Between control and drift: negotiating improvement in a small software firm

Gitte Tjornehoj
Department of Computer Science, Aalborg University, Denmark, and
Lars Mathiassen
Computer Information Systems, J. Mack Robinson College, Georgia State University, Atlanta, Georgia, USA

Received 27 September 2006
Revised 21 March 2007
17 October 2007
Accepted 26 October 2007

Abstract

Purpose – While the literature on software process improvement (SPI) offers a number of studies of small software firms, little is known about how such initiatives evolve over time. On this backdrop, this paper aims to investigate how adoption of SPI technology was shaped over a ten year period (1996-2005) in a small Danish software firm.

Design/methodology/approach – The investigation is based on a longitudinal, interpretative case study of improvement efforts over a ten-year period. To help structure the investigation, we focus on encounters that impacted engineering, management, and improvement practices within the firm. The study contributes to the SPI-literature and the literature on organizational adoption of technology.

Findings – The paper finds the improvement effort fluctuating and shaped between management's attempt to control SPI technology adoption and events that caused the process to drift in unpredictable directions.

Practical implications – The experiences suggest that managers of small software firms remain flexible and constantly negotiate technology adoption practices between control and drift, creating momentum and direction according to firm goals through attempts to control, while at the same time exploring backtalk, options, and innovations from drifting forces inside and outside the firm.

Originality/value – Based on the research, the paper recommends substituting the "from control to drift" perspective on organizational adoption of complex technologies like SPI with a "negotiating control and drift" perspective.

Keywords Technology led strategy, Computer software, Control systems, Denmark

Paper type Research paper

Introduction

In his latest work, Claudio Ciborra addressed modernity and the consequences for managing technology in industry (Ciborra *et al.*, 2000; Ciborra, 2002). He argued that traditional thinking of successful technology adoption as planned and managed does not suffice in the modern, global world. On the contrary, technology drifts away from plans just as fast as new plans are made. What technology ends up being is a result of events in a complex, unpredictable, and unmanageable web that is shaped by bricolage

The authors dedicate this paper to Claudio Ciborra (1951-2005) for his contributions to Information Systems theory. Claudio thrived in controversy and he was particularly critical to CMM and much of the SPI literature. In that spirit, the authors have developed Claudio's thinking on technology adoption based on a detailed study of ten years of SPI practices in a small software firm.

Information Technology & People
Vol. 21 No. 1, 2008
pp. 69-90

0959-3845
DOI 10.1108/09593840810860333

(Ciborra, 1994a, Ciborra and Hanseth, 1998, Lévi-Strauss, 1962, 1972) and formative context (Ciborra and Lanzara, 1994). The world is moving, he argued, from being governed through control towards being dominated by side effects, drifting technologies, and increasing unpredictability and risk.

In this context of growing complexity, software process improvement (SPI) has become an increasingly interesting technology for software firms struggling to stay competitive. The Capability Maturity Model (CMM) (Paulk *et al.*, 1993) and the recent CMM Integrated (CMMI Product Development Team, 2002, CMMI product development Team, 2000) have sparked a new discipline of engineering management that plays a dominating role in practice and research (Hansen *et al.*, 2004). The CMMs are based on the ideal of a rational, control-centered culture for software development (Ngwenyama and Nielsen, 2003) and although other SPI approaches have been suggested (Ares *et al.*, 2000, Bennetts and Wood-Harper, 1998; Kehoe and Shah-Jarvis, 1995; Schneider, 2002) they do not differ from the CMMs when it comes to underlying values (Hansen *et al.*, 2004).

In general, SPI research is based on short periods of empirical evidence. While the literature offers a number of successful cases of adopting CMM (Humphrey *et al.*, 1991, Wohlwend and Rosenbaum, 1994, Dion, 1992), reports on failure and difficulties have lately increased (Mathiassen *et al.*, 2002; Ngwenyama and Nielsen, 2003; El-Emam *et al.*, 2001; Rodenbach *et al.*, 2000; Börjesson and Mathiassen, 2003; Villalon *et al.*, 2002; Börjesson and Mathiassen, 2005; Hansen *et al.*, 2004). Especially within small software firms, adoption of SPI-technology seems problematic since these organizations lack sufficient resources to invest in improvements (Steel, 2004; Brouse and Buys, 1999; Villalon *et al.*, 2002; Brodman and Johnson, 1994) and SPI knowledge is not sufficiently tailored to their needs (Kilpi, 1998; Saastamoinen and Tukiainen, 2004). Small software firms are highly sensitive to dynamic environments (Ward *et al.*, 2001) and dominating approaches to SPI fit poorly (Leung and Yuen, 2001; Varkoi *et al.*, 1999) because it normally takes several complex and expensive initiatives to reach new maturity levels (Aaen *et al.*, 2001).

On this basis, we have adopted Ciborra's framework to investigate the following research question "How can small software firms manage adoption of SPI-technology?" Our focus is hence not on specific maturity models or software processes, but on how adoption of SPI-technology (including maturity models, improvements approaches, principles for organization, and tools) can help small software firms improve engineering practices. The investigation is based on a longitudinal, interpretative case study (Walsham, 1993, Pettigrew, 1990) of improvement efforts over a ten-year period from 1996 to 2005 in a Danish software firm, SmallSoft. To help structure the investigation, we focus on encounters that impacted engineering, management, and improvement practices within the firm (Peterson, 1998; Cho *et al.*, 2006; Newman and Robey, 1992). The study contributes to the SPI-literature and the literature on organizational adoption of technology.

The argument is structured as follows. First, we present relevant theory on SPI in small firms and introduce the concepts of control and drift based on the latest work of Ciborra (Ciborra, 2002; Ciborra *et al.*, 2000). Then, we explain our longitudinal, interpretative case study approach (Walsham, 1993; Pettigrew, 1990) structured around key encounters (Cho *et al.*, 2006; Newman and Robey, 1992; Peterson, 1998). Subsequently, we introduce SmallSoft and provide a detailed account of how

SPI-technology was adopted during the period 1996-2005. Finally, we discuss the case in response to the research question and highlight the contributions and implications of the study.

Theoretical background

Small software firms face special challenges when improving software practices. These challenges relate to the resources available and to having minimal influence over the environment. In the following, we review what we know about SPI in small software firms and we present Ciborra's (2002) distinction between control and drift as a framework for investigating management challenges in this particular context.

Improvement in small software firms

There are a number of studies of small firms trying to adopt CMM. Typically, these studies describe the difficulties that small software firms encounter and how these can be successfully resolved. Several studies suggest adaptations of CMM to small firms ranging from planning and implementing SPI-initiatives pragmatically according to firm needs (Batista and Figueiredo, 2000; Harjumaa *et al.*, 2004; Varkoi *et al.*, 1999; Kilpi, 1998; Scott *et al.*, 2002; Jakobsen, 1998; Kautz *et al.*, 2000; Kautz, 1999; Kelly and Culleton, 1999) to composing new SPI-frameworks tailored for small firms and comprised from CMM key process areas (Leung and Yuen, 2001; Wilkie *et al.*, 2005, Casey and Richardson, 2004; Kautz *et al.*, 2000; Horvat *et al.*, 2000; Ruiz *et al.*, 2002). The main insight offered by this research is that adapting CMM can be beneficial to small firms when done through pragmatic interpretations of available frameworks. Managers are advised to consider in which sequence, with what focus, and on which level of ambition different CMM process areas are applied to assess current practices and implement new ones. In support of such a pragmatic approach, Dybå (2003) found that among 120 Scandinavian companies the smaller ones adopted SPI-technology just as effectively as larger companies. Also, Paulk (1998) argues that small firms' adoption of CMM "may be different in degree, but they are not different in kind" (Paulk, 1998) from those of other organizations. In general, it takes "professional judgment and understanding of how the CMM is structured to be used for different purposes" (Paulk, 1998).

Another group of studies have discarded CMM and developed alternative approaches while still keeping the basic values of the rational software organization intact. Most of these studies report from one or a few cases of successful application and, on that basis, recommend the presented approach to other small software firms. Examples are DSDM and people process (Coleman and Verbruggen, 1998), AMETIST (Thowart, 1999), TAPESTRY (Kuvaja *et al.*, 1999), 3P approach (Brouse and Buys, 1999), IMPACT (Scott *et al.*, 2001), Software Process Matrix (Richardson, 2001, 2002), MESOPYME (Villalon *et al.*, 2002), COSMEA (Grechenig and Zuser, 2004; Steel, 2004), and SPICE (Truffley *et al.*, 2004). While these studies vary considerably in focus, they all advocate a need for adopting SPI-technology and they propose ways to downsize the effort to better match the resources and culture of small software firms.

Managers in small software firms can hence find advice for SPI-technology adoption in the literature. Most of the studies are, however, based on observations over limited time periods, focusing on initial adoption. Only a few studies report on how SPI-initiatives evolve over a considerable time span (Balla *et al.*, 2001; Truffley *et al.*, 2004; Richardson, 2001; Kuvaja *et al.*, 1999; Mustonen-Ollila and Lyytinen, 2003). Moreover, most studies focus on adapting elements of CMM or on developing alternative models that fit the needs of small software firms. In this study, we have therefore pushed models into the background and given center stage to the understanding of how SPI-technology was adopted in a small software firm over a period of ten years. To make sense of this adoption process we focus on the relationship between managerial control and drift caused by events inside and outside the firm.

From control to drift

In the book *From Control to Drift* Claudio Ciborra *et al.* (2000) outline a vicious circle of how firms strive for management control of technology adoption, but instead experience drifting due to forces of turbulent environments, implementation tactics, power of the installed base, complexity of the technology, side-effects, surprises, and users' resistance and creativity. Limits to learning and the pre-existing formative context (Ciborra and Lanzara, 1994) keep firms in this cycle and reinforce the perceived need for control.

Ciborra and his associates focus on corporate information infrastructures as elements in globalization, but they also discuss technology adoption in general (Ciborra *et al.*, 2000). Ciborra (2002) elaborates further and points to a hidden crisis of the Information Systems field. This crisis is caused by the dominating role of rational models in managing organizations and technology. He argues that this view is constraining our understanding of the world and prevents us from seeing other dimensions of technology (Ciborra, 2002).

The rational science view leads us, according to Ciborra (2002), to misunderstand or not see the world as experienced in the everyday life of agents, users, designers, and managers. We introduce and enforce geometrical models trying to make the world fit. But while trying to control and plan technology in this way, it drifts away from plans as side effects (Hanseth *et al.*, 2001) and surprises (Ciborra, 1994b) happen. Humans respond by reinventing technology through improvisations, bricolage (Ciborra, 1994a, Ciborra and Hanseth, 1998, Lévi-Strauss, 1962, 1972), and hacking[1] (Ciborra, 1999). The adoption process is therefore shaped differently than expected through formative contexts (Ciborra and Lanzara, 1994) or the already installed base (Ciborra and Hanseth, 1998). When bounded in their imagination by specific formative contexts (Ciborra and Lanzara, 1994), humans have limited innovative capabilities. However, coincidence, breakdowns, and human coping can spark technology drifting and result in more innovative outcomes:

> Drifting can be looked at as the outcome of two intertwined processes. One is given by the openness of the technology, its plasticity in response to the re-inventions carried out by users and specialists, who gradually learn to discover and exploit features, affordances, and potentials of systems. On the other hand, there is the sheer unfolding of the actors' being in the work flow and the continuous stream of interventions, tinkering, and improvisations that color perceptions of the entire system life cycle (Ciborra, 2002, p. 87).

In this view, usage, maintenance, redevelopment, and improvement take place simultaneously and range from sabotage over passive resistance, to learning and micro inventions, and even radical shifts. As a result, Ciborra argues, we need to change our thinking and practices from control to drift. Such a move will allow firms to support human innovation instead of controlling, and facilitate cultivating and hosting technology instead of planning and designing it. These ideas have recently been addressed and employed in the special issue of *Journal of Information Systems* on "Claudio Ciborra and the IS Field: Legacy and Development" (Brigham and Introna, 2006; Saccol and Reinhard, 2006; Elbanna, 2006).

Although the adoption of SPI-technology in SmallSoft differs from Ciborra and his associates' cases of corporate infrastructures of global companies (2000) in terms of company size, the interconnectedness, unpredictability, and degree of external pressure on the firms are similar. SPI-technology displays characteristics of openness and plasticity and is also promoted as having strategic importance to firms. Hence, the discourse outlined in Ciborra's later work lends itself nicely to making sense of the particular case of SPI-technology adoption.

Research approach

Longitudinal interpretative case study

We have framed the investigation as a longitudinal, interpretative case study (Walsham, 1993, 1995; Pettigrew, 1990) based on a years-long collaboration with SmallSoft through a university-industry network (Mathiassen, 2002). This approach allowed us to analyze in detail how adoption of SPI-technology evolved over time and to link the findings to the theoretical debate over control and drift (Ciborra *et al.*, 2000; Ciborra, 2002).

As suggested in Pettigrew's theory of longitudinal field research on change, the study presents a peek into an ongoing process of change. We base the study on several years of personal interaction with SmallSoft. We complement these insights with access to extensive documentation and interviews, and the study ends in a situation where management has just been through a period of breakdowns, reflections and learning, and has decided to engage in yet another SPI-initiative. This view of ongoing change leads to open-ended interpretations and conclusions. We handle the time aspect of the study by focusing on encounters punctuating relatively stable episodes of evolution (Cho *et al.*, 2006; Peterson, 1998; Newman and Robey, 1992). We have applied the criteria of contextual, processual, historical, and pluralist data to triangulate between personal experiences through involvement, documentary and archive data, observational and ethnographic material, and in-depth interviews. The authors have gained knowledge from direct involvement in the improvement efforts in SmallSoft, at different times and in different roles. These differences help to balance the presentation of the case.

SmallSoft

SmallSoft is a small Danish IT-firm with approximately 50 employees. Their core competence is to combine domain specific engineering knowledge with IT-competencies to serve their customers by developing new solutions or by adapting standard software.

SmallSoft was founded when a large engineering firm closed down in 1987 because of bankruptcy. Three engineers formed a consultancy firm operating in the same industry segment. They soon began to develop dedicated software tools in-house to support consultation, which led them to develop a material-management-tool as a joint venture with a main customer. Even though the virtues of this software product were its built-in domain knowledge, not its technical quality, it evolved into an important standard software product and the development process laid the basis for the software practices of SmallSoft today.

In the early 1990s, software production was established as a separate business area and SmallSoft started to employ additional developers and established a new department for tailored IT-systems. After a couple of years, the IT-business area dominated the firm that had grown to include 42 developers; 25 developers worked in the Standard Department, responsible for the original and still important standard software that over the years had developed into a portfolio of subsystems and versions; 15 developers worked in the Tailored Department, tailoring systems to a variety of customers; finally, a new two-person department developed a storage management product based on a recent acquisition.

Since the three founding engineers had just watched a bankruptcy from the inside, they developed a firm culture based on a low risk attitude and a belief in core engineering skills and long-term customer relations. During the difficult first years, standards for managing, staffing, and building customer relations and solutions were established. Despite growth, SmallSoft's top level management and board continued to exercise a defensive business and investment strategy, expecting surplus every single month. Customer relations were built on long-term personal relations and employees were valued as experts, since most development work demanded detailed domain knowledge. Software engineering skills were generally considered less important. During the mid-nineties, the development of the standard software product was fleshed out from development of customer versions to overcome severe quality problems within the Standard Department. This subgroup of developers, dedicated to the standard software, found software skills, software processes, and product quality increasingly important.

Data collection

Data were collected covering a period of ten years, where we periodically collaborated with SmallSoft as employees, as university teachers, and as researchers engaged in action research (Mathiassen, 2002; McKay and Marshall, 2001; Susman and Evered, 1978). Our relationship with the firm began in 1999 when one author was employed as project leader, participating in the implementation of the new quality assurance system (QA-system) (launched in 1996) and ended in 2005 when an action research project was concluded. Through these activities, it was possible to collect a diverse and rich selection of data from a variety of sources covering a relatively long period of time. First, as part of a university-based action learning program (see section SPI-Action Learning) a group of employees at SmallSoft engaged in an SPI-initiative, starting in 2000 and supervised by one of the authors; the resulting report included assessments of maturity, analysis of practice, descriptions of interventions, and a new SPI-organization. Second, in an interview in March 2004 the key manager at SmallSoft described and reflected on important events in the firm's quality assurance (QA) initiative from 1996 to 2004. Third, in fall 2005 we interviewed other key actors about their view on SPI-technology adoption. The fourth main source was the other author's participation in an action research project with SmallSoft from fall 2003 to fall 2005; this initiative provided e-mail correspondence, documents, notes and minutes from meetings, plus extensive research notes. Adding flesh to these primary data sources, we also had access to the firm's internal SPI-document archive, graduate student reports from collaboration with the firm, recordings from meetings and work situations inside the firm, and interviews with research colleagues. Finally, we had extensive experiences from our personal participation in the process of adopting SPI-technology at SmallSoft.

Two aspects of this data collection process require special attention. First, part of the data came from the authors' engagement in an action learning project and an action research project; this direct participation introduces possible bias in selecting and interpreting data. Second, part of the data draws on interviews with different stakeholders in which the interviewees were asked to retrospectively reflect on important SPI-technology adoption. While the use of retrospective data in process studies is quite common, it involves the risk that interviewees forget or rationalize what originally happened. To reduce the adverse effect of these factors we took advantage of our access to many different complementary sources to triangulate findings (Yin, 1994).

Data analysis

Initially, we combined Newman and Robey's encounter-episode distinction (1992) with Ciborra's control-drift distinction (Ciborra *et al.*, 2000; Ciborra, 2002) to form a historical map covering a time span of ten years (1996-2005). Working systematically through the data sources, we identified candidate encounters by selecting events that were either mentioned in the interviews as important, or that we found to have had significantly impacted the adoption process. In doing so, we focused on events that were caused by management's attempts to control SPI-technology adoption and on events that occurred in SmallSoft's environment. As an initial step towards a process analysis, we mapped the identified encounters according to the chronological timeline and described them briefly together with the intermediary episodes.

Subsequently, we went deeper into the data and started to analyze each encounter-episode. In doing so, we focused on SPI related activities, on their impacts on software development practices, and on activities and events in SmallSoft's environment relevant to the adoption of SPI-technology. As these more detailed analyses progressed, we iterated the selection and description of encounters and episodes until a satisfactory analysis emerged. Hence, as we analyzed how SPI-technology adoption at SmallSoft was shaped by actors with different interests over time, we constantly looked for evidence that our selection of encounters reflected significant events in shaping SPI-technology adoption at SmallSoft. These efforts led to confirmation of some encounters; but we also had to discard, modify, or replace other encounters.

Finally, to make sense of how SPI-technology was adopted at SmallSoft viewed from the perspective of control and drift, we identified two intrinsically related socio-technical networks. First, there was the relatively stable and powerful production-network in which managers and software developers across SmallSoft's three departments developed new solutions in response to customer requests. Second, there was the less stable and weaker improvement-network through which a small group of different actors over time attempted to improve practices in the production-network through the adoption of new development technologies. These two networks offer complementary perspectives on Smallsoft's adoption of

SPI-technology, one focusing on production of software and the other on the ongoing SPI-efforts. In line with Actor Network Theory (Latour, 1987; Callon, 1986; Callon and Law, 1989; Law, 1991; Walsham, 1997), each network includes actors within the firm, ways of working, current and emerging technologies, as well as relationships to forces outside the firm.

Working through the encounters and data once again we wrote up a coherent account of SPI-technology adoption and of how it was shaped through managerial control and improvisations triggered by unpredictable events in the environment. While the choice of encounters at this point had stabilized, our interpretations changed as we negotiated the final version of the story between the data, the theoretical framing, and the experiences and knowledge of the two authors.

Process analysis

The adoption of SPI-technology in Smallsoft passed through seven encounters and subsequent episodes during 1996-2005. This chronology is summarized in Table I.

ISO-9001 Certification

The QA-effort was initiated in 1996 and successfully completed in 1998 with an ISO-9001 certificate. Smallsoft's QA-system was developed by a group consisting of a department manager operating as QA-coordinator and two key developers representing both management interests and engineering practices. The QA-coordinator's interest was to adopt best practices firm-wide by including these in the QA-system as mandatory procedures. Top-level management focused, however, on the certificate; they believed it would help to promote Smallsoft as a professional software house. The QA-system was implemented as an on-line library of procedures, checklists, and templates supplemented with an internally developed requirement, configuration, and change management tool. No permanent QA-group and no standard for operating the QA-system or other quality techniques were implemented. Project managers were expected to comply with the QA-system and all employees received introductory training in an attempt to spark the improvement of practice in Smallsoft's production-network.

After the certification, management left no doubt about the order of priorities in Smallsoft as successful sales resulted in increased workloads; everybody's focus was on producing software and serving customers. There was no time and energy left to institutionalize procedures or reflect on resulting practices. Neither the QA-group, nor management followed up on the implementation of the QA-system. The newly formed improvement-network was largely reduced to the QA-system itself and did not bring any noteworthy changes into practice. However, in the Standard Department, severe problems with configuration management were solved by implementation of a new tool into the QA-system; the tool was initially developed by employees in response to personal needs. When the tool was introduced firm-wide it met no resistance. As a result, configuration management practices changed quickly and led to better customer relations.

There was no doubt that the certification process improved the quality of the software process in Smallsoft, but management and the employees very quickly lost interest in quality assurance after the successful certification.

Table I. Chronology of SPI-technology adoption

	SPI-activity	SPI-organization	Immediate impact	Subsequent impact
ISO 9001 Certification (1996-1998)	Designing ISO-9001 QA system	An *ad hoc* representative QA-group (two employees and the QA-coordinator)	Successful ISO-9000 certification and implementation of new configuration management tool	Integration of QA-system into practice fails. Sustained use of configuration management tool
SPI-Action Learning (Spring 2000 to January 2001)	Participating in action learning program, in project planning, and SPI	An *ad hoc* action learning group (three key employees). QA coordinator internal sponsor	Successful learning as new knowledge and energy was fed into the firm. No change of practices	Assessments planned and possible improvements identified
SPI-Pilot Projects (Autumn 2000 to March 2001)	Assessing processes, planning and designing new process, conducting SPI pilots	An *ad hoc* action learning group (three key employees). Two production projects as pilots. Top management and supervisor in steering committee	Successful CMM assessment. Knowledge of appropriate change practice gained. New processes designed and tested	New knowledge of SPI and organizational change acquired. Sporadic quality review practices
Forming SPI Organization (January 2001 to June 2001)	Designing, negotiating, and deciding on new centralized SPI organization	A new centralized SPI group appointed (two from action learning group and QA coordinator). *Ad hoc* PIT groups for each SPI initiative	Successful design and decision on a new SPI –organization. Members appointed to SPI-group that started working	No change of practice. Successful evaluation of organizational change documented
SPI-Champion Exit (November 2002 to Spring 2004)	Participating in SPI-action research project. Forced to focus on sales activities ended the SPI-activities	Up until breakdown: The same central SPI-group with action researchers as consultants. After breakdown: No SPI-organization	The SPI-champion left the firm, partly because of the breakdown of SPI-initiatives	All SPI-initiatives stop
Learning from SPI-Failure (Summer 2004 to Summer 2005)	Management realizing SPI-breakdown and starting to reflect on present and past improvement practices	No SPI-organization. QA-coordinator still engaged personally	QA-coordinator develops new understanding of SPI-practice. Renewed collaboration with researchers	Ongoing reflections on ideas and research to improve SPI efforts
A Grassroots Approach (November 2005)	Designing and implementing an SPI PIT-organization	A permanent PIT organization implemented. All employees participate in improvement team. QA coordinator is sponsor	Successful and enthusiastic design and implementation of new grassroots SPI organization	Three PITs (process improvement team) working. Eighteen employees and managers involved in improvement network. Future success unpredictable

SPI-action learning

In spring 2000, the local university offered Smallsoft an action learning education (Mathiassen *et al.*, 1999) in project management and improvement. Smallsoft's management welcomed this opportunity for inexpensive improvements and granted participation of three key employees (the action learning group). The education would engage them in an informal university-industry-network and management hoped that this would develop new relationships within the region, boost Smallsoft's image as a professional firm, and provide a useful update on software engineering. Management left the initiative to the action-learning group and kept its own focus on the production-network.

Theory was taught at the university, and action learning was conducted as supervised projects, applying theory to support interventions in each participating firm. While attending the course, the action-learning group studied project management and SPI theory and engaged in discussions with teachers and fellow students from other firms.

The action-learning group identified difficulty prioritizing improvement work in relation to everyday production work as the main threat to successful SPI in Smallsoft. Supported by their academic supervisor, the group decided to formally establish an improvement-network as a counterweight to the production-network and as a means to more effectively involve management and colleagues in SPI-activities.

SPI-pilot projects

The action-learning group agreed with management to conduct interventions according to the IDEAL-model (McFeeley, 1996). In the diagnosis phase, they assessed Smallsoft's process maturity through a simplified CMM-like questionnaire focused on CMM level two: requirement management, project planning, project tracking and oversight, subcontract management, quality assurance, and configuration management.

Smallsoft failed the assessment, fulfilling less than half of the requirements for the areas "project tracking and oversight" and "quality assurance", and up to 70 per cent for "configuration management". The scores surprised the action learning group, since requirement management and project planning were emphasized in the QA-system offering procedures, standards, and IT-based tools for both.

The action-learning group presented the assessment to management and their supervisor. Management wanted to keep investments low by integrating previously planned improvements on project planning, and the group agreed on two interventions: "Project planning, tracking, and oversight" and "quality assurance". At the same meeting, the concept of "quality meetings", implementing external reviews throughout a project's life-cycle, was created and agreed on as the QA-initiative.

The action-learning group wanted to facilitate SPI by testing the new processes, by gaining improvement experience, and by winning ambassadors for the improvement-network. Hence, interventions into two pilot projects were planned by the action learning group following the management policy of no extra workload for pilots. The group planned four meetings with each pilot, two of which prepared and planned the intervention while two practiced the new processes under the group's supervision.

The introductory discussion of assessment results and participation in planning the interventions created a positive attitude. However, applying the newly designed procedures and templates for project planning was a failure. The first pilot had already committed externally to a plan, so they preferred instead to focus on risk and stakeholder analysis. The other pilot planned according to the proposed procedure, but shortly after, the customer wanted to turn the project into a flexible prototyping exercise. Both initiatives failed, simply because they lacked coordination with the stronger production-network.

The introduction of quality meetings was more successful in both pilots. The action learning group used a workshop to introduce a coherent, flexible, and simple one-page tool to support planning, conducting, and documenting quality meetings. Most developers were interested in more management attention on individual projects and supported the new practice, confident that it would improve products and practice. The first quality meeting in both pilots led to improvements in plans and products, and the participants found the practice simple, useful, and effective.

The pilots were evaluated both by management and as part of the education and the results contributed to the forming of a new SPI-organization for Smallsoft.

Forming the SPI-organization

While engaged in the pilots, the action-learning group negotiated a plan for continued SPI with management over three meetings. As a result, management gradually adopted SPI as a way to promote Smallsoft as a professional software house thus supporting the improvement-network.

After the diagnosis in October 2000, management was introduced to SPI by the university supervisor and committed to participate in a workshop on further adoption of SPI-technology in Smallsoft. At the workshop, the action-learning group provided management with a status of the pilots, an overview of all existing improvement initiatives at Smallsoft, and a tailored proposal for SPI in Smallsoft based on CMM. The group had recorded 25 improvement initiatives that were either planned or so-called “bubblers” – improvements spontaneously initiated within a department and somewhat “bubbling up from below”. The “bubblers” were a well-known part of the Smallsoft culture, as employees were expected to do their best and change practices if needed. This high degree of delegation led over time to important improvements of software practices, starting as local or even individual initiatives.

Management approved the proposal as interesting and feasible, but asked for more detailed planning before they could commit themselves. In the beginning of March 2001, when the pilots were completed, the action-learning group presented a detailed description of an SPI-organization for Smallsoft. The group suggested limiting innovation to project management, but top-level management wanted to include software engineering and customer relations and increased the man hours for this work from 500 to 800 per year. Revising and implementing practices tested in the pilots and preparing for a new CMM assessment were the first tasks. The SPI organization (Figure 1) was centered around an SPI-group, coordinating and planning initiatives handled by dedicated process improvement teams (PITs) consisting of members of the SPI-group and interested developers.

It was planned to present the new SPI-organization and the SPI-group at the next employee meeting in April 2001. Due to the well-prepared proposal, the comfortable order situation, and the positive experience with the QA-system, management was very supportive and proclaimed that Smallsoft would reach CMM level three in three

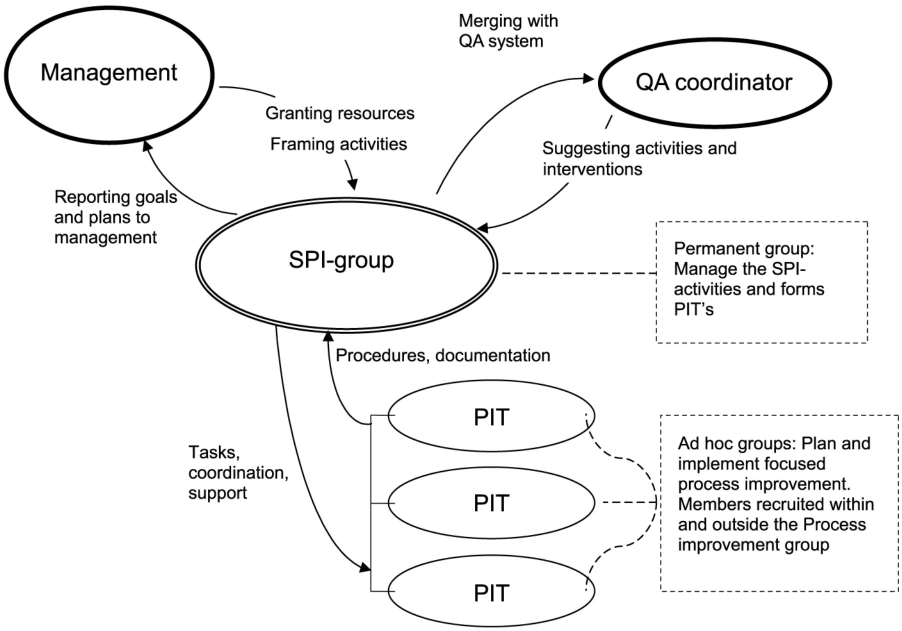

Figure 1.
SPI-organization based on CMM

years. However, the QA-coordinator and the SPI-group were skeptical and the formation of the SPI-group made some experienced developers feel undercut in their professional authority.

The SPI-group included the QA-coordinator and employees with extensive SPI-knowledge. All members were, however, deeply involved in the production-network and could hardly find time or energy for SPI. When the group met for the first time in June 2001, members were already running late according to plans and another meeting was required in August before they – six months late – presented the SPI-organization at a staff meeting in November.

One member acted as SPI-champion; he deeply believed in the quality meeting concept as a realistic way to improve practices and products. He pushed the concept as part of the August meeting agenda and presented detailed and revised quality meeting practices. His enthusiasm persuaded the other members to implement the concept in their departments before the next SPI-group meeting. He personally conducted the first quality meeting in September 2001.

In summary, while management believed SPI-technology adoption was on track, this period was characterized by low SPI-activity and neglect or even hostility from developers outside the improvement-network. The SPI-group could hardly find time to meet and the SPI-champion did not succeed in engaging the other members. No additional developers were involved since no PITs were formed.

Priorities in the firm still favored the production-network, but the QA-coordinator, having no energy left for SPI-initiatives, still looked for ways to empower the improvement-network. In fall 2002, he engaged a student group from the local university to analyze the ongoing SPI efforts. The students recommended decentralization of SPI, based on self-improving software teams, due to the customer centered culture and the high degree of delegation at Smallsoft. The analysis was discussed with the QA-coordinator, but it had no visible influence on the improvement-network. In fact, adoption of SPI-technology had failed and the improvement-network continued to be characterized by rather weak and heterogeneous interests.

SPI-champion exit

Early 2003, Smallsoft entered an action research project with the local university hoping to revitalize SPI-technology adoption. The researchers and the SPI-group planned SPI-activities such as a best practice survey, implementation of code review procedures, and improvement of the quality meeting practice (five quality meetings were conducted during the first six months of 2003). The amount of activities was overwhelming for the SPI-group, but since the SPI-champion hoped the collaboration could bring about real change, he dedicated personal energy and working hours for others.

During fall 2003, Smallsoft experienced a decline in market and intensified their sales efforts. They had to downsize for the first time ever in spring 2004. The situation drained energy and took the focus away from the renewed SPI-effort, and Smallsoft postponed or cancelled most activities. The SPI-champion became increasingly frustrated and finally left the firm. The champion disagreed with management priorities and believed that a strong improvement-network was crucial for the long-term survival of the firm. As a result, all SPI-activities stopped, but the research project manager pushed Smallsoft to either re-engage in or to leave the research project. This external pressure led to negotiations of continued collaboration.

Learning from SPI-Failure

The failure of the centralized, norm-based SPI-initiative made the QA-coordinator reflect on the situation. At this point, in fall 2004, he defined the central organization as a complete failure:

> We wanted to have a central SPI-group that would coordinate and manage SPI-activities. We told people to forward their ideas for improvement to us, and then we would try to implement them in practice. It was a complete failure (interview with the QA-coordinator)

He pointed to three reasons for the failure: lack of time in the SPI-group, neglect from everybody else, and the need to maintain monthly economic surplus for the firm limiting investments in innovation. The QA-coordinator was, however, still committed to the improvement-network, since he believed that continuous improvements were necessary to sustain the firm.

He realized the dominance of the production-network over the improvement-network and reflected on how successful improvements had been driven by production needs, thus creating overlap between the production-network and the improvement-network. Several improvements had been implemented during the relatively short lifetime of the firm, especially prior to the formation of the SPI-group. These improvements were often initiated by developers experimenting with innovations and subsequently spread by word of mouth ("bubblers"). Two of the most significant; the configuration management system and the QA-system, were initiated this way before they were upgraded to firm level and granted resources. After the SPI-group was formed, employees and management still reflected on and learned from the SPI-experiences, but hardly focused on adapting personal practice accordingly.

However, improvements did appear i.e. a better quality of project planning and an increased frequency of code-reviews.

> I started worrying if forming a central SPI-group had killed the grassroots improvements, and if no activity in the SPI-group, would result in no improvements at all (interview with the QA-coordinator)

The QA-coordinator started to advocate an approach to SPI where improvements were to be initiated from the grassroots of the firm (individually or locally), driven by personal involvement of employees. This idea was in line with the earlier recommendations from the student group. Three considerations led him to the following conclusion: small firms cannot afford to invest in a dedicated improvement unit; the centralized SPI-initiative had implied increasing overhead; and a centralized SPI-group would not know established practices well enough to propose realistic and efficient changes. The researchers agreed to investigate how a grassroots approach to SPI could be facilitated and implemented so that the improvement-network was more in line with the production-network at Smallsoft.

A grassroots approach

Projects and individuals of Smallsoft had proven capable of improving practices, but the exploration of grassroots approaches to SPI raised the question of how local improvements spread firm wide. In December 2004, the researchers conducted a social network analysis of the improvement-network, displaying very close connections within each department, but only loose connections across department borders and to top level management (Nielsen and Tjørnehøj, 2005). Thus, Smallsoft had a highly developed network for local improvements, but no basis for sharing knowledge and practices across departments. The researchers suggested either making SPI-initiatives local to departments losing the benefits from sharing knowledge, or building relations between departments to support knowledge sharing.

These insights strongly impacted an SPI-strategy meeting in March 2005, engaging two department managers (including the QA-coordinator) and top level management in a sincere discussion of experiences with SPI, the QA-system, and the problems and benefits of having self-governed developers and departments. In the light of a sales boom, making it difficult for the new department in order to keep up with demands, management felt they could learn from the two established departments to avoid well-known failures. Thus, knowledge sharing between departments was given priority within Smallsoft.

One of the researchers had just learned how a reputed Danish firm had successfully moved to level 3 in CMM by involving all employees in PITs. This success inspired Smallsoft to form eight improvement initiatives across departments in a matrix-like PIT organization. Management decided to engage all developers in one or more PITs and, as a strong sign of management commitment, they assigned man-hours to the task. Employees were assigned during summer 2005 to ensure participation from all departments and more projects in each PIT. In August 2005, the PIT organization was launched at a kickoff meeting. Most developers participated and engaged in discussing the implications of the new approach. The urgent PITs started immediately while less urgent ones would start within a year. The PITs were self-organized and autonomous with respect to improvement directions.

In November 2005, three out of eight PITs were active, but working at a slower pace than expected. The PITs found major differences in practices between the three departments. Members from the established departments were expected to transfer practices to the new department, but the new department had its own traditions and preferences. Despite these challenges, the PITs made progress and arrived at realistic solutions. A total of 18 developers and parts of management were now participating in an improvement-network that seemed to grow stronger than ever before during adoption of SPI-technology at Smallsoft.

Discussion

Smallsoft did indeed experience the difficulties that the literature identifies as being related to adoption of SPI-technology; lack of resources and SPI-knowledge (Steel, 2004; Brouse and Buys, 1999; Villalon *et al.*, 2002; Brodman and Johnson, 1994; Kilpi, 1998; Saastamoinen and Tukiainen, 2004); sensitivity towards changing environments (Ward *et al.*, 2001); and incongruity between small firm culture and SPI-theory (Leung and Yuen, 2001; Varkoi *et al.*, 1999). However, while the literature primarily suggests different forms of downsizing and pragmatic usage of CMM or alternative SPI-technologies designed for small firms, Smallsoft handled these challenges differently.

SPI-adoption at Smallsoft

Smallsoft dealt with the lack of resources and knowledge through networking with the university and local firms (Kautz, 1998). They took advantage of opportunities in their environment and invested in low-cost attempts to adopt SPI-technology without jeopardizing their need to produce surplus every month. However, outcomes always came down to the daily struggle between the everyday tasks of developers and the extra workload and risk that changing habits and work practices implied, even when resources were granted from management (Mustonen-Ollila and Lyytinen, 2003).

Table I summarizes the presented encounters with SPI-technology adoption at Smallsoft followed by episodes focused on everyday production work. In the short term, each SPI-initiative was successful; developers and managers found them useful and promising; but in the subsequent episode, SPI-innovations were overshadowed by production concerns so effects seldom materialized as planned.

However, employees and management did reflect on the experiences, internalized some of the theories, methods and techniques, and improvements did emerge over time, but not as planned results of firm level decisions, rather as local or individual changes in practice. This was underpinned in the interviews with the key actors in fall 2005, when we investigated if any improvements had taken place. They emphasized both personal learning and more concrete examples of improvements emerging from the encounters, e.g. matured project planning and tracking and reviews to increase quality of code. Also the “Learning from SPI-failure” encounter was emphasized by the QA-coordinator as a major breakthrough for Smallsoft’s SPI efforts.

The ongoing struggle to develop and sustain an improvement-network along with the dominating production-network involved managers and employees; it crossed organizational borders, it was reinforced through different forms of technology, and it was heavily influenced by external events and forces. Throughout, actors tried to persuade colleagues to see SPI as a long-term contribution to their production interests, and management (in particular the QA-coordinator) continued to support the improvement-network through controlled initiatives, insisting that SPI was important for Smallsoft. These efforts were, as described above, mainly successful when improvements were perceived by employees as concrete and sustainable extensions of the production-network.

The “Grassroots approach” following the “Learning from SPI-failure” encounter was indeed based on the experiences from Smallsoft’s ongoing struggle to adopt SPI-technology. It exploited individual and local initiatives in a new organization that supported improvements through management decisions on subjects, priorities, and resources and through negotiation of new practices between the everyday users representing the production network. Measured on the traditional short term scale, this initiative was successful, but it remains yet to be seen whether it will bring sustainable changes.

Between control and drift

According to Ciborra (2002), control is based on a rational view of the world in which managers understand and plan events by applying simplistic theoretical models to decisions and practices. Drift, on the other hand, emphasizes how side effects, bricolage, hacking, formative context, and people’s everyday coping make reality drift away from plans, thereby opening up to options and innovations that were otherwise unthinkable.

Smallsoft’s adoption of SPI-technology displays both control and drift elements (Ciborra, 2002; Ciborra *et al.*, 2000). However, contrary to Ciborra, the two elements interacted, their relative dominance shifted as the process unfolded, and both elements had positive impacts on the adoption of SPI-technology. Rather than seeing control and drift as alternative management philosophies, the Smallsoft experience suggests that these are complementary, and intrinsically related opposites of a dialectical relationship (Bjerknes, 1991; Van de Ven and Poole, 1995; Mathiassen, 1998; Robey and Boudreau, 1999). While SPI-adoption at Smallsoft was shaped by several, interacting contradictions (e.g. between developers and managers, between different departments and customers, and between espoused theories and theories-in-use (Argyris and Schön, 1978)), we found the contradiction between control and drift to be particularly helpful in making sense of the reported ten years of SPI-technology adoption.

Initially, the principal aspect of the contradiction was control when Smallsoft chose a centralized and controlling improvement approach to their ISO-9001-certification, while drifting had less influence on the result through the personal hacking of a configuration management system. In the following episode, however, management believed to have laid out the direction and did not exercise control through systematic follow up on the QA system. As a result, drifting dominated and developers did not comply with the procedures unless they were immediately useful. Only when the certificate was renewed did the ISO-9001 assessments reinforce control.

When Smallsoft unexpectedly was offered SPI-action learning, a change initiated by drift, management engaged to regain control over and create momentum for the SPI-process. The formal organization of the education and the rational worldview of the theories taught and later applied emphasized control both during this encounter and the SPI-pilot projects. While some drifting appeared as a result of pressures from customers, control dominated based on influences from management, the action-learning group, the teachers, the adopted theories, and the experiences from other influential Danish firms.

The SPI-organization was also initially conceived as a centralized decision and control mechanism, involving experts, management, and internal change agents, and leaving only limited room for everyday coping, unexpected events, hacking, or unplanned innovations. The proposed organization was at first broadly accepted, but then by and large ignored. In the following episode the power of controlling forces diminished because the SPI-group struggled to find time for improvements due to everyday working responsibilities in their departments. As a result, the SPI-organization and – activities drifted away from what was intended.

The attempt to control SPI-adoption by entering the action-research project was effectively stopped by drifting forces from outside Smallsoft. As the market declined and the economic situation at Smallsoft got worse, all planned improvement initiatives were put on hold and the SPI-champion left the firm. Following this breakdown, management realized that one-sided rational attempts to control SPI were costly for Smallsoft, because the benefits of drifting forces, i.e. effective bricolage, hacking, and tinkering from self-organized "bubblers" and unplanned events, almost disappeared. On the other hand, sharing knowledge, best practices, and strategic goals across departments was important for Smallsoft to stay competitive and introduced needs for exercising some control.

The "unthinkable" innovation, the Grassroots Approach, gave management a sense of control while leaving amble space for negotiating new practices based on everyday coping and survival strategies developed by employees who volunteered to engage in PITs. Though management still exercised control by prioritizing focus areas, by granting resources, by appointing experts, and by requesting monthly reporting, they also embraced drifting to help develop and share successful innovations.

This account of the struggle between control and drift illustrates how the two opposites interacted and both positively impacted the ongoing SPI-adoption at Smallsoft. New SPI-theories and -models offered control approaches and brought new knowledge to management, internal SPI-agents, and employees. Management's attempts to control SPI-adoption framed collaborative experimenting and learning that kept the improvement-network alive. Management's continued control efforts kept Smallsoft vigilant by insisting on and pushing the organization towards change.

At the same time, the options unexpectedly offered from outside the firm, such as education, student involvement, and the research project, provided Smallsoft with resources and knowledge to effectively engage in improvements. The employees' everyday coping, bricolage and hacking helped adapt SPI-technology to the everyday reality of the firm by stopping failing initiatives, by aligning improvement initiatives with the production-network, and by creating unplanned innovations. Hence, while continued efforts to control created momentum, informed the process, and provided strategic direction, drifting forces secured real world results by making changes take place in the production-network.

Hence, the Smallsoft experience suggests that managers of small software firms remain flexible and constantly negotiate technology adoption practices between control and drift; creating momentum and direction according to firm goals through attempts to control, while at the same time exploring backtalk, options, and innovations from drifting forces inside and outside the firm. In such a dialectical approach to technology adoption, control and drift represent complementary and intrinsically related approaches that, if appropriately negotiated, support each other and lead to acceptable adoption outcomes.

Note

1. Ciborra explains bricolage as the activity when humans leverage the world as defined by the situation, improvisation as dealing with sudden unpredictable interventions and hacking as an ingenious activity that through iterations, reuse, and reinterpretations of the existing programming environment leads to the implementation of new solutions. Ciborra (2002).

References

Aaen, I., Arendt, J., Mathiassen, L. and Ngwenyama, O. (2001), "A conceptual map of software process improvement", *Scandinavian Journal of Information Systems*, Vol. 13, pp. 81-101.

Ares, J., García, R., Juristo, N., López, M. and Moreno, A.M. (2000), "A more rigorous and comprehensive approach to software process assessment", *Software Process: Improvement and Practice*, Vol. 5, pp. 3-30.

Argyris, C. and Schön, D.A. (1978), *Organizational Learning: A Theory of Action Perspective*, Addison-Wesley Publishing Company, Reading, MA.

Balla, K., Bemelmans, T., Kusters, R. and Trienekens, J. (2001), "Quality through managed improvement and measurement (QMIM): towards a phased development and implementation of a quality management system for a software company", *Software Quality Journal*, Vol. 9, pp. 177-93.

Batista, J. and Figueiredo, A.D.D. (2000), "SPI in a very small team: a case with CMM", *Software Process: Improvement and Practice*, Vol. 5, pp. 243-50.

Bennetts, P.D.C. and Wood-Harper, A.T. (1998), "The soft system methodology as a framework for software process improvement", *Journal of End User Computing*, Vol. 10, pp. 12-19.

Bjerknes, G. (1991), "Dialectical reflection in information systems development", *Scandinavian Journal of Information Systems*, Vol. 3, pp. 55-77.

Brigham, M. and Introna, L.D. (2006), "Hospitality, improvisation and Gestell: a phenomenology of mobile information", *Journal of Information Technology*, Vol. 21, pp. 140-53.

Brodman, J.G. and Johnson, D.L. (1994), "What small businesses and small organizations say about the CMM", paper presented at the 16th International Conference on Software Engineering, Sorrento.

Brouse, P.S. and Buys, R.T. (1999), "Affordable ways to improve application development", *IT Professional*, Vol. 1, pp. 47-52.

Börjesson, A. and Mathiassen, L. (2003), "Making SPI happen: the IDEAL distribution of effort", paper presented at the 36th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Big Island, Hawaii.

Börjesson, A. and Mathiassen, L. (2005), "Improving software organizations: the agility challenge", *Information, Technology & People*, Vol. 18, pp. 359-82.

Callon, M. (1986), "Some elements of a sociology of translation: domestication of scallops and the fishermen of St Brieuc Bay", in Law, J. (Ed.), *Power, Action and Belief: A New Sociology of Knowledge?*, Routledge & Kegan Paul, London.

Callon, M. and Law, J. (1989), "On the construction of socio technical networks: content and context revisited", *Knowledge and Society*, Vol. 9, pp. 57-83.

Casey, V. and Richardson, I. (2004), "A practical application of the IDEAL model", *Software Process: Improvement and Practice*, Vol. 9, pp. 123-32.

Cho, S., Mathiassen, L. and Nilsson, A. (2006), *Event-Based Actor Network Analysis of IT-Based Change*, Georgia State University, Atlanta, GA.

Ciborra, C.U. (1994a), "From thinking to tinkering", in Ciborra, C.U. and Jellasi, T. (Eds), *Strategic Information Systems*, John Wiley and Sons, Chichester.

Ciborra, C.U. (1994b), "A platform for surprises: the organization of global technology strategy at Olivetti", *IFIP WG8.2 Working Conference on Information Technology and New Emergent Forms of Organizations: Transforming Organizations with Information Technology*, North-Holland Publishing Co., Amsterdan.

Ciborra, C. (1999), "Notes on improvisations and time in organizations", *Accounting, Management and Information Technologies*, Vol. 9, pp. 77-94.

Ciborra, C. (2002), *The Labyrinths of Information: Challenging the Wisdom of Systems*, Oxford University Press, New York, NY.

Ciborra, C. and Hanseth, O. (1998), "From tool to Gestell – agendas for managing the information infrastructure", *Information Technology & People*, Vol. 305 -327.

Ciborra, C. and Lanzara, G.F. (1994), "Formative contexts and information technology: understanding the dynamics of innovation in organizations", *Accounting, Management and Information Technologies*, Vol. 4, pp. 61-8.

Ciborra, C.U., Braa, K., Cordella, A., Dahlbom, B., Failla, A. and Hanseth, O. (2000), *From Control to Drift: The Dynamics of Corporate Information Infrastructures*, Oxford University Press, Oxford.

Coleman, G. and Verbruggen, R. (1998), "A quality software process for rapid application development", *Software Quality Journal*, Vol. 7, pp. 107-22.

Dion, R. (1992), "Elements of a process-improvement program", *IEEE Software*, Vol. 9, pp. 83-5.

Dybå, T. (2003), "Factors of software process improvement success in small and large organizations: an empirical study in the Scandinavian context", *ACM SIGSOFT Software Engineering Notes*, Vol. 28, pp. 148-57.

El-Emam, K., Goldenson, D., McCurley, J. and Herbsleb, J. (2001), "Modeling the likelihood of software process improvement: an exploratory study", *Empirical Software Engineering*, Vol. 6, pp. 207-29.

Elbanna, A.R. (2006), "The validity of the improvisation argument in the implementation of rigid technology: the case of ERP systems", *Journal of Information Technology*, Vol. 21, pp. 165-75.

Grechenig, T. and Zuser, W. (2004), "Creating organic software maturity attitudes (COSMA) selected principles and activities for software maturity in small and medium software enterprises", paper presented at the 4th International Conference on Quality Software (QSIC'04). Braunschweig.

Hansen, B., Rose, J. and Tjørnehøj, G. (2004), "Prescription, description, reflection: the shape of the software process improvement field", in Powell, P. and Grant, K. (Eds), *UK Association of Information Systems Conference*, Glasgow Caledonian University, Glasgow.

Hanseth, O., Ciborra, C. and Braa, K. (2001), "The control devolution: ERP and the side effects of globalization", *SIGMIS Database*, Vol. 32, pp. 34-46.

Harjumaa, L., Tervonen, I. and Vuorio, P. (2004), "Using software inspection as a catalyst for SPI in a small company", *Product Focused Software Process Improvement*, Springer, Berlin/Heidelberg.

Horvat, R.V., Rozman, I. and Györkös, J. (2000), "Managing the complexity of SPI in small companies", *Software Process: Improvement and Practice*, Vol. 5, pp. 45-54.

Humphrey, W.S., Snyder, T.R. and Willis, R.R. (1991), "Software process improvement at Hughes Aircraft", *IEEE Software*, Vol. 8, pp. 11-23.

Jakobsen, A.B. (1998), "Bottom-up process improvement tricks", *IEEE Software*, Vol. 15, pp. 64-8.

Kautz, K. (1998), "Software process improvement in very small enterprises: does it pay off?", *Software Process: Improvement and Practice*, Vol. 4, pp. 209-26.

Kautz, K.H. (1999), "Making sense of measurement for small organizations", *IEEE Software*, Vol. 16, pp. 14-20.

Kautz, K.H., Hansen, H.W. and Thaysen, K. (2000), "Applying and adjusting a software process improvement model in practice: the use of the IDEAL model in a small software enterprise", *22nd International Conference on Software Engineering*, ACM Press, Limerick.

Kehoe, R. and Shah-Jarvis, A. (1995), *ISO 9000-3: A Tool for Software Product and Process Improvement*, Springer-Verlag, New York, NY.

Kelly, D.P. and Culleton, B. (1999), "Process improvement for small organizations", *Computer*, Vol. 32, pp. 41-7.

Kilpi, T. (1998), "Product management challenge to software change process: preliminary results from three SMEs experiment", *Software Process: Improvement and Practice*, Vol. 3, pp. 165-75.

Kuvaja, P., Palo, J. and Bicego, A. (1999), "TAPISTRY – a software process improvement approach tailored for small enterprises", *Software Quality Journal*, Vol. 8, pp. 149-56.

Latour, B. (1987), *Science in Action: How to Follow Scientists and Engineers through Society*, Harvard University Press, Cambridge, MA.

Law, J. (1991), "Introduction: monsters, machines, and socio technical relations", in Law, J. (Ed.), *A Sociology of Monsters: Essays of Power, Technology and Domination*, Routledge, London.

Leung, H.K.N. and Yuen, T.C.F. (2001), "A process framework for small projects", *Software Process: Improvement and Practice*, Vol. 6, pp. 67-83.

Lévi-Strauss, C. (1962), *La pensée sauvage*, Librairie Plon, Paris.

Lévi-Strauss, C. (1972), *The Savage Mind*, Oxford University Press, New York, NY.

McFeeley, B. (1996), *IDEAL: A User's Guide for Software Process Improvement*, Software Engineering Institute, Pittsburgh, PA.

McKay, J. and Marshall, P. (2001), "The dual imperatives of action research", *Information Technology & People*, Vol. 14, pp. 46-59.

Mathiassen, L. (1998), "Reflective systems development", *Scandinavian Journal of Information Systems*, Vol. 10, pp. 67-118.

Mathiassen, L. (2002), "Collaborative practice research. information", *Technology & People*, Vol. 15, pp. 321-45.

Mathiassen, L., Borum, F. and Pedersen, J.S. (1999), "Developing managerial skills in IT organizations: a case study based on action learning", *Journal of Strategic Information Systems*, Vol. 8, pp. 209-25.

Mathiassen, L., Pries-Heje, J. and Ngwenyama, O. (Eds) (2002), *Improving Software Organizations: From Principles to Practice*, Addison-Wesley, Reading, MA.

Mustonen-Ollila, E. and Lyytinen, K. (2003), "Why organizations adopt information system process innovations: a longitudinal study using diffusion of innovation theory", *Information Systems Journal*, Vol. 13, pp. 275-97.

Newman, M. and Robey, D. (1992), "A social process model of user-analyst relationship", *MIS Quarterly*, Vol. 16, pp. 249-66.

Ngwenyama, O. and Nielsen, P.A. (2003), "Competing values in software process improvement: an assumption analysis of CMM from an organizational culture perspective", *IEEE Transactions on Engineering Management*, Vol. 50, pp. 100-12.

Nielsen, P. and Tjørnehøj, G. (2005), "Mapping social networks in software process improvement: an action research study", *Business Agility and Information Technology Diffusion*, Springer, Boston, MA.

Paulk, M.C. (1998), "Using the software CMM in small organizations", *The Joint Pacific Northwest Software Quality Conference and the Eighth International Conference on Software Quality*, pp. 350-61.

Paulk, M.C., Curtis, B., Chrissis, M.B. and Weber, C. (1993), *Capability Maturity Model for Software Version 1.1, Technical Report CMU/SEI-93-TR-24*, Software Engineering Institute, Pittsburgh, PA.

Peterson, M.F. (1998), "Embedded organizational events: units of process in organization science", *Organization Science*, Vol. 9, pp. 16-33.

Pettigrew, A.M. (1990), "Longitudinal field research on change: theory and practice", *Organization Science*, Vol. 1, pp. 267-92.

Richardson, I. (2001), "Software process matrix: a small company SPI model", *Software Process: Improvement and Practice*, Vol. 6, pp. 157-65.

Richardson, I. (2002), "SPI models: what characteristics are required for small software development companies?", *Software Quality Journal*, Vol. 10, pp. 101-14.

Robey, D. and Boudreau, M.-C. (1999), "Accounting for the contradictory organizational consequences of information technology: theoretical directions and methodological implications", *Information Systems Research*, Vol. 10, pp. 167-85.

Rodenbach, E., Van Latum, F. and Van Solingen, R. (2000), "SPI – a guarantee for success? A reality story from industry", *Product Focused Software Process Improvement*, Springer, Berlin.

Ruiz, M., Ramos, I. and Toro, M. (2002), "A dynamic integrated framework for software process improvement", *Software Quality Journal*, Vol. 10, pp. 181-94.

Saccol, A.Z. and Reinhard, N. (2006), "The hospitality metaphor as a theoretical lens for understanding the ICT adoption process", *Journal of Information Technology*, Vol. 21, pp. 154-64.

Schneider, K. (2002), *Experience Based Process Improvement. Software Quality – ECSQ 2002: 7th International Conference, Helsinki, Finland, 9-13 June 2002, Proceedings*, Springer, Berlin/Heidelberg.

Scott, L., Carvalho, L., Jeffery, R., D'ambra, J. and Becker-Komstaedt, U. (2002), "Understanding the use of an electronic process guide", *Information and Software Technology*, Vol. 44, pp. 601-16.

Scott, L., Jeffery, R., Carvalho, L., D'ambra, J. and Rutherford, P. (2001), "Practical software process improvement – the IMPACT project", paper presented at the 13th Australian Software Engineering Conference (ASWEC'01).

Steel, A.P.C. (2004), "Low-rigour, rapid software process assessments for small software development firms", paper presented at the 2004 Australian Software Engineering Conference (ASWEC'04).

Susman, G.I. and Evered, R.D. (1978), "An assessment of the scientific merits of action research", *Administrative Science Quarterly*, Vol. 23, p. 582.

Saastamoinen, I. and Tukiainen, M. (2004), Springer-Verlag, Trondheim, "Software process improvement in small and medium sized software enterprises in Eastern Finland: a state-of-the-practice study", paper presented at Software Process Improvement: 11th European Conference, EuroSPI.

Team, C.P. (2002), *CMMISM for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1, Staged)*, Technical Report SEI.

Team, C.P.D. (2000), *SCAMPISM, V1.0 Standard CMMISM Assessment Method for Process Improvement: Method Description, Version 1.0*, Carnegie Mellon University, Pittsburgh, PA.

Thowart, K. (1999), "The AMETIST process improvement experiment: towards efficient team development in small companies", *Software Process: Improvement and Practice*, Vol. 4, pp. 11-18.

Truffley, A., Grove, B. and McNair, G. (2004), "SPICE for Small Organizations", *Software Process: Improvement and Practice*, Vol. 9, pp. 23-31.

Van de Ven, A.H. and Poole, M.S. (1995), "Explaining development and change in organizations", *Academy of Management Review*, Vol. 20, pp. 510-40.

Varkoi, T., Makinen, T. and Jaakkola, H. (1999), "Process improvement priorities in small software companies", paper presented at Portland International Conference on Management of Engineering and Technology, Technology and Innovation Management, PICMET '99 Portland, OR.

Villalon, J., Agustin, G.C., Gilabert, T.S.F., Seco, A.D., Sanchez, L.G. and Cota, M.P. (2002), "Experiences in the application of software process improvement in SMES", *Software Quality Journal*, Vol. 10, pp. 261-73.

Walsham, G. (1993), *Interpreting Information Systems in Organizations*, Wiley, Chichester.

Walsham, G. (1995), "Interpretive case studies in IS research: nature and method", *European Journal of Information Systems*, Vol. 4, pp. 74-81.

Walsham, G. (1997), "Actor-network theory and IS research: current status and future prospects", in Lee, A.S., Liebenau, J. and Degross, J.I. (Eds), *Information Systems and Qualitative Research*, Chapman and Hall, London.

Ward, R.P., Fayad, M.E. and Laitinen, M. (2001), "Software process improvement in the small – a small software development company's most difficult challenge: changing processes to match changing circumstances", *Communications of the ACM*, Vol. 44, pp. 105-7.

Wilkie, F.G., McFall, D. and McCaffery, F. (2005), "An evaluation of CMMI process areas for small- to medium-sized software development organizations", *Software Process: Improvement and Practice*, Vol. 10, pp. 189-201.

Wohlwend, H. and Rosenbaum, S. (1994), "Schlumberger's Software Improvement Program", *IEEE Transactions on Software Engineering*, Vol. 20, pp. 833-9.

Yin, R. (1994), *Case Study Research: Design and Method*, Sage, Thousand Oaks, CA.

Corresponding author
Gitte Tjornehoj can be contacted at: gts@cs.aau.dk