Social Networks in Software Process Improvement

Nielsen, Peter Axel; Tjørnehøj, Gitte

Published in:
Software Process Improvement and Practice

DOI (link to publication from Publisher):
10.1002/spip.419

Publication date:
2010

Document Version
Publisher's PDF, also known as Version of record

SOFTWARE PROCESS IMPROVEMENT AND PRACTICE
Softw. Process Improve. Pract. (2009)
Published online in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/spip.419

Social Networks in Software Process Improvement

Peter Axel Nielsen* and Gitte Tjørnehøj
Department of Computer Science, Aalborg University, Selma Lagerlöfs Vej 300, Aalborg East DK-9220, Denmark

Research Article

Software process improvement in small organisation is often problematic and communication and knowledge sharing is more informal. To improve software processes we need to understand how they communicate and share knowledge. In this article have studied the company SmallSoft through action research. In the action research we have applied the framework of social network analysis and we show this can be used to understand the underlying structures of communication and knowledge sharing between software developers and managers. We show in detail how the analysis can be done and how the management can utilise the findings. From this we conclude that social network analysis was a useful framework together with accompanying tools and techniques. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: software process improvement, social network analysis, communication, knowledge sharing, software management

1. INTRODUCTION

Software process improvement (SPI) has long been a concern for software companies. The development of the Capability Maturity Model (CMM) (Humphrey 1989, 1992, 2002) and later CMMI (Ahern *et al*. 2001, 2003; Chrissis *et al*. 2003) by the Software Engineering Institute sparked a huge interest in the field. The CMM family of models has also been supplemented with the IDEAL approach (McFeeley 1996) that addresses how to utilise the CMM.

The focus in this paper is on SPI in small and medium-sized companies. A core characteristic of small software companies seems to be that they often face changing environments and are more vulnerable than large companies. Ward suggests, 'the processes by which software is developed are likely to change with circumstances – perhaps even change dramatically – even while general principles like the need for good communication remain constant' (Ward *et al*. 2001).

Within SPI there is an explicit concern that the maturity-model approaches, like CMM and CMMI, are not adequate for improving small software companies. An early survey raises the concern that the CMM does not fit small software companies (Brodman and Johnson 1994). Several studies of SPI for small companies reveal many difficulties: small companies cannot necessarily afford the investment in SPI (Kautz and Larsen 1997); small companies lack SPI knowledge (Cater-Steel 2001); they see SPI as bureaucratic (Kelly and Culleton 1999); and they see traditional SPI methods as too costly (Villalon *et al*. 2002).

The difficulties for small software companies cannot be attributed to the CMM-based approaches alone as there are reported examples of successful SPI, some which are CMM-based and some which are not. Kautz *et al*. describe a successful improvement effort of a small software company, in which CMM was used for the initial maturity assessment and IDEAL was used to structure the improvement effort (Kautz and Thaysen 2001; Kautz *et al*. 2001).

* Correspondence to: Peter Axel Nielsen, Aalborg University, Computer Science, Aalborg, Denmark
† E-mail: pan@cs.aau.dk

Also (Kelly and Culleton 1999) reports on attempts to develop and test new SPI approaches for small software companies based on CMM. In yet another study it is suggested that while CMM is used for assessment, it is necessary to supplement it with what the authors call an 'action package concept' to overcome small companies' lack of follow-through into action planning and implementation (Villalon *et al*. 2002).

There are reports of successful SPI where CMM or other maturity models were not used. Kautz has studied process improvement in three small companies (Kautz 2000, 1999). The success of SPI in these companies is attributed to four factors (Kautz 2000): a tailored approach; an experience network between companies; external assistance; and partial external funding. In another study a medium-sized company's problems with current software processes were assessed with a technique for problem diagnosis, which was not on the basis of a maturity model and many of the identified problems could later be alleviated (Iversen *et al*. 1999; Nielsen *et al*. 2002). In this study the success of the improvement effort was attributed to the particular way experience and knowledge were shared during the problem diagnosis.

Sharing knowledge, also sometimes referred to as sharing experience, is fundamental in all these reports. Hence, we have undertaken research to understand knowledge sharing better and in greater detail. We report on an action research effort in the software company SmallSoft on how knowledge sharing can be understood through social network analysis and how software managers can utilise social network analysis to manage SPI efforts.

The current theories behind knowledge sharing in SPI and social network analysis are presented in more detail in section 2. In section 3, we outline our research approach and describe the data collection and data analysis. In section 4, we present the case company SmallSoft and how we used social network analysis in that company. In section 5 we discuss the role of social network analysis in SPI in general and for SmallSoft in particular. The article concludes in section 6.

2. KNOWLEDGE SHARING AND SOCIAL NETWORK ANALYSIS

It appears that part of the success of SPI in small companies has to do with how knowledge is shared among the developers and managers partaking in the SPI effort. Knowledge management is a relevant perspective to apply in SPI in general. Kautz and Thaysen concur with this and put forward that knowledge in SPI is not only to be seen as a simple commodity, but needs to be understood in a much broader and social context (Kautz and Thaysen 2001). Understanding knowledge management is a key to SPI according to other studies (Baskerville and Pries-Heje 1999; Mathiassen and Pourkomeylian 2003; Pries-Heje and Pourkomeylian 2004).

There are several reasons why knowledge management is an important perspective. First, it has been established that software development depends hugely on communities-of-practice, which differ from the formal organisation (Mathiassen 1998, p. 88). Communities-of-practice create the specific context as well as the shared experience and understanding of their members in such a way that they shape how new or modified processes are adapted, implemented or rejected.

Second, SPI is a problem-solving activity (Mathiassen *et al*. 2002, p. 4) where problems in software processes have to be identified, needs have to be understood, possible improvements have to be devised and prioritised, and actions to improve must be taken. All these activities require communication of different perceptions and interests, of plans and priorities, and of outcomes.

Third, SPI is also a knowledge creating activity (Mathiassen *et al*. 2002, p. 7) where SPI knowledge needs to be elicited from experience, some experience has to be explicated, concerns for capture and quality of available knowledge have to be addressed, and validated feedback has to be provided.

Fourth, organisational influence processes are important in SPI (Nielsen and Ngwenyama 2002). This study of influence processes concludes that it is crucial to understand the networks through which power and influence is exercised; but also that a major source of power is knowledge and communication skills.

On this background we find it interesting to analyse the networks through which knowledge is shared and communicated in greater detail. We expect that it can advance SPI in general, and in small companies in particular. Social network analysis is a framework for such detailed analyses.

DOI: 10.1002/spip

Social network analysis is a general framework and a set of techniques applied to study the relationships between organisational actors and their exchange of resources; see (Wasserman and Faust 1994; Cross and Parker 2004). In social network analysis organisations are viewed as consisting of actors linked together in networks through action, exchange, and interpretation and sharing of resources like information and knowledge. Social network analysis seeks to provide a way to look at an informal organisation, which exists in parallel to the formal and hierarchical organisation chart. In this view, organisations are made up of interdependent actors with relational ties between them. Network models conceptualise structure as lasting patterns of such relational ties (Wasserman and Faust 1994). Wasserman and Faust further define actors as discrete individuals, or corporate or collective social units, (i.e. not only as a single person). The relational ties can be of varying types: evaluation of one person by another (as with friendship), transfer of material resources, affiliation, and authority (as between managers and subordinates), and behavioural interaction like sending messages and engaging in a discussion (Wasserman and Faust 1994, p. 18).

Social network analysis is not a new approach. It has been developed and applied in a large number of organisational studies (see Tichy *et al*. 1979; Wasserman and Faust 1994; Scott 2000), but it has not been applied directly in SPI efforts before. Social network analysis has been applied to understand software teams (e.g., Yang and Tang 2004; Ehrlich *et al*. 2007; Long and Siau 2007; Müller *et al*. 2008). It has however mostly been applied to distributed software teams and to open source development.

The framework does not provide a unit of analysis and data may be collected about many different kinds of actors and relational ties. It is, however, common to collect data about the contents of the relational ties as well as their intensity and reciprocity. On the basis of the collected data, the approach requires the study of network properties and structural characteristics. Some of the properties that we will also investigate later in this study are the following:

- *Density:* how well-connected are the network's actors?
- *Centrality:* who is the 'most important' actor in a network?

A network can be analysed for these properties, but these are just a few of the analyses that can be performed on a social network. The analyses all have a foundation in graph theory (Borgatti and Everett 1992; Wasserman and Faust 1994; Scott 2000), but the interpretation and the semantic implication of these analyses remain specific to the setting where the data were collected.

Our application of social network analysis focuses on the social networks through which software process improvement may happen and in particular we focus on communication about SPI as a means for sharing knowledge.

3. RESEARCH APPROACH

The research followed collaborative practice research (CPR) that is an action research approach (Mathiassen 2002). The CPR approach builds on (Checkland 1991; McKay and Marshall 2001) and it guides how interventions into software companies' practice can take place and how scholarly knowledge is gained. Mathiassen argues that CPR is suited for research into: (i) how SPI activity may be understood through practice studies, (ii) how support for SPI activity may be developed, and (iii) how interventions may improve SPI activity.

For this purpose the two authors were part of a SPI group in the company SmallSoft over several months; not on a daily basis, but sufficiently often to get a good understanding of the company and its SPI activity.

The data collection and data analysis for the action research study was performed in two parts. For the purpose of understanding the software company and its context the researchers collected data about: (i) SmallSoft's background, (ii) its SPI activity, (iii) its history with SPI, and in particular (iv) all minutes from meetings in the SPI group, and (v) progress reports. The data were analysed informally to inform the researchers and to write the case background in section 4.

For the purpose of taking action informed by social network analysis the researchers collected and analysed the data following a more stringent procedure. The procedure is similar to that outlined by (Cross and Parker 2004, p. 143), and it contains the following steps:

1. Identify the group
2. Collect data about relationships

DOI: 10.1002/spip

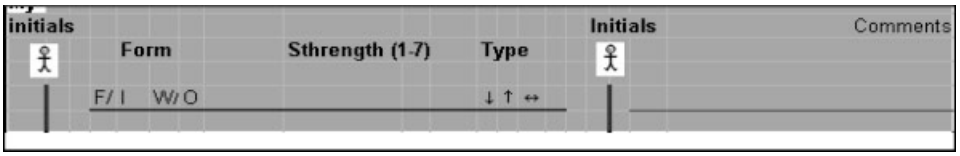

Figure 1. Graphical questionnaire

3. Visually analyse the results
4. Feedback the results to the group and validate the results
5. Evaluate the outcome

As we wanted to investigate to what extent and in which ways the company communicated and shared knowledge about software process improvement we identified the relevant group as all developers and all managers in SmallSoft.

Each respondent was asked to fill in a graphical questionnaire. They were asked to assess their communication on issues of improving software processes in the company during the last six months. They were asked to identify and characterise the communication as they recalled it. For each interaction they were asked to provide the name(s) of their colleagues in the interaction and to assess whether the communication had been (see Figure 1):

- Formal or informal by circling 'F' or 'I'.
- Written or oral by circling 'W' or 'O'.
- Downward, upward, or lateral influence process by circling one of the three arrows.
- Strength indicated by a number from 1 to 7; 1 meaning very low (e.g. receiving an email) and 7 meaning very high (e.g. collaboration or continuous dialogue).

The graphical survey questionnaire is shown in Figure 1. The accompanying instruction told the respondent to also register the initials of their communication partners and use a new line for every interaction. The instrument provides this pattern for all interactions.

This means that every reported line in a returned questionnaire is evidence of a relationship.

All questionnaire data were transferred directly to a spreadsheet. The format used in the spreadsheet was then loaded into NetDraw[1], which is a tool for social network analysis that can display graphs with actors as nodes and relationships as edges. Both nodes and edges can have attributes.

The tool offers various display features and analyses, which are performed automatically by the built-in graph algorithms. The tool was used to analyse and keep an overview of the data using graphical elements to visualise structures in the social networks (e.g. to select parts of the graph, show different attributes and weights, identify central actor, cut-point). The tool was also used to find and illustrate several network structures like centrality, components, k-cores, etc. We explain these concepts in more detail in subsection 4.2 where the actual analysis is described.

The analysis of the network data was iterative. The researchers were consistently looking for patterns in the network models, which confirmed or rejected working hypotheses about the company's SPI activities. That led to analytical insight, which in turn led the researchers to modified and new working hypotheses. The iterative analysis was temporarily stopped to validate the findings with two department managers. Their feedback was used to extend the iterative analysis. It also gave a detailed impression of which network models were relevant from a management point of view. The managers found that some of the models provided interesting research hypotheses and proper findings, but were not providing valuable managerial insight. The managers' feedback also led the researchers to prompt several developers to respond to the questionnaire to increase the data coverage.

The analysis ended with a second session with all three managers (for a description of the case company see subsection 4.1). This second session later moved on to a presentation of the findings and a rather detailed discussion of what to do about the network problems and SPI (see section 4.5).

4. SOCIAL NETWORKS IN SPI

The modelling of social networks followed the approach outlined in section 3. In this section we

[1] NetDraw: www.analytictech.com/Netdraw/netdraw.htm.

DOI: 10.1002/spip

first present the case background, then we present the models and analyses, and finally we report from the validation and managerial utilisation of the findings.

4.1. Case Background

SmallSoft is a small software company with two departments. The ERP Department develops a large ERP system and maintains it at a number of customer sites. The department's tasks are characterised by long-term and close contacts with a few large customers. The software developers have much domain knowledge within logistics in their customers' area. The head of this department is also responsible for the quality system and the company's ISO9000 certificate. He was also heading the SPI group. The Tailor-Made Department develops several tailored systems for many different customers. Their products range from traditional administrative systems to web portals. The application domains vary and the developers' primary expertise lies within software engineering and project management.

Previously, improvements in SmallSoft's software development were casual and spread through collaboration and informal contacts between colleagues. A few significant improvements had attracted management's attention and were turned into company-wide improvements. One company-wide improvement led to an internal software development project, which produced a support tool for tracking development tasks. Most improvements, however, were small and remained personal or local among a few colleagues.

When the research began, the company was introduced to a basic SPI approach and soon top management announced the slogan 'CMM level three – in three years.' A SPI group was formed and a developer from each of the departments was appointed to the group. The group took on the responsibility of assessing the current practices, planning improvement initiatives, and implementing these. Successful improvements were supposed to be added to the existing quality system. The manager of the ERP department later characterised this new set-up as a failure. His perception was that some developers felt pushed aside and that others stopped focusing on improvements waiting for the results from the SPI group. The SPI group on their part lacked time and resources and organised only one improvement initiative. At the same time the company experienced a market decline and subsequent low sales figures, and this led to a shift of focus away from improvement activities and towards sales activities and monthly sales figures.

Despite these setbacks, SmallSoft's management recognised the value of their previous improvements as vital for their business success and found it necessary to proceed. The two department managers' shared perception was that future improvements had to be rooted in a strategy that would provide faster feedback as well as visible and immediate benefits for the software developers. It was in this atmosphere that the analysis of social networks was initiated.

4.2. Social Network Analysis of SPI in SmallSoft

The analysis had the immediate purpose of understanding SmallSoft's social networks as a basis for managerial decisions about SPI. To that end we chose to visualize the models of communication networks that emerged from the data when displayed with NetDraw.

The most basic network model is shown in Figure 2. The node distribution feature in NetDraw provided its visual layout.

The model should be read in the following way. Circles represent developers; white circles denote that they are from the Tailor-Made department and grey circles show developers from the ERP department. Developers 29 and 30, depicted in black, are no longer in the company. Developers 4, 10–13 have not responded to the questionnaire and no others have reported communication with these developers. Triangles denote managers; manager 9 is the CEO. The number of respondents therefore is 23 of 28 staff, or 82%. When these models were used in SmallSoft the real names of developers and managers were shown.

The graph analyses follow Scott as well as Wasserman and Faust (1994); Scott (2000). The initial *graph analysis* was to look for components and central actors, because this provides a good overview to begin with. These first analyses have been performed on the network from Figure 2 where connections are un-directed and considered without their attributes and strengths.

The component analysis was on the basis of the formal concepts of component, cut-point, and

DOI: 10.1002/spip

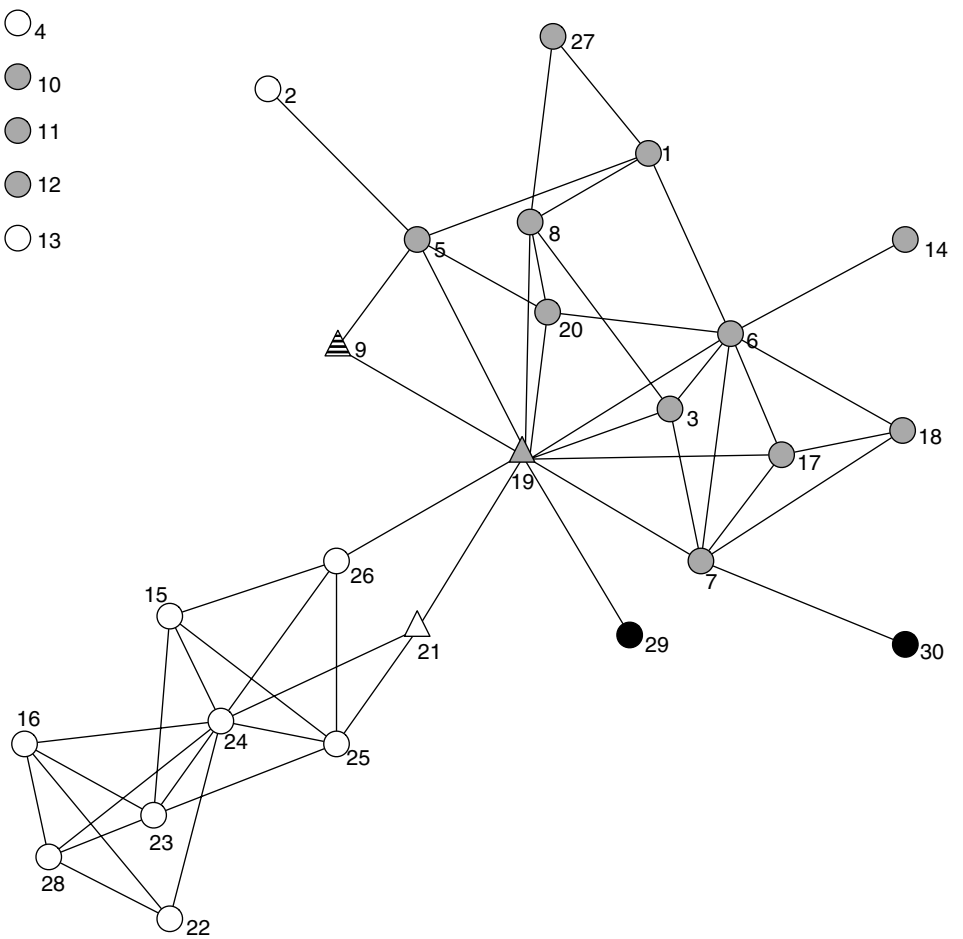

Figure 2. Basic network for communicating about SPI in SmallSoft

clique. A *component* in a social network is defined as a maximal connected subnet (Scott 2000, p. 101; Wasserman and Faust 1994, p. 109). The model of SmallSoft reveals that it consists of a single component because all developers and managers are related to at least one other except developers 4, 10–13 whom no one communicated with. These outliers as well as developers 29 and 30 were removed in subsequent analyses. A *cut-point* is a node whose removal would increase the number of components (Scott 2000, p. 107; Wasserman and Faust 1994, p. 112). In Figure 2, manager 19 is a cut-point who would split the company in two components, and developer 5 is a cut-point who would disconnect developer 2 from the main component. Similarly developer 6 is a cut-point who would disconnect developer 14. A *clique* is a subnet in which every possible pair of nodes is directly connected and the clique is not contained in any other clique (Scott 2000, p. 114–115; Wasserman and Faust 1994, p. 254). Counting only those subnets with more than three nodes the following subnets are cliques: (16, 22, 24, 28); (15, 24, 25, 26); (6, 7, 17, 18). Cliques are highly connected and as there are no cliques larger than four; they are mere small islands in SmallSoft.

The centrality analysis is on the basis of the formal concepts of degree centrality, closeness centrality, betweenness centrality and peak. The measure for *degree centrality* is defined as a node's number of direct relations (Scott 2000, p. 83 Wasserman and Faust 1994, p. 178). The more direct connections a node has the more central it is. The node with the highest degree centrality is: manager 19 with a degree of 10; see Table 1. The measure for *closeness*

DOI: 10.1002/spip

Table 1. Centrality measures

Rank	Degree	Closeness	Betweennes
1	M19 (10)	M19 (39)	M19 (144)
2	D24 (8)	D26 (44)	D24 (60)
3	D6 (8)	D21 (45)	D26 (57)
4	D7 (6)	D6 (48)	M21 (41)

centrality is defined as the sum of distance to all other nodes (Scott 2000, p. 86 Wasserman and Faust 1994, p. 184). The closer a node is to all other nodes, the more central it is; the closeness of nodes for all nodes in the SmallSoft network is as follows: manager 19 has the distance 39 to all other developers and managers in SmallSoft; see Table 1. The *betweenness centrality* of a node is defined as the proportion of node pairs that has the node on its path (Scott 2000, p. 87; Wasserman and Faust 1994, p. 191). NetDraw computed these to: manager 19 is on the path between 144 pairs; see Table 1. A *peak* is a node with a higher centrality measure than any other node which (s)he is directly connected to. Manager 19 is such a peak in all three centrality measures, while developer 24 is a peak on degree and betweenness centrality.

The qualitative analysis started with a working hypothesis that Smallsoft was a very informal organisation, but with strong monitoring by its two department managers. Furthermore, prior to the social network analysis the researchers held the perception that SmallSoft's management was in control and that all SPI activity had to be communicated through the managers. Consequently, the researchers did not assume that there were social subnets with the capacity nor the inclination to communicate independently on SPI and take action on SPI.

The subsequent analysis of the model led to the following results. As mentioned above, first of all it shows that five developers are completely outside all communication about SPI. Second, it identifies one main component containing both the ERP and Tailor-Made departments. Manager 19 is the most central actor as he is the actor ranking first on all three centrality measures. Manager 19 is also a peak as he is more central than any other actor he is connected to. This is not surprising as 19 is the manager of the ERP department and responsible for the quality system, the ISO9000 certificate, and also the SPI manager. He is connected to the top manager, CEO 9, and all connections between the ERP department and Tailor-Made department go through him.

Manager 21, the manager of the Tailor-Made department, is far less central and not a peak. He shares the linkage to the ERP department with developer 26. In the Tailor-Made department, developer 24 with a degree of 8 is the only peak and he is connected to everyone in the department. The path from any of the managers to any of their developers is less than or equal to two edges. In the ERP department this is due to the central role of the manager and in the Tailor-Made department it is due to developer 24.

Returning to the graph analysis for details we chose to go into depth with the cohesiveness of the company and the two main departments. To that end we modelled the k-cores network in NetDraw. A *k-core* is a maximal subnet in which each node is adjacent to at least *k* other nodes (Scott 2000, p. 110; Wasserman and Faust 1994, p. 266). The 3k-core displays the actors with a degree greater than or equal to three. Where the clique analysis shows the size of the most well-connected islands (none were larger than four), the k-core analysis shows the overall cohesiveness of SmallSoft.

The 3k-core model in Figure 3 is only slightly different from the basic model – only the CEO and three developers were removed. The 4k-core is much smaller as it removed five more developers. The 3k-core model shows the connectivity of the network and it is evident that the inner coherence of the company is relatively strong. It is noteworthy though that the CEO is not part of the inner network where SPI is addressed.

4.3. Analysis of the Relation Attributes

Figures 4 and 5 show the attributes of the communication for the main component. The four models in Figure 4 illustrate the differences between formal and informal communications and between written and oral communications. The communication is mostly informal and all actors are involved in informal communication. Formal communication is only found around the two peaks (manager 19 and developer 24) and between the two departments. Written communication has a stronger presence in the Tailor-Made department and around the manager of the ERP department. Oral communication is widespread and every actor participates in oral

DOI: 10.1002/spip

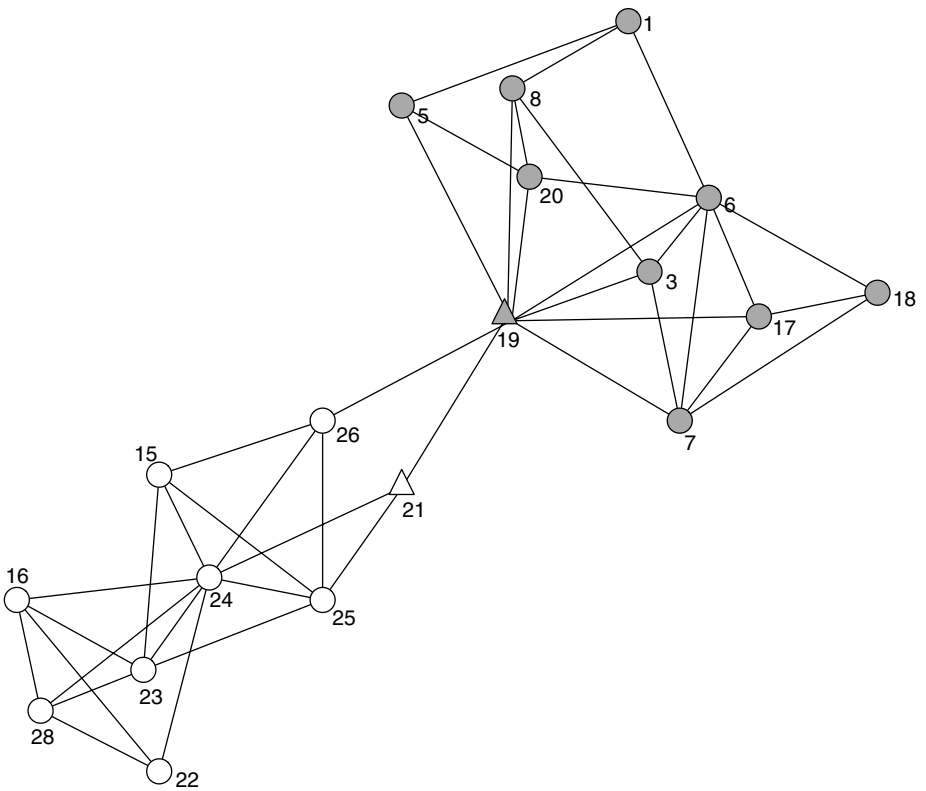

Figure 3. The 3k-core model for SmallSoft

communication. It is worth noticing that the communication between the departments is formal but oral.

The questionnaire asked all respondents to categorise the direction of the influence process in the communication as downward, upward or lateral. An influence process is an attempt by an originator to influence another individual or group to achieve goals (Nielsen and Ngwenyama 2002; Kotter 2003). Downward and upward influencing refers to relations in the formal hierarchy, while lateral refers to influencing between peers. Figure 5 shows all communication ties that have been reported as lateral by the respondents excluding all those reported as downward and upward. It is not surprising that the managers become isolated in the lateral network. What is interesting is that the lateral networks are present and involve all developers. What is also interesting, but not immediately apparent in Figure 5 is that there is communication among developers that has not been reported as lateral influencing. These are small in numbers, but they show that communication networks in SmallSoft are not entirely congruent with the hierarchy of formal authority.

Respondents had been asked to assess the strength of the communication on a scale from 1 to 7 with seven as the highest. Figure 6 shows a model of communication strength where respondents have reported strengths of 5–7 corresponding to the top half (not counting a strength of 4 which is the middle position). According to the weak-tie theory by Granovetter low strength is efficient for knowledge sharing because it bridges otherwise disconnected groups while high strength will lead to redundant information because group members know what the others know (Granovetter 1973). This has been further qualified in a more recent study in which it was shown that weak ties help search for useful knowledge, but transfer of complex knowledge requires strong ties (Hansen 1999).

Overall, the communication ties between the departments seem rather low. The model also shows that there are strong communication ties in the Tailor-Made department with degrees of 2 or more for most developers while the manager 21 has no

DOI: 10.1002/spip

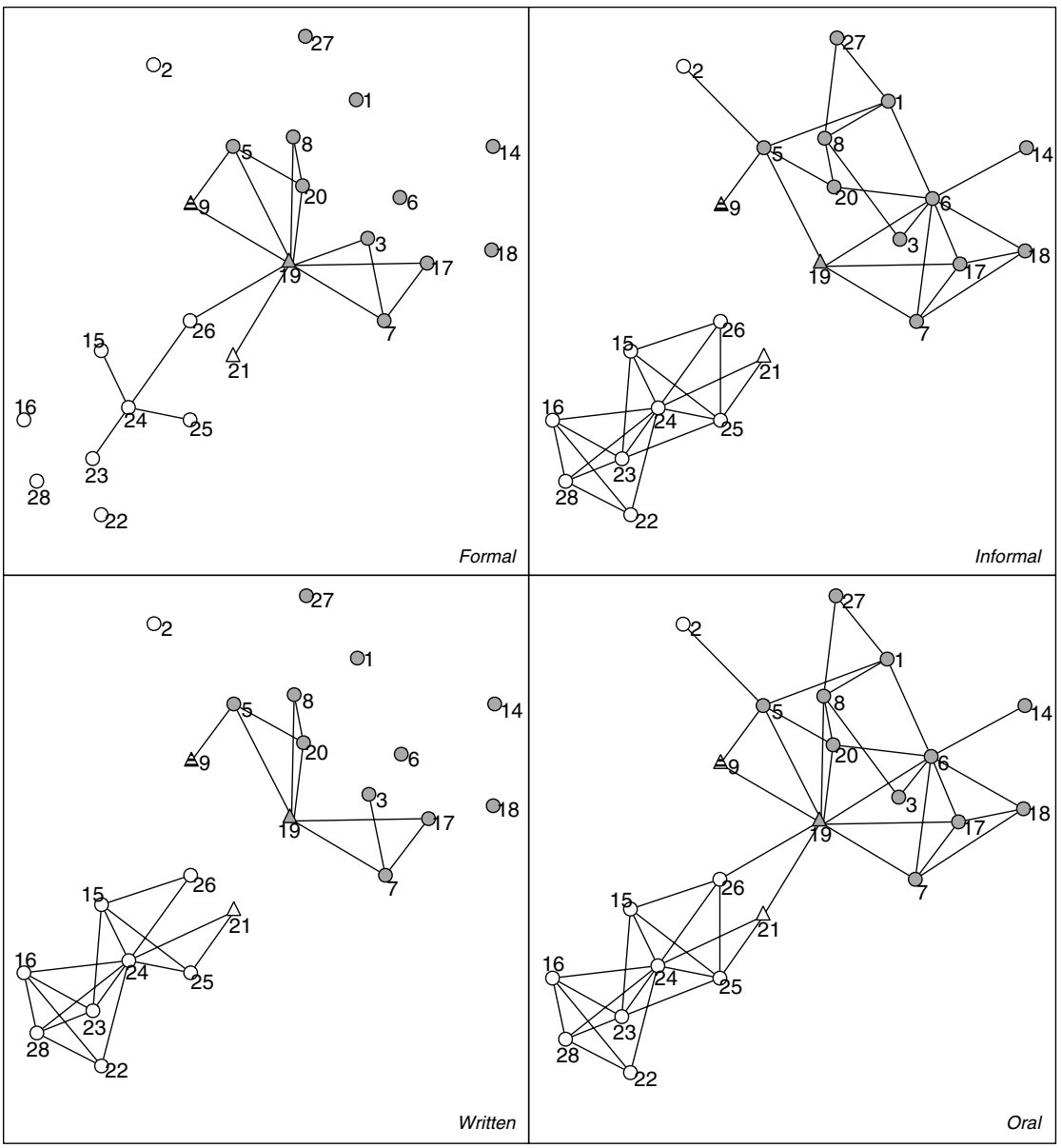

Figure 4. The models of attributes in the SmallSoft

strong communication with his own developers. For the ERP department the pattern is uneven as some are strongly connected while several are weakly connected. This would suggest on the basis of Hansen's theory that SmallSoft as a whole will have the social network to search for useful knowledge, but transfer of the more complex knowledge in software process improvement will happen within the departments.

4.4. Analysis Presented to Management

Overall the social networks show two departments with an informal, mostly oral and widespread interaction within the departments, but with sparse contact between departments and to top management. The ERP department has a central manager, 19, gate-keeping the department against all the other actors in the company in a more formal way than

DOI: 10.1002/spip

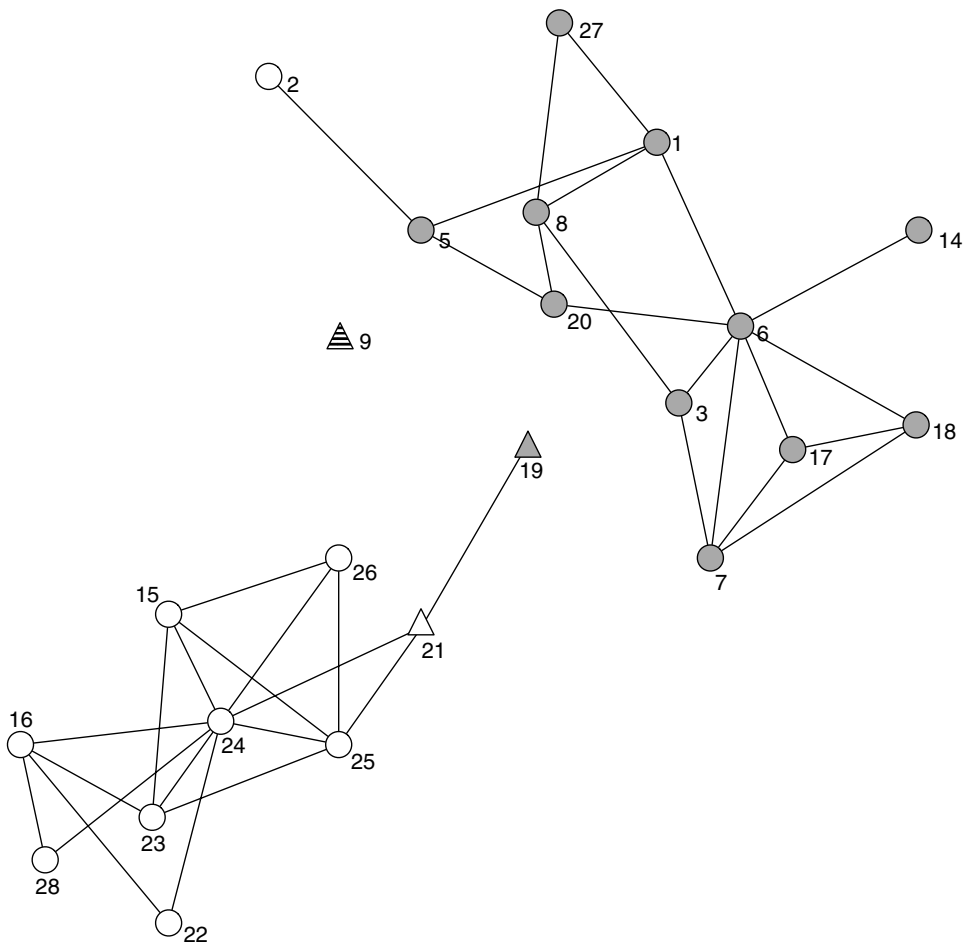

Figure 5. A model of lateral communication

usual in other parts of the company. He controls the communication on improvements both within his own department and at the management level. He is the only middle manager with contact to the top management.

The Tailor-Made department has a strong internal actor in developer 24 keeping the department connected and communicating intensively with many other developers. The manager of the department, 21, plays a lesser role in SPI as he has fewer ties and partakes only in lightweight communication. He only connects to the whole department through developers 24 and 25. This looks like a widespread delegation of responsibility for SPI.

Until the time of the analysis SmallSoft had followed a centralised and formal improvement strategy. There are considerable misfits between a centralised strategy and the underlying social networks. This may largely explain the failure of the improvement effort so far. The underlying social networks are uneven. In the ERP department, developers are unaccustomed to written communication. In both departments the sub networks are also lateral and thus less disposed to acting on formal management directive. In contrast, the applied centralised SPI strategy is management-driven and communicated in formal writing. The social network analysis thus leads to the conclusion that either the social networks must change or another strategy must be chosen. Social networks are emergent and cannot easily (if at all) be designed and it is thus more appropriate to change the strategy.

Faced with these alternatives Smallsoft's management wants to change to a decentralised strategy. They assess that this will suit the company better

DOI: 10.1002/spip

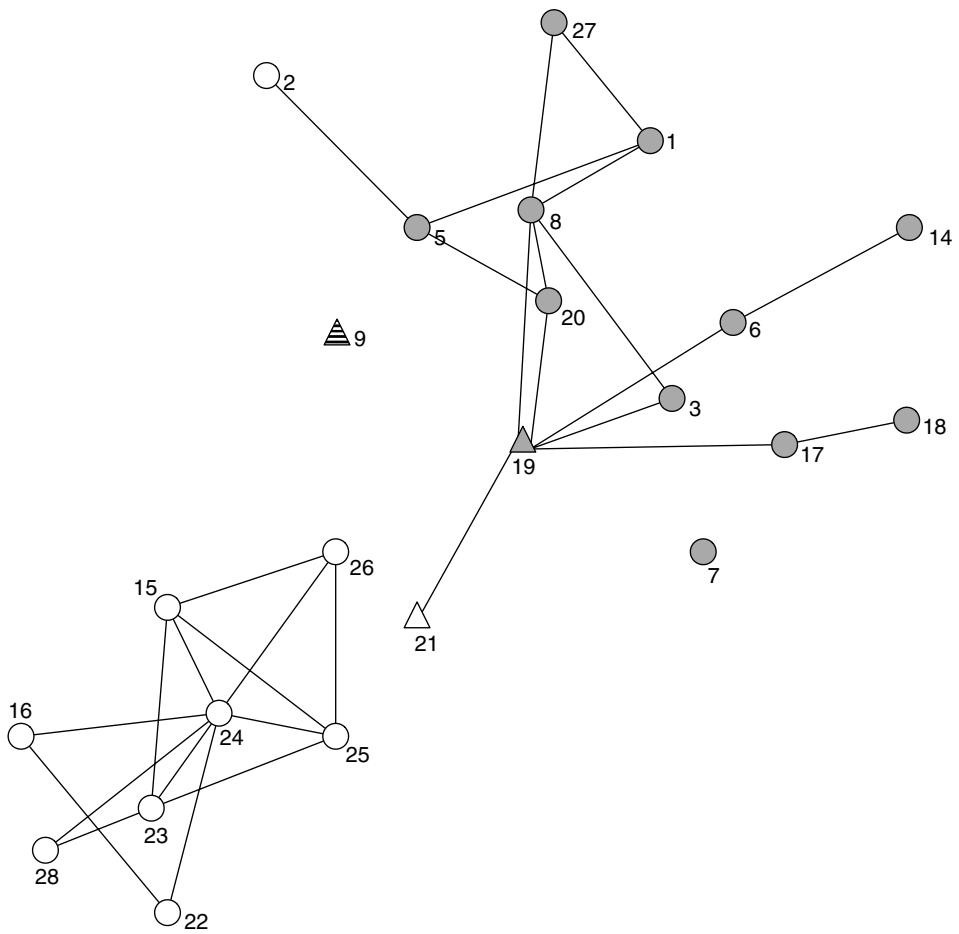

Figure 6. A model of strengths in communication; s ≥5

and will involve more developers. When embarking on a decentralised SPI strategy the researchers analysis led to the following:

- The remarkably weak ties between the two departments certainly hinder a central and cross-departmental SPI approach also in the future.
- A serious management commitment to SPI will be very difficult to exercise with so little communication on SPI involving the top manager; perhaps the lack of management involvement shows that SPI is not of strategic importance to the company's business strategy.
- Few improvements will spread easily from one department to the other. Closer ties need to be built between the two departments and at the level of the developers. If this is impossible or undesirable, the departments should be seen as separate social networks and independent SPI activities should be organised in each department deliberately decreasing dependency on cross-department knowledge sharing.
- Any SPI initiative in SmallSoft will benefit from at stronger collaboration among the managers and also involving the CEO.
- The ERP department could benefit from decentralisation, less formalisation and delegation of responsibilities. Manager 19 could very well be overloaded with responsibilities. If that is the case, he is a bottleneck that inhibits improvements and hinders knowledge sharing and communication in the department. Management commitment to SPI is on the basis of real involvement and focus.
- The network structures uncovered by the analyses do not hinder ideas and improvements being communicated amongst developers.

DOI: 10.1002/spip

These advices for the SPI managers are very much in line with (Cross and Parker 2004). They suggest that it is a management task to initiate, develop and maintain networks. They further propose that the internal network structure of a company should be aligned with its environment. For a small company like SmallSoft the environment for the ERP department changes only slowly, but it is vulnerable to a few missed sales opportunities. In the Tailor-Made department there are often changes that it should respond to. Management should hence consider whether they want to move developers between the departments.

4.5. Management Reflections

The researchers presented the models and their analysis to the three managers, who then discussed the findings derived from the network models. The managers' understanding of the current situation differed between the two middle managers on the one hand and the CEO on the other hand. The two department managers, 19 and 21, saw the SPI activities from within, and the CEO observed SPI from outside. Not surprisingly, the CEO disagreed with the finding that he was marginal to SPI in the company. All three managers recognized the problematic situation with the loose coupling between the two main departments. Though being in favour of a decentralised SPI strategy, they agreed that reducing collaboration between the departments would increase business risk and that it would be too costly and inefficient if each department organised its own independent improvement activities. Thus they looked for a solution that would build closer ties between developers across departments to achieve easier diffusion of improvements through informal and oral communication, i.e. gradually cultivate and improve the underlying social networks. On the other hand, the new solution should provide management with sufficient overview and insight so that improvement activities could receive more management attention and be supported by more formal communication from the management.

In line with (Cross and Parker 2004, p. 91) the researchers suggested that the company should introduce particular teams responsible for each of their improvement areas. This advice suggests that building bridges between individuals and between subnets can improve network relations. Building bridges means among other things (a) initiating relations and (b) develop professional and personal relations.

This suggestion was not immediately approved and decided at the meeting between the researchers and managers, but within a few months a few new teams were organised across the departments. They involved software developers with special interest in each improvement area. Management assigned 20–40 working hours per month to each team to support the activities. Like all other projects in the company the teams had to report their work and progress to management through monthly reports.

To support and coordinate the improvement teams a coordinating SPI group was formed. The new improvement initiative was kicked-off at a meeting for all teams where some of the social networks were presented to explain management's reasons for establishing the new teams.

5. DISCUSSION

The most significant finding of the social network analysis for SmallSoft was that there was already communication about SPI and that knowledge about software development was already shared. The managers already knew this in general, but they did not know the details. The network models showed many details which the managers were unaware of and which they had not addressed in their dealing with software process improvement.

The social network analyses proved valuable in SmallSoft. They provided the researchers with substantial insight for their action research endeavour. They were also useful for Smallsoft's managers in several ways:

- The network models provided images of the communication and knowledge sharing about SPI which the managers trusted as they had been involved in their validation.
- The models contained angles, pointers and clues that the managers had never thought about before. The SPI manager in particular genuinely found the models interesting as a kind of mirror in which he could now see his own organisation in a new light.
- The models had been useful in illustrating and explaining the findings from the researchers to the managers. They proved valuable as a starting

DOI: 10.1002/spip

point for the discussion of an appropriate strategy for SPI in the company, as they emphasized two major problems in the current situation that the managers could agree upon.
- The models were used as a basis for taking decisions about SPI and how to improve the underlying networks that had the potential of pushing the SPI effort forward. These decisions led to actions involving management more strongly and also to the creation of improvement teams across the departments.

It is evident from the SmallSoft case that communication and knowledge sharing in SPI is an integral part of SPI. Researchers and managers should acknowledge this and more attention should be paid to communication and knowledge sharing in SPI efforts. The research literature of small software companies has mostly been concerned with measuring process maturity and the problems that small companies have with maturity models and the CMMI in particular. There is research addressing the need for a closer look at knowledge management from a social perspective (see Kautz 2000; Kautz and Thaysen 2001); there is research addressing knowledge as a commodity to be stored in an experience base (see Conradi and Dingsøyr 2000; Rus and Lindvall 2002); there are reported experiences from building knowledge networks in software organisations (Kautz and Hansen 2008). There are however no reports where the underlying informal networks have been analysed as we have done here.

The SmallSoft action research study also shows an inherently difficult dilemma. On the one hand, the managers want to exercise leadership in SPI, and, on the other hand, there are underlying communities-of-practice in software development (Mathiassen 1998; Wenger and Snyder 2000), which cannot be designed or managed directly. The social network models show some of these communities-of-practice and the managers are with these in hand very aware that they cannot change the social networks as they change formal structures, responsibilities, and tasks. Perhaps, the way we have here used social network analysis points to a way in which the managers can nevertheless navigate and manoeuvre in this landscape.

Our study illustrates that modelling social networks is particularly relevant for understanding SPI activity in small companies. Small software companies are less likely to favour a formal, centralised SPI approach and SmallSoft is no exception here. It is thus reasonable to discuss how the lessons learned in our action research study concerning communication, knowledge sharing and social network analysis may be generalised.

Modelling social networks fits well with a low budget approach to SPI. It is cost-effective to analyse the underlying social networks that are an important part of the infrastructure for a more informal SPI approach in small companies as it has been performed in this study. Small companies lack the economical inclination to invest in a formal, rational, centralised infrastructure.

Modelling social networks enables small companies to discuss, to exploit the possibilities that already exist, and to focus on necessary improvements as a basis for SPI. We thus propose that the way we have modelled social networks can be well transferred to other, similar organisations. On the basis of the described experience, we suggest that it will work for small companies.

We can only speculate about whether it will also be feasible for large software organisations. It is likely that the visualisations from the tool we used will be less useful with more than a hundred developers and other software packages for social network analysis might be more useful.

However, irrespective of the size of company, our study shows that communication about SPI is also necessary in large organisations. Knowledge sharing happens in emergent communities-of-practice that can in part be uncovered with social network analysis. What we know so far is that in order to facilitate discussions that bring improvements forward, the network models must show the networks in a visual way that can be grasped by the involved actors without them being experts in social network analysis.

A CMM-driven strategy can be supplemented with social network analysis and that particular way of looking at the informal organisation. In a CMM-driven strategy the focus is on processes and much less on people (Aaen 2003). A social network analysis thus offers the opportunity to focus simultaneously on how people communicate and how this communication supports knowledge sharing and as such becomes a prerequisite for the organisational change.

The data collection methods and the analysis, and the use of models in discussions and reflections are not specific for SmallSoft. They are all transferable

DOI: 10.1002/spip

to other small software organisations. Thus, we claim generality for the applicability of modelling of social networks and for the performing of social network analyses. What cannot be transferred to other organisations are the specific models, the analyses of SmallSoft, and the specific outcomes of the discussions.

6. CONCLUSION

In this article we have reported on an action research study in a small software company. Data were collected through a case study and a questionnaire that was designed specifically to get information about communication patterns in the social networks involving software developers and their managers. The social network analysis was subsequently utilised in a management reflection to further change and leadership of SPI. Our findings can be summarised as follows:

1. Communication and knowledge sharing about software process improvement follow other patterns than official and bureaucratic channels. It is important to understand the structure of these informal communication networks as they can promote or hinder a particular improvement effort.
2. Such communication and knowledge sharing networks can be studied through social network analysis. Social network analysis and its accompanying tools and techniques offer several very useful analyses. The managers in the case company appreciated these findings and consequently acted upon them. In particular, they deliberately sought to remedy identified shortcomings in the network structures.
3. Social network analysis is very likely to be useful in other small organisations as data collection, visualisation, and the analysis techniques fit well with the particular challenges faced by small software companies, which want to engage in SPI.

The limitations of this action research study are related to its purpose, which has been to explore the usefulness of social network analysis for software process improvement. This exploratory study is only on the basis of a single action research study and that limits its generalisation. The study shows a high validity due its high response rate and due to the fact that findings were iteratively validated. Although the analyses performed with the software tool are reliable and it is therefore likely that a repetition would reach the same outcomes again, the results are only valid for SmallSoft. Hence we do not claim generality for the analyses.

In a continued effort to make social network analysis more useful for software process improvement we will undertake further research with more software companies and improve the questionnaire, as well as the particular analyses and their interpretations.

ACKNOWLEDGEMENT

We wish to thank the developers and managers of SmallSoft for access to the company. For valuable comments on earlier drafts of this article we thank Karlheinz Kautz and the anonymous reviewers.

REFERENCES

Aaen I. 2003. Software process improvement: Blueprints versus recipes. *IEEE Software* **20**(5): 86–93.

Ahern DM, Clouse A, Turner R. 2001. *CMMI Distilled: An Introduction to Multi-discipline Process Improvement*. Addison Wesley: Reading, MA.

Ahern DM, Clouse A, Turner R. 2003. *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*. Addison-Wesley: Reading, MA.

Baskerville R, Pries-Heje J. 1999. Knowledge capability and maturity in software management. *Data Base for Advances in Information Systems* **30**(2): 26–43.

Borgatti SP, Everett MG. 1992. Notions of position in social network analysis. *Sociological Methodology* **22**: 1–35.

Brodman JG, Johnson DL. 1994. What small businesses and small organizations say about the CMM. *16th International Conference on Software Engineering*, Sorrento, Italy, 331–340.

Cater-Steel AP. 2001. Process improvement in four small software companies. *Proceedings 2001 Australian Software Engineering Conference*. Canberra, IEEE Computer Society, 262–272.

Checkland P. 1991. From framework through experience to learning: The essential nature of action research. In *Information Systems Research: Contemporary Approaches and Emergent Traditions*, Nissen H-E, Klein HK, Hirschheim RA (eds). North-Holland: Amsterdam, 397–403.

DOI: 10.1002/spip

Chrissis MB, Konrad M, Shrum S. 2003. *CMMI: Guidelines for Process Integration and Product Improvement*. Boston, Addison Wesley.

Conradi R, Dingsøyr T. 2000. Software experience bases: A consolidated evaluation and status report. *Proceedings of the Second International Conference on Product Focused Software Process Improvement*. Berlin, Springer, 391–406.

Cross R, Parker A. 2004. *The Hidden Power of Social Networks: Understanding How Work Really Gets Done in Organizations*. Harward Business School Press: Boston, MA.

Ehrlich K, Valetto G, Helander M. 2007. Seeing inside: Using social network analysis to understand patterns of collaboration and coordination in global software teams.*Second IEEE International Conference on Global Software Engineering*, 297–298.

Granovetter MS. 1973. The strength of weak ties. *American Journal of Sociology* **78**(6): 1360.

Hansen MT. 1999. The Search-transfer problem: The role of weak ties in sharing knowledge across organization subunits. *Administrative Science Quarterly* **44**(1): 82–85.

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

Humphrey WS. 1992. Introduction to software process improvement. Technical report, CMU/SEI-92-TR-007. Software Engineering Institute, Carnegie-Mellon University: Pittsburgh, PA.

Humphrey WS. 2002. Three process perspectives: Organizations, teams, and people. *Annals of Software Engineering* **14**(1–4): 39–72.

Iversen J, Nielsen PA, Nørbjerg J. 1999. Situated assessment of problems in software development. *Data Base for Advances in Information Systems* **30**(2): 66–81.

Kautz KH. 1999. Making sense of measurement for small organizations. *IEEE Software* **16**(2): 14–20.

Kautz K. 2000. Software process improvement in very small enterprises: Does it pay off? *Software Process Improvement and Practice* **4**(4): 209–226, 10.1002/(SICI)1099–1670(199812)4 : 4<209::AIDSPIP105 >3.0.CO;2–8.

Kautz K, Larsen EAA. 1997. Diffusion theory and practice: Disseminating quality management and software process improvement innovations. *5th European Conference on Information Systems*, Cork, Ireland, 1–16.

Kautz K, Hansen BH. 2008. Mapping knowledge flows. In *Software Processes & Knowledge: Beyond Conventional Software Process Improvement*, Nielsen PA, Kautz K (eds). Software Innovation Publisher: Aalborg.

Kautz K, Thaysen K. 2001. Knowledge, learning and IT support in a small software company. *Journal of Knowledge Management* **5**(4): 349–357.

Kautz K, Westergaard H, Thaysen K. 2001. Understanding and changing software organisations: An exploration of four perspectives on software process improvement. *Scandinavian Journal of Information Systems* **13**: 31–49.

Kelly DP, Culleton B. 1999. Process improvement for small organizations. *Computer* **32**(10): 41–47.

Kotter JP. 2003. Power, dependence, and effective management. In *Organizational Influence Processes*, Porter LW, Angle HL, Allen RW (eds). M. E. Sharpe: New York, 127–141.

Long Y, Siau K. 2007. Social network structures in open source software development teams. *Journal of Database Management* **18**(2): 25–40.

Mathiassen L. 1998. Reflective systems development. *Scandinavian Journal of Information Systems* **10**: 67–117.

Mathiassen L. 2002. Collaborative practice research. *Information Technology & People* **15**(4): 321–345.

Mathiassen L, Nielsen PA, Pries-Heje J. 2002. Learning Spi in practice. In *Improving Software Organizations: From Principles to Practice*, Mathiassen L, Pries-Heje J, Ngwenyama O (eds). Addison-Wesley: Boston, MA, 3–21.

Mathiassen L, Pourkomeylian P. 2003. Managing knowledge in a software organization. *Journal of Knowledge Management* **7**(2): 63–80.

McFeeley B. 1996. IDEAL: A User's Guide for Software Process Improvement. Technical report, CMU/SEI-96-HB-001. Software Engineering Institute, Carnegie-Mellon University: Pittsburgh, PA.

McKay J, Marshall P. 2001. The dual imperatives of action research. *Information Technology & People* **14**(1): 46–59.

Müller C, Meuthrath B, Baumgraß A. 2008. Analyzing Wiki-based networks to improve knowledge processes in organizations. *Journal of Universal Computer Science* **14**(4): 526–545.

Nielsen PA, Iversen JH, Johansen J, Nielsen LB. 2002. The adolescent effort. In *Improving Software Organizations: From Principles to Practice*, Mathiassen L, Pries-Heje J, Ngwenyama O (eds). Addison-Wesley: Reading, MA.

Nielsen PA, Ngwenyama O. 2002. Organizational influence processes in software process improvement. In *European Conference on Information Systems*, Wrycza S, Kautz K (eds): ECIS Gdansk.

DOI: 10.1002/spip

Pries-Heje J, Pourkomeylian P. 2004. Beyond software process improvement: A case study on change and knowledge management. *EuroSPI Conference*. Trondheim, Norway, 2004.

Rus I, Lindvall M. 2002. Knowledge management in software engineering. *IEEE Software* **19**(3): 26–38.

Scott J. 2000. *Social Network Analysis: A Handbook*. Sage Publications: London.

Tichy NM, Tushman ML, Fombrun C. 1979. Social network analysis for organizations. *Academy of Management Review* **4**(4): 507–519.

Villalon J, Agustin GC, Gilabert TSF, Seco AD, Sanchez LG, Cota MP. 2002. Experiences in the application of software process improvement in SMES. *Software Quality Journal* **10**(3): 261–273.

Ward RP, Fayad ME, Laitinen M. 2001. Software process improvement in the small. *Communications of the ACM* **44**(4): 105–107.

Wasserman S, Faust K. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press: Cambridge.

Wenger EC, Snyder WM. 2000. Communities of Organizational Factor. *Harward Business Review* **78**(1): 139–145.

Yang HL, Tang JH. 2004. Team structure and team performance in IS development: a social network perspective. *Information & Management* **41**(3): 335–349.

DOI: 10.1002/spip