 2, the effects of this top priority for the Primus program would become apparent in a formal CMM assessment ending this period. During the period, much attention had to be focused on the program and its progress from top management downward.

Following the assessment, the branch was partly reorganized from a matrix organization into a line organization. The business areas were made directly responsible for obtaining improvements and process responsibility was decentralized.

Implementation was stimulated by the creation of local CMM task forces within all product areas. These task forces were supported by a requirement that all product areas should monitor implementation progress every week by doing small CMM assessments based on questionnaires. The results of these assessments were reported to the CMM program and management would be notified about progress.

It is as yet unknown if the branch will pass on to CMM level 2, but the reorganization and focused implementation efforts seem to be a success, at least in the short run:

- During the last few months, considerable progress was achieved and the branch may pass a full-scale CMM assessment shortly. On the other hand, the immense management pressure applied to get the branch this far has stressed the branch to the extent that it may regress below level 2 again soon after the assessment.
- A culture of measuring seems to be established via the frequent small assessments. It is interesting to see if this culture is sustainable or becomes even more developed in the future, or if measurements tend to wear out after the assessment.
- Using small assessments meant that the objectives of the Primus program became clear to project managers and developers, and if the improvement proves to be a lasting one, this experience may support implementing further improvements in the future.

Institutional Analysis

Case 2 is a sequel to Case 1 and we therefore concentrate on differences between these two cases. The knowledge deployment activities were maintained and strengthened (entry I) mainly by directing all development projects to conduct small CMM assessments every week (innovation directive, entry IV). This made employees in the line organization learn about CMM and relate CMM requirements to their own work.

Corporate level innovation directives were suspended (entry I). Branch management decided that the Primus program should be the sole improvement effort for six months. This meant that the Secundus program was put on hold and the Tertius program was redefined to act as support and reference for the Primus program. Small CMM assessments in all projects forced the projects to identify problem areas in their development process and to demand process improvements for such areas (entry IV).

Mobilization was a key activity in the transformation from Case 1 to Case 2. The results were a massive focus on CMM level 2 at the levels of development project and business area was supported by the establishment of local CMM task forces, the transformation from a matrix organization toward a line organization, and the business managers becoming sponsors of the improvement efforts (entry II).

Standard setting was decentralized when the improvement activities were in effect moved from branch level to the level of business area. The result of this was that standardization came closer to the existing practices (entry III). At the same time, standard setting was strengthened by the reduction in improvement areas and a stronger focus on the remaining improvement activities (entry IV).

Case 3: Methodicus Corporation

Methodicus Corporation is a Danish software-developing organization founded in the 1980s. The company delivers sophisticated technical systems solutions, products, and support to the public and private sector in Denmark and abroad. The company has four offices worldwide with a total of around 125 employees; the main department has 100 employees. Only the main department develops and maintains software solutions, whereas the other three do sales and marketing. A small quality function headed by a quality manager was responsible for quality assurance and method development. Most of the employees are well educated. Around half of them hold a Master's degree in computer science or computer engineering, one fifth has another university degree, and a quarter has a non-university education in computer science. The staff is generally young as most of the employees started their career in Methodicus Corporation.

In early 1997, Methodicus Corporation established the Altius project—a software process improvement project—in cooperation with external researchers—among them one of the authors—as participants in the same major action research project on software process improvement mentioned in Case 1. The goal of the Altius project was to bring Methodicus Corporation from CMM level 1 to level 3.

Methodicus Corporation initiated the Altius project through a meeting where all members in the organization were present together with senior management and the external researchers. The researchers presented the project, its objectives and plans, followed by senior management expressing commitment to participate. Senior management presented the overall vision for Methodicus Corporation, and presented strategies for how the Altius project could contribute to fulfill the vision. The "kick-off" meeting was held in an informal atmosphere, where the employees were encouraged to ask questions and raise issues regarding this three-year initiative. There was no doubt, however, that this was what senior management wanted, and that no alternatives existed.

A project group was established. Initially a project manager assigned half his time to this project and one full-time member staffed it. The project manager was chosen because he was well respected from an earlier important and successful improvement project. The group member had no previous experience with process improvement. She was hired in as ordinary software developer and became a member of the group partly due to an interest and partly due to her availability when the Altius project was initiated. After a few months, she became a part-time member and later left the group. Another employee replaced her. The group changed three times during the first year. For example, a new quality manager was appointed and invited to join the group after five months.

Two separate assessments lead by the researcher team identified the areas of improvement. The results from these assessments were in good accordance and five technical groups were established. Each group's brief was to address the findings related to one or two CMM level 2 key process areas. Four out of the six identified areas of improvement had high priority, and the two remaining areas were nominated to have low priority. This reflected the desire by management to improve all six areas of CMM level 2 simultaneously—a desire fueled by management hopes to achieve level 2 capability via organization level processes in t one fell swoop. Nonetheless, this allocation of priority to most areas bears resemblance to no priorities at all.

Participants from each department in Methodicus Corporation staffed the technical groups with one person from the Altius project group in order to coordinate the work and to ensure consistency and coordination. The members of the groups were mainly volunteers, but in a few cases, the Altius project manager found it necessary to “convince” a key person to volunteer for a specific group, e.g., the four business managers were found necessary for the project planning and tracking improvement area. The five groups together comprise about one third of the organization’s entire development staff, all of them except one or two being volunteers. This indicates a very high commitment and motivation from managers and developers toward the Altius project.

A strategic action plan guided the overall effort. The aim of this plan was to mandate and guide the improvement work to be performed by the technical groups. The plan was ready five months after project start and defined the expected results of the technical groups.

All groups were required to make a technical report describing a conceptual basis for work within their field and to outline a plan for implementation. In general, adoption was expected to take place after forming groups responsible for implementation within each of the five fields. Consisting of practitioners and one person from the quality function, these groups met regularly during implementation. When the technical reports were finished, this requirement was not met. Instead responsibility for implementation was generally awarded to the quality function and the technical groups were only supposed to meet again three to six months after implementation had started to monitor progress. The quality function was not mandated with formal authority or resources to push implementation; rather it seems to be anticipated that expressed management support to the conceptual bases would suffice.

A very problematic area was allocating resources to the Altius project. Several times during the project, allocated resources were withdrawn in order to fulfill an immediate business need, e.g., the full-time member of the Altius project group was allocated to another project in order to do ordinary systems development. Other time-consuming responsibilities also involved the members of the group. This problem was also seen in the technical groups, where the members were expected to do improvement work without being relieved from their ordinary responsibilities as systems developers. The work in the technical groups thus depended on how busy the members were in carrying out their ordinary responsibilities. Most groups found it difficult to find the time to do improvement work in the technical groups and consequently they were all delayed compared to the plan. One group member noted:

> *Most people know that the lack of results is not due to a lack of commitment...but due to a lack of resources....Our problem...is resource allocation, it is difficult to keep good resources.*

When the reports from the technical groups were ready, senior management decided that two highly respected and competent people should be allocated full-time to continue the implementation. After one month, these two persons were withdrawn as full-time Altius project members and reallocated to other pressing business needs where their expertise was crucial.

Allocation of resources for implementation played an inferior role in the plans and certainly in management decisions. Implementation generally appeared as a decision made by management. The plans contained few deliberations on how to achieve the changes. The improvements were mainly perceived as a result of deciding appropriate technical changes. A remarkable example of this is found in the report on configuration management: here implementation was primarily seen as a matter of purchasing and installing a software tool combined with management decisions. The technical group was not necessary for implementation and this probably explains why this particular group was terminated.

Despite the Altius project being pronounced as strategic to Methodicus Corporation, the reality was that the project was an internal project with little priority compared to external projects. When important customers required that their projects be staffed with key personnel, this implied a reallocation of staff intended for the Altius project. Thus the vital technical group on Project Management starved when suddenly key members were withdrawn and assigned full time to a different project.

Status for the Altius project after completion of the first year: All technical groups had produced plans for improvements within their respective fields. Senior management had reviewed these plans and indicated which conceptual bases should be implemented. The management expected to allocate additional resources for implementation activities within a few of the areas. It was not yet decided and announced who would be responsible for the individual areas, and the technical groups faded away.

Institutional Analysis

Methodicus Corporation knew little about CMM prior to this project and consequently had to engage in knowledge building to ameliorate this (entry I). An expert strategy was used here, with some assistance from academics. Only members of the Altius project group would build general knowledge about CMM primarily by reading publications on CMM. The Altius project manager visited a U.S. company with extensive CMM experience.

Knowledge deployment was mainly achieved by having members of the Altius group cooperate with employees with expertise within specific areas. Together they studied CMM requirements and aimed to transform these into local software development processes (entry II).

Subsidies were not used as a strategy. To the contrary, it was expected that process improvement activities would be performed without compensation. Furthermore in most cases employees working with process improvements were mandated to participate in development projects as usual.

Innovation directive was one of the most prominent strategies pursued in the case of Methodicus Corporation. Senior management directed the Altius project to develop a software process compliant to CMM level 2 based on existing process descriptions from the quality system in the corporation (entry I). With this in place, senior management required all organizational units to follow this process (entry IV). Once CMM level 2 was achieved, the same improvement strategy would be pursued for CMM level 3.

Another prominent strategy in this case is mobilization. Senior management felt confident that it would be possible—indeed easy—to build a commitment throughout the organization toward helping devise and utilize the new software process (entry II).

In line with this, senior management was convinced that the new process could easily be pronounced an organizational standard to be used in all projects with minor customization (entry IV). Being a small organization, it would be possible to establish an organization-wide standard, which at the same time would support existing practice (entry III).

Discussion

In this section, we assess the six categories of management intervention in terms of their likely successful impact on the SPI projects. We accomplish this first by condensing the three cases in terms of how the significance of the six measures was perceived and the reported experiences of their use. Following this we present a few learning points that can be drawn from this study.

Condensation of Measures

Table 2 summarizes the case studies. From this table we can see that, although there are differences between the cases, important patterns in the use of management measures seem to emerge.

Based on this condensation, the three cases illustrate these uses of management measures:

Knowledge building was used in all three cases. Nevertheless it seems that knowledge building is not a key focus area for the involved organizations. Knowledge about SPI is quite well established and readily available. Instead the major challenge was dispersing SPI knowledge and putting process innovations into use.

Knowledge deployment was addressed in all three cases, but not in a coordinated and well-prepared manner. Knowledge deployment activities were aimed at teaching potential adopters about SPI innovations and how to use them. There was little or no attention on how to perform or evaluate these knowledge transfer activities.

Subsidy was not applied at all to achieve the desired effects on the supply side or on the demand side. Instead it seems that management expected both the method units and adopters to embrace and accommodate SPI changes without allocating more resources. The incentive structures of the two organizations were not changed or altered because of SPI projects.

Innovation directives were used in a lavish way. SPI projects were initiated and mandated by management either at a local level or at a distant corporate level. This seems to be the most popular form of controlling SPI projects. However, the experiences from the three cases here leave little doubt that directing SPI projects in this way is not sufficient to achieve the desired improvements.

Mobilization was applied as a management tool in the two latter cases. In the second case, mobilization came about when responsibility for the SPI project was delegated to local management. In the Methodicus case, mobilization was instrumental in the initial phase of the SPI project. However the improvement efforts seem to wear out when mobilization is not followed by other measures.

Table 2. An Overview of the Three Cases
(“m.n.a.” equals mechanism not applied)

Dimension of intervention	**Comintus Corporation (phase 1)**	**Comintus Corporation (phase 2)**	**Methodicus Corporation**
Supply push x Influence			
Knowledge building	Dedicated expert group assisted by academics provide SPI knowledge	Local responsibility for SPI innovation	Expert group of practitioners assembled to create SPI knowledge. Assistance from academics.
Knowledge deployment	Hiring knowledgeable individual. Academics bring in knowledge.	No change	Academics bring in knowledge.
Subsidy	m.n.a.	m.n.a	m.n.a.
Innovation directive	m.n.a.	m.n.a	m.n.a
Demand pull x influence			
Mobilization	m.n.a.	Managers sponsor and promote SPI project	Key activity to create attention and shared vision
Knowledge deployment	Education of potential adopters	No change	Using practitioners in knowledge building
Subsidy	m.n.a.	m.n.a.	m.n.a.
Supply push x regulation			
Innovation directive	Directives for areas and goals of SPI initiatives dictated by corporate HQ	Abandoned in phase two	Dictated to move organization to CMM level 2
Standard setting	Centralized standardization efforts (not coordinated)	Decentralized local standards	Dictate SPI efforts to be launched
Demand pull x regulation			
Innovation directive	Mandated use of SPI	Mandated use of mini assessments	Mandated use of SPI innovations in all future projects
Standard setting	Centralized standardization efforts (not coordinated)	Decentralized local standards	Dictate one “right” way to follow for projects.

Standard setting is at the core of SPI projects, both on the provision side and the adopters' side. In the Comintus Phase 1 case, uncoordinated SPI efforts resulted in confusion and wasted resources in the provision of SPI innovations. In the Comintus Phase 2 case, fewer and more focused innovations and standards at the business level led to substantial progress on the SPI project. In the Methodicus case, the relatively small size of the organization and the composition of the technical groups ensured a close relationship between SPI innovations and practice.

What Management Tends to Forget

Based on our analysis, we now formulate a number of learning points that management engaged in SPI improvements should try to keep in mind to avoid some of the pitfalls we have identified.

1. Management tends to forget that adequate resources must follow SPI initiatives. Changing and innovating existing software development procedures is risky and resource demanding as well as time consuming. It is therefore imperative that management establishes sufficient organizational slack for SPI activities to be nurtured and protected against the sharp jaws of tradition and tight budgets.
2. Management tends to forget that informing and educating software developers in new development and maintenance procedures is not equal to changing software development processes and indeed practices.
3. Management tends to forget that forcing software development projects to change by issuing directives may not yield the desired effect.
4. Management tends to forget that incentive structures are one of the most basic and well-established forms of changing the behavior of organizational actors. As long as management shares the view that *by the end of the day, running code is* ***king*** and allows "normal" software development projects to take priority over SPI initiatives, they will not catch on.
5. Management tends to forget that software development processes are complex interactions between people, technology, and software requirements. Such processes cannot be "matured" in one drastic sweep (i.e., attempt to mature all key process areas simultaneously). We therefore call for a less radical and instead an evolutionary approach to SPI.

Conclusions

In this paper, we have proposed an institutional framework that differentiates alternative mechanisms for management involvement in SPI projects. Six mechanisms were presented and three cases were classified using the model.

Several conclusions can be drawn from the analysis. First, management is a significant player in the diffusion process. Its level of involvement and actions varies with organizational policy and knowledge about SPI projects.

Second, management of SPI projects seems to be concentrated on standard setting and issuing innovation directives and to some degree on mobilization. From our analysis, it is clear that there is plenty of room for further management involvement and that there are many possibilities for management to engage in SPI projects still left open or unattended. One obvious tactic that is not widely applied is subsidies. We believe that without allocating sufficient resources to SPI projects they will have little opportunity to be successful. Another form of subsidy could be to modify the rewarding and incentive structure of the companies to accommodate SPI experience or knowledge.

Finally, it appears that even though SPI knowledge is well established and available there is still much work to be done before SPI innovations can be readily applied in an organizational context.

Clearly further research is needed in this area if a validated theory of the role of management in SPI projects is to be established. Such a theory would give a delicate vocabulary to describe and analyze SPI processes and help management to choose among a family of mechanisms handy for management to promote SPI projects. Longitudinal empirical studies that seek to examine dependencies between the type of management action and the speed and direction of SPI projects form one necessary step in this direction.

Acknowledgments

Thanks to the participating organizations. This research was in part funded by CIT—the Danish National Centre for IT Research.

References

Attewell, P. "Technology Diffusion and Organizational Learning: The Case of Business Computing," *Organization Science* (3:1), 1992, pp. 1-19.

Boyer, R. "Technical Change and the Role of 'Regulation'," in *Technical Change and Economic Theory*, G. Dosi, C. Freeman, R. Nelson, and L. Soete (eds.), Pinter Publishers, London, 1988, pp. 67-94.

Brooks, F. P. "No Silver Bullet: Essence and Accidents of Software Engineering," *Computer* (20:4), 1987, pp. 10-19.

Damsgaard, J. *The Diffusion of Electronic Data Interchange: An Institutional and Organizational Analysis of Alternative Diffusion Patterns*, Ph.D. Thesis, R-96-2041, Department of Computer Science**,** Aalborg University, Denmark, 1996.

Damsgaard, J., and Lyytinen, K. "Government Strategies to Promote the Diffusion of Electronic Data Interchange (EDI): What We Know and What We Don't Know," *Information Infrastructure and Policy* (5:3), 1996, pp. 169-190.

Damsgaard, J.; Rogaczewski, A.; and Lyytinen, K. "How Information Technologies Penetrate Organizations: An Analysis of Four Alternative Models," in *Proceedings of the IFIP TC8 Working Conference on Diffusion, Transfer and Implementation of IT*, L. Levine (ed.), North Holland, Amsterdam,1994, pp. 1-21.

Damsgaard, J., and Scheepers, R. "Power, Influence and Intranet Implementation: A Safari of South African Organizations," submitted for publication, 1998.

Diaz, M., and Sligo, J. "How Software Process Improvement Helped Motorola," *IEEE Software* (14:5), 1997, pp. 75-81.

Haley, T. J. “Raytheon’s Experience in Software Process Improvement,” *IEEE Software* (13:6), 1996, pp. 33-41.

Herbsleb, J.; Zubrow, D.; Goldenson, D.; Hayes, W.; and Paulk, M. “Software Quality and the Capability Maturity Model,” *Communications of the ACM* (40:6), 1997, pp. 30-40.

Hollenbach, C.; Young, R.; Pflugrad, A.; and Smith, D. “Combining Quality and Software Improvement,” *Communications of the ACM* (40:6), 6, 1997, pp. 41-45.

Humphrey, W. S. *Managing the Software Process*, Addison-Wesley Publishing Company, Reading, MA, 1989.

Humphrey, W. S.; Snyder, T. R.; and Willis, R. R. “Software Process Improvement at Hughes Aircraft,” *IEEE Software* (8:4), 1991, pp. 11-23.

King, J. L.; Gurbaxani, V.; Kraemer, K. L.; McFarlan, F. W.; Raman, K. S.; and Yap, C. S. “Institutional Factors in Information Technology Innovation,” *Information Systems Research* (5:2), 1994, 139-169.

Markus, M.L. “Power, Politics, and MIS Implementation,” *Communications of the ACM* (20:6), 1983, pp. 430-444.

Minztberg, H. *Structure in Fives: Designing Effective Organizations*, international ed., Prentice-Hall, London, 1983.

Paulk, M. C. “Effective CMM-Based Process Improvement,” *Proceedings of the Sixth International Conference on Software Quality*, Ottawa, Canada, 1996, pp. 226-237.

Paulk, M. C.; Curtis, B.; Chrissis, M. B.; and Weber, C. V. *Capability Maturity Model for Software, Version 1.1*, CMU/SEI-93-TR-24, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1993.

Paulk, M. C.; Weber, C. V.; Garcia, S. M.; Chrissis, M.; and Bush, M. *Key Practices of the Capability Maturity Model, Version 1.1*, CMU/SEI-93-TR-25, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA., 1993.

Perrow, C. *Complex Organizations—A Critical Essay*, 3rd ed., McGraw-Hill, New York, 1986.

Porter, M. E. *Competitive Advantage: Creating and Sustaining Superior Performance*, The Free Press, New York, 1985.

Pressman, R. S. *Software Engineering: A Practitioner’s Approach*, 4th ed., McGraw-Hill, New York, 1997.

Rogers, E. M. *Diffusion of Innovations*, 3rd ed., The Free Press, New York, 1983.

Schein, E. H. *Organizational Culture and Leadership*, Jossey-Bass Publishers, San Francisco, 1992.

Software Engineering Measurement and Analysis Team. *Process Maturity Profile of the Software Community 1997 Update*, SEMA 5.98, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1998.

Software Productivity Consortium Services Corporation. *Managing Process Improvement: A Guidebook for Implementing Change*, (SPC-93105-CMC). Herndon, VA, 1993.

Tornatzky, L. G., and Klein, K. J. “Innovation Characteristics and Adoption-Implementation,” *IEEE Transactions on Engineering Management* (EM-29:1), 1982, pp. 28-45.

About the Authors

Jan Damsgaard is assistant professor at Aalborg University. He has a Ph.D. and a European Doctorate degree in Computer Science (Aalborg University, 1996) and he also holds a Master's Degree with a major in Computer Science and a minor in Psychology from Aalborg University. His research interests include information systems, diffusion of information technologies, and in particular diffusion of complex, standard-based and networked technologies. E-mail: damse@cs.auc.dk

Ivan Aaen is associate professor at Aalborg University. He has a Ph.D. degree in Computer Science (Aalborg University, 1990) and is a Master of Science in Engineering. His research interests include information systems, software engineering, and in particular software process improvement. E-mail: ivan@cs.auc.dk