8 PROBLEM DIAGNOSIS SOFTWARE PROCESS IMPROVEMENT

Jakob Iversen
Peter Axel Nielsen
Aalborg University
Denmark

Jacob Nørbjerg
Technical University of Denmark
Denmark

Abstract

This paper addresses software process improvement. In particular it reports on action research undertaken to understand the problems with software processes of a large Danish company. It is argued that in order to understand what the specific problems are we may, on the one hand, rely on process models like CMM or Bootstrap. On the other hand, we may also see the specific and unique features of software processes in this company through what we call problem diagnosis. Problem diagnosis deals with eliciting problems perceived by software project managers and with forming commitment structures to enable process improvement to effectively take place. It is argued that problem diagnosis a useful approach and that it has advantages over model-based assessment.

Keywords: Software process improvement, CMM, Bootstrap, assessment, action research.

Introduction

Software process improvement (SPI) is an organizational change process, which introduces new and improved methods, techniques, and tools, as well as changes to work organization, attitudes, work and management practices at all levels of a software producing organization (Grady 1997; Humphrey 1989).

Within SPI well-known models of software process maturity are applied, such as the Software Engineering Institute's capability maturity model (CMM) (Paulk et al. 1993) and Bootstrap (Kuvaja et al. 1994). Such models guide the assessment process and help formulate an overall improvement strategy. But also the improvement project itself needs to be organized, the details of the improvements must be formulated, commitment to the improvement of actors at all levels must be negotiated, and the changes must be implemented and embedded in the organization.

Organizational change processes are generally hard to plan and implement, and there are many competing theories of organizational change (van de Ven and Poole 1995). Studies of SPI projects have given important insights into how to organize and implement SPI initiatives (Bush 1997; Culver-Lozo and Linnell 1996; Diaz and Sligo 1997; Goldenson and Herbsleb 1995; Grady 1997; Herbsleb et al. 1997; Sakamoto, Kishida and Nakakoji 1996; Snyder 1997). However, it is generally recognized among practitioners and researchers that we need more empirically-based knowledge about SPI.

The paper describes the development and use of what we have called the *problem diagnosis approach* during the early stages of an SPI project. A central purpose of problem diagnosis is to base the SPI project on the software developer's (in this case, the project managers') own perceptions of software process problems and suggestions for improvement rather than on the prescriptions and norms of a maturity model.

Problem diagnosis was developed during the initial phases of an SPI project in Brüel & Kjær, a medium-sized Danish electronics firm. There was considerable scepticism toward SPI when the project was initiated for a number of reasons (see the following section on SPI assessment and the later section on intervention). The scepticism was particularly strong among the powerful group of project managers and the problem diagnosis approach was introduced as part of a strategy to convert this scepticism into commitment and active participation in SPI. The results so far are positive: An interest in SPI has now been created among the project managers, and they have all chosen an SPI area to work with in their next project.

The present paper focuses on the development and use of the problem diagnosis approach in a specific context. Our experience with the approach does raise more fundamental and theoretical questions, however, regarding the use of model-based vs. more situational or contextual approaches (like problem diagnosis) to software process assessment.

The following section discusses SPI assessments and the reasons for the approach chosen at Brüel & Kjær, and the section after that describes our research method. The case study is presented in fourth section, and the lessons learned are presented in fifth section. The findings are then discussed and the paper is concluded.

SPI Assessments

An important characteristic of software process improvement is that the effort starts with an assessment of the organization's software process (Humphrey 1989; McFeeley 1996). The purpose of the assessment is to identify strengths and weaknesses of the current software process and outline an improvement strategy.

At Brüel & Kjær, a Bootstrap-based assessment had taken place about six months prior to the initiation of the SPI project, but since no follow-up action had been taken, the SPI group realized it needed to revive the SPI program somehow. Consequently, the assessment was revived, but with a new purpose and in with a new approach:

- identify the problems in Brüel & Kjær's software process as perceived by their software managers and developers,
- build an understanding of the causes of the problems,
- support the development of a specific improvement strategy, and
- involve the project managers actively in the assessment as well as in the subsequent implementation of improvement activities.

Furthermore, the new assessment should not require substantial resources and should be straightforward to carry out.

These objectives were first of all motivated by the specific circumstances at Brüel & Kjær. Especially the project managers' culture was powerful. That was seen as a threat to any initiative not explicitly supported by the project managers. But the objectives were also based on more general approaches to organizational learning and change processes. First of all, they were based on Schön's (1983) observation that organizational change must be based on an understanding of the situation at hand. They were also based on the lessons learned from studies of the introduction of new technology—especially information technology—in organizations which show how important it is to involve organizational actors in the identification of organizational problems and solutions (Greenbaum and Kyng 1991; Kyng 1991; Schuler and Namioka 1993).

In recent years, several assessment methods have been developed (Thomson and Mayhew 1997). The perhaps most well-known and common of these is the CMM-based appraisal for internal process improvement (CBA IPI) from the Software Engineering Institute (Dunaway and Masters 1996). Examples of other assessment methods are Bootstrap (Kuvaja et al. 1994), SPICE (Rout 1995), QBA (Arent and Iversen 1996), and progress assessment (Daskalantonakis 1994). The methods have all been used in real software process improvement projects (cf. Culver-Lozo and Linnell 1996; Diaz and Sligo 1997; Haase et al. 1994; Haley 1996; Herbsleb et al. 1997; Humphrey, Snyder and Willis 1991; Iversen et al. 1998; Sakamoto, Kishida and Nakakoji 1996).

All of these assessment methods are based on an underlying model of "good" software practice as well as a model of the improvement process itself. The CBA IPI and the QBA are, for example, both based on the CMM's approach to software process improvement where the assessment: (1) determines which of the capability maturity model's five levels the organization's software process belongs to, and (2) identifies the practices that must be improved in order to reach the next higher level. The SPI project at Brüel & Kjær could, however, not base the new assessment on any of these methods for a number of reasons:

- The risk of overemphasizing the model at the expense of the perceived problems with software processes.
- The risk that project managers would commit to desirable changes.
- The desire that the approach should be easy to learn and efficient.

These risks and desires may well be shared with many SPI initiatives.

The approach described in this paper was developed as an alternative to model-based assessments in order to meet the objectives summarized above.

Research Method

The research approach we have taken is action research. Action research is, according to Galliers (1992), a research approach where there is an attempt to simultaneously achieve practical value to the client organization and contribute to theoretical knowledge. A number of explanations of action research cover our approach (in particular Checkland 1981, 1991; Foster 1972; Hult and Lennung 1980; Rapoport 1970). In Checkland (1991, p. 400), the core of action research is presented as a group of researchers joining a real-world problem situation, taking part in improving the situation, and declaring a framework and a methodology so that reflection subsequently can yield research findings.

The researchers have visited the organization continuously for a year and a half, and the engagement will continue for another year and a half. Once a month (sometimes more) a full-day meeting has been held in the SPI group which includes three university researchers (the authors), two consultants, and four employees from Brüel & Kjær who are working with SPI in the organization. At the meetings, the researchers and the consultants act as support (both theoretical and practical) for Brüel & Kjær's SPI project. This support encompasses, but is not limited to, carrying out interviews, running meetings, providing advice, writing minutes.

The research process is documented in a number of ways. First, the researchers write field notes and diaries (see Jepsen, Mathiassen and Nielsen 1989 for a description of the approach). These are produced when the researchers find it necessary and relevant. Second, minutes from the SPI group's meetings are made and shared among the members of the group and the management of Brüel & Kjær. Third, meetings are recorded on tapes which are available for later reproduction of statements. Fourth, interviews have been conducted which are both recorded on tape and elaborate minutes have been written immediately afterwards (see also the subsection on current status). These interviews have been conducted in accordance with Patton's (1990) guidelines.

The Intervention at Brüel & Kjær

Brüel & Kjær has long been a leading manufacturer of high-precision instruments for sound and vibration, as well as condition monitoring. The company's products are used for many purposes in a diverse range of industries worldwide such as automotive, railway, power, telecommunication, petrochemical, government, education, and aerospace. The products are highly sophisticated, heavily based on embedded real-time

software, and cover the entire measurement chain from transducer to presentation and post-processing of results.

Brüel & Kjær was recently divided into two separate companies of radically different size. Brüel & Kjær Condition Monitoring Systems (CMS) is the smaller of the two with about 50 employees. It develops condition monitoring systems for rotating machines in nuclear and thermal power plants, oil platforms, refineries, etc. The development department in CMS consists of 20 employees, 10 of whom are software developers. Products are developed by project teams with up to eight people working for 0.5 to 2 years on a project.

Brüel & Kjær Sound & Vibration (S&V), has approximately 650 employees who develop and produce a complete range of sound and vibration measurement instruments and systems. The development department employs 100 people, 70 of whom are software developers. Approximately 15 to 20 development projects are conducted yearly by project teams of four to eight people working for 0.5 to 1.5 years on a project. Both CMS and S&V participate in the SPI project.

Brüel & Kjær has in recent years undergone a transition from a hardware to a (primarily) software developing organization. Originally, the company developed and manufactured measurement instruments based on mechanical and electronic components, but market demand for increasingly advanced and integrated measurement systems has changed this. Today, a typical Brüel & Kjær product consists of a PC with a number of DSP cards connected to one or more transducers that record the measurements input to the system. The system not only measures and displays measurement results, but is also able to guide the operator through a complicated measurement process, analyze the results, and produce a variety of reports. As a consequence, the bulk of development effort has shifted from hardware to software and Brüel & Kjær now considers itself to be an 80% software-developing organization.

Product and systems development in Brüel & Kjær is entirely project oriented and most projects are organized as so-called integrated projects (IPs). An IP includes hardware developers, software developers, sales and marketing people, etc. Consequently, there are different types of project managers: an IP project manager who is in charge of the entire project, and technical project managers for, respectively, the software and the hardware parts of a project. The project management style is mainly people-oriented, focusing on aspects such as education, training, and team-building, rather than the more technical project management issues, such as estimating, planning, tracking, etc.

The development process is documented in an ISO 9001 certified quality system, with separate hardware and software parts. The quality system is available on-line to all developers via the company intranet. Project managers, particularly IP and software project managers, complain however that the development model underlying the quality system is oriented toward hardware instead of software projects. Consequently, the project managers, who enjoy a high degree of autonomy, often “work around” the quality system and use alternative development models instead. Such deviations from the quality system are allowed if properly documented, but this consumes substantial resources and results in process variations across projects which are not always desirable.

SPI Background

The company has been involved in a number of SPI efforts in the past:

- The Prevention of Errors through Experience-driven Test Efforts (PET) (Vinter et al. 1996).
- Preventing Requirements Issues from becoming DEfects (PRIDE) (Vinter, Lausen and Pris-Heje 1998).
- A Practical Approach to Concurrent Engineering (PACE).
- Trial Application of a Reuse Software Application Library (TARSAL).
- Bootstrap Assessment (Hass, Johansen and Andersen 1997).

The first two of these initiatives were ESSI projects directed at locating and eliminating causes for defects in delivered products. The projects were based on detailed analysis and classification of 600 registered error reports and it was found that the largest source of defects in Brüel & Kjær's products were problems related to requirements and the second largest was lack of systematic unit testing. The findings resulted in a number of recommendations and suggestions to improve the requirements and testing processes. Some of the recommendations and suggestions have been tested successfully in pilot projects and will also form a basis for part of the SPI project.

The project managers from the pilot projects have adopted the approaches in subsequent projects, but the recommendations from PET and PRIDE are not generally disseminated in the organization. PACE and TARSAL were terminated prematurely because resources were reallocated to product-related projects.

In August 1996, Brüel & Kjær was assessed by means of the Bootstrap method as part of an effort to determine the overall process maturity in the Danish electronics industry (Hass, Johansen and Andersen 1997). Strong areas identified by the assessment included the ISO 9000 quality system, highly motivated developers, a high level of commitment to change, and a strong focus on human aspects while improvements were needed in the following areas:

- software development model
- software processes
- requirements specification
- project management
- module and integration testing
- configuration management

Overall, the assessment placed Brüel & Kjær at level 2.25 on the Bootstrap model's maturity scale. This indicates—on average—fairly disciplined processes at the project level, but with some weaknesses, as shown above, and a need to implement organization-wide process standards.

There was a high motivation for software process improvement immediately after the assessment. Other organizational changes meant, however, that the only specific action taken was the initiation of the SPI project about six months later and the motivation generated by the assessment had by then been lost.

Initiation of the SPI Project

Brüel & Kjær has established the following goals for the SPI effort:

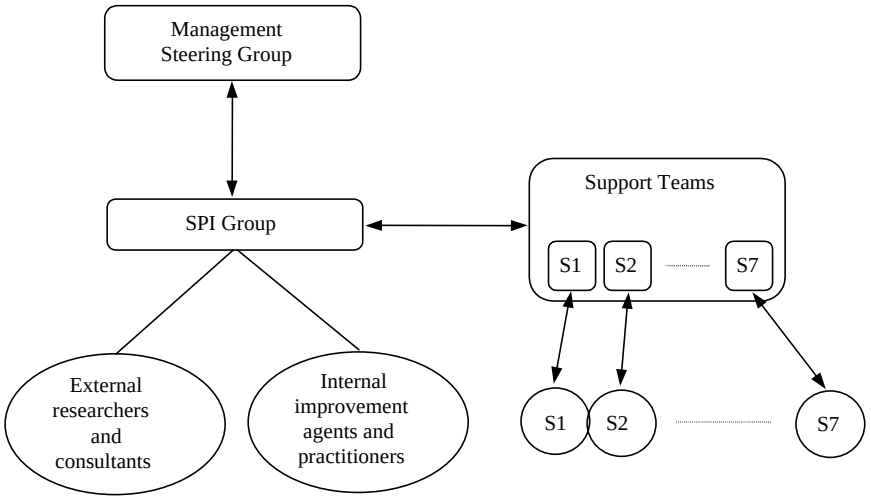

Figure 1. Organization of the SPI Project at Brüel & Kjær

- Better control of the growth in software development expenses.
- Standardize the development of software.
- Increase predictability in software development with regard to, for example, calendar time, resources, functionality, and quality.

The basic strategy of the SPI project in Brüel & Kjær is to focus on the project managers. It is believed that gaining their support is vitally important for the success of the process improvement project.

The project was initiated in March 1997 and is organized as shown in Figure 1. The Software Process Improvement Group (SPI group) consists of external researchers and consultants from a Danish research program on SPI, and improvement agents and practitioners from Brüel & Kjær. This group coordinates SPI activities and acts as catalyst for the improvement processes. The Management Steering Group is ultimately responsible for the project, including allocation of resources. The Steering Group consists of the technical directors from the two divisions: CMS and S&V.

Seven project managers of the projects (P1 through P7) are permanently associated with the SPI-effort. Improvements (e.g., the implementation of a new development model) are experimented with in one or more of the projects P1 to P7 and subsequently implemented in all projects in the organization. A support team with two or three members of the SPI group monitors and supports the improvement work in projects P1 to P7.

Diagnosis

Early in their work, the SPI group realized that it lacked detailed information about software processes and problems at Brüel & Kjær. Interesting information was available

from the PET and PRIDE projects, but these projects only addressed specific parts of the software process and were therefore not sufficient foundation for the formulation of an overall SPI strategy. The recommendations from the Bootstrap assessment were available, but the assessment report did not include a detailed and sufficiently rich description of the development process. The SPI group could thus not get a clear picture of the background for the recommendations, e.g., what exactly was the problem with the development model? One possible solution would be to conduct a new Bootstrap assessment. However, the SPI group opted not to do so because it feared that a new assessment, only six months after the previous, would do more harm than good and destroy whatever was left of the interest in SPI.

At the same time, it became clear that the SPI initiative had to be based on the active cooperation and enthusiasm of the project managers. Therefore, this group had to be involved in identifying problems and suggesting solutions. The SPI group therefore decided to develop and apply the diagnosing approach described here. This meant that the SPI group decided to

- conduct interviews with the managers of projects P1 to P7 about their views of strengths and weaknesses in Brüel & Kjær's software process,
- suggest improvement initiatives based on this information, and
- engage the project managers in the implementation of the initiatives.

The SPI group decided to use an interview guide approach where "topics and issues to be covered are specified in advance, in outline form, [but where] the interviewer decides sequence and wording of questions in the course of the interview" (Patton 1990, p. 288). This approach ensures that the interviews cover important issues and topics, but allows the interview to be matched to the topics and problems relevant to the particular project manager.

The interview guide approach puts a high demand on the interviewer's ability to phrase and rephrase questions "on the fly," and to follow-up on interesting themes without spending too much time on less relevant topics. The SPI group tried to reduce these risks by having more than one interviewer at each interview.

All interviews were tape-recorded to support the writing of summaries.

Interview Guide

The primary purpose of the interviews was to give the SPI group a better understanding of Brüel & Kjær's software development practices and the project managers' perceptions of strengths and weaknesses in the process. In addition to this, the SPI group wanted the project managers' reactions to the findings from the Bootstrap assessment as well as to a number of potential software process problems discussed in the group.

To meet these objectives, the interview guide contained the following five categories of questions:

- Description of a specific software development project.
- Critical events and problems in the project.
- Practices, problems, and suggestions for improvement within specific software process areas discussed in the SPI group.
- Other strengths and weaknesses in Brüel & Kjær's software development process.

- Practices, problems, and suggestions for improvement within the six improvement areas identified in the Bootstrap assessment.

The interview guide itself was developed in two stages. A first version with a list of cues and topics for the interviewers was developed before the first interview. This interview showed, however, that it was often difficult for the interviewers to phrase questions "on-the-fly." The second (and final) version of the guide therefore contained more explicitly phrased questions, but the interviewers could still choose to phrase questions suited for the interview. The final version of the guide also included an introductory section, explaining the framework and purpose of the interview, and a final section about the interviewee's impression of the interview as well as comments and suggestions within areas not covered by the interview.

The Interviews

There were two interviewers at each interview. A third member of the SPI group took notes and wrote a summary after the interview. The rest of the SPI group were observers. The interviewers had the initiative and were responsible for sufficient coverage of all topics in the guide, but the observers and the note taker were allowed to ask follow-up questions. This was an additional guarantee that all interesting issues would be sufficiently covered and the observers' and note taker's occasional intervention resulted in a more informal and relaxed atmosphere during the interviews—according to the interviewed project managers.

Usually, one interviewer was external (researcher or consultant) and one was internal (Brüel & Kjær SPI agent). This proved very useful. The internal's prior knowledge about the organization allowed identification and follow up on problematic issues or attempts to escape a question, but that same knowledge might on other occasions lead the internal to leave an issue too early, in which case the external would usually follow up.

Each interview more or less followed the structure in the guide. However, in the first interviews, it took more time than expected to work through the first categories of questions leaving very little time for the questions about the recommendations from the Bootstrap assessment. This was probably caused by a combination of insufficient interview skills and the externals members' lack of knowledge about Brüel & Kjær, which resulted in many clarifying questions about practices and products.

Analysis

The interviews gave the SPI group detailed information about Brüel & Kjær's software process and many suggestions from the project managers about how to improve the process. The group needed to systematize all the problems, issues, and opportunities raised during the interviews. The interviews took place over a long period of time and there was a considerable risk that important lessons would be lost, filtered, or at least seen through the lens of the last interviews.

The group discussed the lessons learned from the interviews on two occasions: after the first interview and after the last interview. Immediately after the first interview, a

discussion took place in the group. First each interviewer or observer in turn stated what they saw as the important lessons. Then these lessons were discussed, categorized, and documented in the minutes of the meeting.

After the last interview, the SPI group sought to reach a common understanding of what were the major lessons from each interview. Before the meeting, each member of the group had gone through the summaries of the interviews and had collected a list of five lessons to be learned from each interview. Each interview was then analyzed through the presentation and discussion of these lessons. There were different opinions of what would constitute a major lesson, but for each interview there was a certain sense of consolidation of the views held by the individual members of the SPI group. After going through all the interviews in this way, a conclusion spanning all interviews was discussed. No common conclusion was reached and the task of compiling the analysis in a short report was assigned to one of the external members of the group.

The first version of the report was a three and a half page summary of the major lessons across all interviews based on the discussion in the SPI group. The purpose was to provide a common conclusion that would serve as motivation to the project managers. The SPI group commented on the initial report. Six areas for improvement were identified based on the analysis and after discussion among the SPI group members. A seventh improvement area, software reuse, was added at the request of the internal group members and cannot easily be justified from the analysis. Hence, the final report contained the following improvement areas:

- descriptive process model
- risk management
- experiments with prototypes
- software requirements specification and requirements management
- project tracking and control
- project conclusion
- software reuse

The report was now a five and a half page document written clearly with the project managers as target audience. Each item on the list was explained briefly.

Workshop

To validate the findings in the report, it was presented to the project managers at a workshop. The workshop also served the purpose of getting the project managers' commitment to undertake specific improvement initiatives as part of their next software development project. Participating in the workshop were the project managers of P1 to P7, the SPI group, and the technical directors. All participants received a copy of the report a few days before the workshop.

The workshop had the following format:

1. Opening, presentation of participants, purpose of workshop.
2. Plenary presentation of main results of the analysis.
3. Group discussion of the validity of the analysis and negotiation of commitment to improvement actions.
4. Plenary discussion on commitment to improvement actions.

The purpose of the validation of the problem analysis was fulfilled in the sense that the project managers could recognize what they believed to be important and relevant problems. Some items were particularly useful, for example, descriptive process model. Other items were relevant but not significant, for example, project conclusion. For one item, software reuse, they did not see the problem and they believed the technology to be too immature to get any benefit from such an improvement initiative.

By the end of the workshop, considerable commitment had been established among the project managers. All but one project manager were committed to undertake process improvement integrated with their current or next software development project.

Current Status

The current status, 10 months after the workshop is as follows:

- Improvement projects within "development models," "requirements specifications," and "software reuse" have been initiated and integrated into software development projects.
- The technical director of S&V has assumed the responsibility for improving the area of "project tracking and control."
- The remaining improvement projects are waiting for the committed project manager to start a new project.

Considerable work has been put into the process of starting improvement projects. With minor variations, the process is as follows. First a half-day introduction to a specific improvement area (e.g., development models, requirements analysis) is held with the project managers from projects P1 to P7, the SPI group, and the technical directors. After the seminar, a separate meeting is held with the project group(s) that are going to work directly with that area. The agenda of the meeting is: (1) the project manager presents the software development task; (2) open discussion of how improvements may be undertaken in such a project; (3) agreement on purpose of the improvement, how it is integrated into the development project, and what support the project will need from the SPI group.

After this initial meeting, an internal contract is formed between the project manager and SPI group describing these conditions. A support team with members of the SPI group is assigned to follow and support the development project's improvement activities. The support team also records the development team's experiences with the changed practices. This experience will be valuable when implementing the improvements in the rest of the organization. The development team and the support team met frequently in the beginning and periodically during the project to ensure the success of the initiative. Additional training of the project group in specific techniques, for example, prototyping, the use of scenarios, and use cases, is provided as needed.

Discussion

The discussion focuses on the particular aspects of the applied problem diagnosis approach that distinguishes it from a model-based assessment.

Whose Problems?

In the early stages of process improvement, an SPI group is faced with the major issue of who shall decide what the important software process problems are. There are several legitimate answers: the members of the SPI group, external consultants, senior management, technical management, project managers, or the programmers.

In Brüel & Kjær, the SPI group chose to focus almost exclusively on the project managers. There were three reasons for this choice. First, the organizational culture of Brüel & Kjær rests on the assumption that the business is best based on strong project managers, who, when the company is in a crisis, will save it by producing extremely innovative products.

Second, the SPI group realized early that the support of the project managers was crucial for the success of the SPI initiative. At one of the first meetings in the SPI group, a technical director noted that the project managers would find ways to maintain existing practices if the software process improvement initiative was not sufficiently attractive to them. The project managers would do this with the best of intentions, namely that they were more capable to judge the needs of the organization than the SPI group. This and similar observations made the SPI group insist that the SPI strategy should be directed primarily at the project managers and be based on their perceptions and suggestions. For the SPI group, this was a simple means to avoid all sorts of traps and pitfalls toward fitting the SPI initiative to the organizational culture.

Third, problem diagnosis was also a very convenient strategy for the action researchers and the external consultants. The interviews with the project managers were an invaluable source of information about Brüel & Kjær's current software processes for the external partakers. The insights, the examples, the anecdotes, the concrete plans and actual actions taken in seven projects were revealed through the interviews. Without these, the external members of the SPI group could not have been useful when defining software process problems and devising an improvement strategy.

All this contributed to the decision to focus on the project managers' perceptions of process problems, thereby getting the SPI group in close contact with the core of the action in software processes and tackling the organization's real and serious software process problems.

At the interviews, the sole focus was on understanding the project managers' point of view. As an example, at one interview the project manager expressed the opinion that requirements specifications are elusive documents. In the particular project, the requirements were not settled until a few weeks before delivery. To him, a requirements specification was therefore not very useful for project planning and monitoring. This came as a surprise to many in the SPI group and a long discussion followed in order to understand precisely what the project manager would take a requirements specification to be, and what were the causes of the problems he had experienced. The interview transcript shows that this discussion took almost 15 minutes and that the interviewers explored the issue at length to get a precise phrasing of the project manager's viewpoint.

In general, problem diagnosis need not be directed solely at project managers. In other companies the organizational culture may well point in the direction of other relevant participants, such as senior management or programmers, but it remains important that the interviewed represent powerful groups in the organization and their viewpoints are essential for understanding the organization's software process problems.

Situated Problem Diagnosis

The problem diagnosis approach applied in Brüel & Kjær is oriented directly at understanding the software problems as perceived by the project managers. The problem diagnosis thus removes focus and attention from the established models of software processes, e.g., Bootstrap and CMM. The idea of maturity in these models has two sides. First, that there is a logical dependency between software process improvements and that an organization therefore must master the basic processes, e.g., project management or configuration management, before embarking on more advanced improvement initiatives such as standardization of methods or management by metrics. Second, that a maturity model will help assessors to focus on the basic software process problems—relative to the organization's level of capability—and ignore problems with more advanced processes.

We do not dislike maturity models, nor do we find them harmful. The problem diagnosis approach does not inhibit using the models. The point is, on the contrary, that we consider the viewpoints of the interviewees to be more important than the models. First, because we think there is a risk that a model-based assessment will ignore problems perceived by organizational actors or that it will result in formulation of software problems in a way that these actors cannot recognize and with which they cannot identify. In that sense, the models may ignore the organizational culture. Therefore, part of the intention with the problem diagnosis approach was to formulate results of the investigation in a language consistent with the organizational culture and in a way with which project managers could associate.

In this way we may say that problem diagnosis is situated. It is created within a specific context, a problem situation, in which the main actors (the project managers) are involved as problem owners and thus as clients for the SPI group's investigation.

This is in many ways in accordance with Schön's idea of reflection-in-action. In Schön's work, there is a distinction between technical rationality, on the one hand, and reflection-in-action on the other. From the point of view of technical rationality, a professional practice is seen as instrumental problem solving. This means that the starting point is a set of given objectives and the task of the professional practitioner is to choose the optimal means to realize these objectives. In doing so, the practitioner uses scientific knowledge to perform specific tasks and to select techniques that apply to different types of situations. Within this view, situations can be categorized scientifically.

From the point of view of reflection-in-action, situations are unique, complex, uncertain, and sometimes even discordant. Awareness of the uniqueness of the situation at hand is crucial and it is often only possible to see and comprehend smaller fragments of the situation. Within this view, knowledge and action are intrinsically related. Practitioners do research in the situations in which they find themselves. They reflect while acting.

To the point where software process improvement is intervention into the practices of the software developers, this is in all respects reflection-in-action. This is irrespective of whether the process is guided by consultants and researchers external to the organization or by an internal SPI group. Therefore, Schön's theory may provide us with some additional insight. From this, we may see software process improvement in the following light:

- Software process improvement situations are unique. Even software process problems are unique and they do not fall in pre-established pigeonholes. The definition of software problems is thus a central activity in SPI initiatives.
- Software process improvement is intervention. There is, on the one hand, a potential for looking at software process problems from outside the organizational culture and arriving at some alternative appreciation of the situation. On the other hand, there is the risk that the interventionists misinterpret the problem situation and influence the SPI initiative in a misfortunate direction.

Altogether, problem diagnosis is in accordance with reflection-in-action. This does not mean, though, that problem diagnosis starts from scratch. Rather, in problem diagnosis, the interview guide and the interviewers borrow ideas of software processes from the software process models. In our case, we relied indirectly on CMM and Bootstrap. Many of the interviewers had studied CMM and similar ideas of software engineering at length. Further, in the last half hour of each interview, the project manager was asked to reflect upon the Bootstrap assessment from the year before. During the interviews, however, little or no reference was made directly to the models; the discussions were focused on the project managers' perceptions.

Participatory Improvement

We may also view the problem diagnosis approach as a kind of participatory software process improvement. It is participatory in the sense that those who are to change working practice are also those involved in the problem diagnosis. This has a number of advantages known from other similar change processes. Among those are:

- Problem diagnosis takes a starting point where the people are (in our case, the project managers' perception of problems). It takes seriously the problem of commencing the change based on the organizational culture, its history, and its population, and their background and experience. Problem diagnosis leads to gradual rather than radical change.
- Problem diagnosis is more democratic than assessing a software organization against a maturity model.
- Based on problem diagnosis, the suggested areas for improvement have a fair chance of actually being improved.

Let's look at the latter in detail. Change is not going to happen by itself. There are all kinds of resistance to change, also legitimate resistance. In Brüel & Kjær, the analysis shows that the resistance would be considerable if the project managers were not involved. Without involvement of the project managers the only possible change would be for management to force a change of formal structures, for example, by changing the quality system. Such changes would not, however, be followed by a change in software process *practices* since this would depend on the active involvement and commitment of those who should change their practices.

In the terms of Mintzberg (1983), the only changes that management can carry out in a professional bureaucracy like Brüel & Kjær, without the involvement of the professional practitioners, are changes in the techno-structure. Such changes will not affect the professionals of the operating core, since the primary mechanism for

coordination (and thus change) is standardization of skills (i.e., education and indoctrination). Simply telling a professional to change his practice has no effect.

The limitation of problem diagnosis is, however, that disagreements and conflicts are not easily handled. Disagreements may easily arise between different perceptions of problems. We did not experience this, since we found surprisingly consistent explanations of software process problems. This may in hindsight be explained by the fact that the involved project managers to a large extent had common interests and goals with the SPI project. The question we are left with is whether problem diagnosis can be used if there are more manifest or dominant disagreements between project managers. We have little experience with this since problem diagnosis was designed to embrace and cater for disagreements on a small scale. Disagreements at this level are dealt with through dialogue; first the dialogue between each project manager and the research team during interviews, and second the dialogue among the project managers and between the group of project managers and the SPI group at the workshop. This will cater for disagreements that can effectively be resolved or handled otherwise through dialogue. Other, more fundamental conflicts, must be resolved through negotiations, for example, between top management and the project managers as well as between project managers.

Another question is whether the lack of serious disagreements during the problem diagnosis at Brüel & Kjær was caused by the deliberate exclusion of other groups, particularly top management and programmers, from the process. On one occasion when a programmer participated in the interview, some conflicting views between the programmer and the project manager were revealed but we did not analyze this aspect closer, both because we did not have sufficient time and resources to do so, and because we wanted to maintain our focus on the project managers. This is of course a limitation of our trial of the problem diagnosis approach.

Lessons Learned

Based on our experience from Brüel & Kjær and on our discussion in the previous section, we find that our approach is more generally applicable. The lessons we have learned about a more general problem diagnosis approach are summarized in the following. The lessons are summaries of points made in earlier sections.

Lesson 1

Problem diagnosis should be used when there is a need to involve project managers (or other essential project team members) in software process improvement. Problem diagnosis should furthermore be used when there is a need to create commitment from project managers (or other essential project team members) in actual changes of software process practices.

This lesson further emphasizes that the problem diagnosing approach is directed particularly at the project level of the organization. Problem diagnosis may also prove to be useful at an overall organizational level, e.g., where common processes are to be installed. It is still too early to conclude this directly based on our experience, but the necessity for committed actors makes it likely.

Lesson 2

Problem diagnosis has the following advantages: (1) it is easy to understand and plan, (2) it focuses attention on perceived problems, (3i) it is efficient as well as effective, and (4) actors are involved and commitment is created.

Problem diagnosis appears to be a fruitful approach to SPI as a supplement or replacement for model-based approaches, especially in situations where an approach requiring fewer training and resources is needed or where the findings and recommendations of a (pure) model-based approach are not shared by organizational actors.

Lesson 3

Problem diagnosis has the following disadvantages: (1) interviewers must have a firm background in software engineering and software process improvement, (2) the approach is only sufficiently efficient if carefully planned in terms of writing notes and summaries, and having a manageable number of interviewees, (3) it is inefficient to interview all organizational actors who need to be involved in discussions and decisions, and (4) it is difficult to select interviewees.

Problem diagnosis is not an easy or straightforward approach. It should be seen as part of a general SPI initiative and be planned and implemented according to the priorities and goals of that approach.

Lesson 4

Problem diagnosis requires the following conditions: (1) interviewers skilled in interviewing, (2) a well-structured interview guide, (3) plans and commitment for processing and interpreting interview data, and (4) SPI group members are involved in interviews and workshop as well as interpretation and decision-making.

As part of an organization's general SPI initiative, problem diagnosis should be planned and performed by the SPI group. This presupposes, however, that the members of this group receive the necessary training in planning, conducting, and analyzing guide-based interviews.

Lesson 5

Organizational politics and pragmatics play a role in problem diagnosis. It is thus advantageous for the SPI effort to involve organizational actors who are able to influence attitudes and decisions.

Problem diagnosis may well bring latent organizational conflicts and differing interests out in the open. The approach does not in itself contain remedies for this, and it should not. These issues can only be dealt with by inviting relevant organizational actors to participate in the process.

Conclusions

This article describes the development and application of the problem diagnosing approach during the initial phases of an industrial SPI project. Problem diagnosis aims to build a situated understanding of an organization's software process problems. We argue that this enables an SPI group to base software process improvement on the perceptions and priorities of the software practitioners, thereby increasing the commitment to SPI among those who are going to change their software practices.

Problem diagnosis grew out of one SPI project and appears to have been a successful answer to problems encountered here. We believe, however, that a pragmatic or situated approach to SPI such as the one presented here can be a useful alternative or supplement to traditional model-based approaches, not only in SPI projects similar to the Brüel & Kjær case, but also as a more general strategy toward actively involving organizational actors in SPI.

However, further research is needed in order to validate problem diagnosis and make the approach generally applicable in SPI.

Among the relevant research questions are:

Does problem diagnosis create sufficient commitment for SPI?

Problem diagnosis has created commitment toward SPI among Brüel & Kjær's project managers and several improvement projects are underway. Further studies are needed to determine whether this commitment is strong enough to sustain and anchor process improvement in the long run.

Does problem diagnosis uncover an organization's "real" software problems?

Problem diagnosis has identified a number of software process related problems in Brüel & Kjær. To determine whether these are the "real" or most important problems, further studies and comparisons with other approaches to software process assessment are required.

Can problem diagnosis cope with conflicts and organizational politics?

Problem diagnosis is designed to formulate an improvement strategy based on the views and perceptions of organizational actors, and does not contain any mechanisms, such as CMM'S five-level model, to prioritize findings. The approach may therefore be difficult to use in some situations. It remains to be seen whether this makes the approach vulnerable to manipulation by powerful organizational actors or a valuable way to reveal and make explicit the social and political dimensions of software process improvement.

Is problem diagnosis applicable in other settings?

Problem diagnosis is still at an experimental stage and it has only been tried in one organization. The approach needs to be further validated, tested, and refined through application in other organizational settings.

Is a situated approach to SPI an alternative or a supplement to "traditional" model-based approaches?

It appears that both situated and model-based approaches to SPI have some limitations. A situated approach like problem diagnosis may identify only the immediate problems perceived by the group of respondents (e.g., the project managers), and it provides no mechanism for prioritizing findings. A model-based approach, on the other hand, may identify diversions from the model, rather than "real" software process problems. It is therefore reasonable to assume that it would be feasible to combine the two approaches in some way, as we have done to some degree in this case. Research into

this issue would involve studies and discussions of the theoretical foundations and assumptions behind the two approaches, as well as empirical studies of SPI. A possible approach would be to view the model-based approaches as parts of an overarching theory (Schön 1983) used in a situated approach.

Acknowledgment

This research has been financially supported by The Danish National Centre for IT Research and The Danish Agency for Development of Industry and Trade. The research would not have been possible without the active participation and cooperation of Brüel & Kjær.

References

Arent, J., and Iversen, J. Udvikling af metode til modenhedsmålinger 1 software-organisationer baseret på (The Capability Maturity Model (Development of a CMM Based Method for Maturity Evaluations—in Danish)]. M.Sc. Thesis, University of Aalborg, Aalborg, Denmark, 1996.

Bush, M. "Misunderstanding the SEPG's role in the CMM: Truth and Consequences," in *Proceedings of Quality Week 1997*, San Fransisco, CA, 1997.

Checkland, P. *Systems Thinking, Systems Practice*. Wiley, Chichester, England,1981.

Checkland, P. "From Framework Through Experience to Learning: The Essential Nature of Action Research," in *Information Systems Research: Contemporary Approaches and Emergent Traditions*, H.-E. Nissen, et al. (eds.), North-Holland, Amsterdam, 1991, pp. 397-403.

Culver-Lozo, K., and Linnell, S. L. "The Impact of Software Process Improvement on Interval Cost and Quality: Strategies for Success," in *Proceedings of Software Process Improvement 1996*, Brighton, England, 1996.

Daskalantonakis, M. K. "Achieving Higher SEI Levels," *IEEE Software* (11:4), 1994, pp. 17-24.

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

Dunaway, D. K.,and Masters, S. "CMM-Based Appraisal for Internal Process Improvement (CBA IPI): Method Description," Technical Report: CMU/SEI-96-TR-007, Software Engineering Institute, Pittsburgh, PA, 1996.

Foster, M. "An Introduction to the Theory and Practice of Action Research in Work Organizations," *Human Relations* (25:6, 1972, pp. 529-556.

Galliers, R. "Choosing Information Systems Research Approaches," in *Information Systems Research: Issues, Methods and Practical Guidelines*, R. Galliers (ed.), Blackwell Scientific Publications, Oxford, England, 1992.

Goldenson, D. R., and Herbsleb, J. D. "After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors that Influence Success," Technical Report: CMU/SEI-95-TR-009, Software Engineering Institute, Pittsburgh, PA, 1995.

Grady, R. B. *Successful Software Process Improvement*, Prentice-Hall, Upper Saddle River, NJ, 1997.

Greenbaum, J., and Kyng, M. (eds.). *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991.

Haase, V.; Messnarz, R.; Koch, G.; Kugler, H. J.; and Decrinis, P. "Bootstrap: Fine-tuning Process Assessment," *IEEE Software* (11:4), 1994, pp. 25-35.

Haley, T. J. "Software Process Improvement at Raytheon," *IEEE Software* (13:6), 1996, pp. 33-41.

Hass, A. M. J.; Johansen, J.; and Andersen, O. "Software Development in the Electronics Industry" (in Danish), Technical Report: D-260, DELTA, Hørsholm, Denmark, 1997.

Herbsleb, J.; Zubrow, D.; Goldenson, D.; Hayes, W.; and Paulk, M. "Software Quality and the Capability Maturity Model," *Communications of the ACM* (40:6), 1997, pp. 31-40.

Hult, M., and Lennung, S.-Å. "Towards a Definition of Action Research: A Note and Bibliography," *Journal of Management Studies* (17:2), 1980, pp. 242-250.

Humphrey, W. S. *Managing the Software Process*. Addison-Wesley, Reading, PA, 1989.

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

Iversen, J.; Johansen, J.; Nielsen, P. A.; and Pries-Heje, J. "Combining Quantitative and Qualitative Assessment Methods in Software Process Improvement," in *Proceedings of the Sixth European Conference on Information Systems 1998*. Aix-en-Provence, France, June 1998.

Jepsen, L. O.; Mathiassen, L.; and Nielsen, P. A. "Back to Thinking Mode: Diaries for the Management of Information Systems Development Projects," *Behaviour and Information Technology* (8:3), 1989, pp. 207-217.

Kuvaja, P.; Similä, J.; Krzanik, L.; Bicego, A.; Saukkonen, S.; and Koch, G. *Software Process Assessment and Improvement—The BOOTSTRAP Approach*. Blackwell, London, 1994.

Kyng, M. "Designing for Cooperation: Cooperating in Design," *Communications of the ACM* (34:12), 1991, pp. 65-73.

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

Mintzberg, H. Structure in Fives: Designing Effective Organizations, Prentice-Hall International, London, 1983.

Patton, M. Q. *Qualitative Evaluation and Research Methods*, 2nd ed., Sage Publications, New York, 1990.

Paulk, M. C.; Curtis, B.; Chrissis, M. B.; and Weber, C. V. "Capability Maturity Model for Software, Version 1.1," Technical Report: 93-TR-024. Software Engineering Institute, Pittsburgh, PA, 1993.

Rapoport, R. N. "Three Dilemmas in Action Research," *Human Relations* (23:6), 1970, pp. 499-513.

Rout, T. P. "SPICE: A Framework for Software Process Assessment," *Software Process: Improvement and Practice* (1:Pilot Issue), 1995, pp. 57-66.

Sakamoto, K.; Kishida, K.; and Nakakoji, K. "Cultural Adaptation of the CMM: A Case Study of a Software Engineering Process Group in a Japanese Manufacturing Factory," in *Software Process*, A. Fuggetta and A. Wolf (eds.), Wiley and Sons, New York, 1996, pp. 137-154.

Schuler, D., and Namioka, A. (eds.). *Participatory Design. Principles and Practices*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1993.

Schön, D. *The Reflective Practitioner. How Professionals Think in Action*, Basic Books, New York, 1983.

Snyder, T. R. "SEI/IEEE Software Process Achievement Award," in *Proceedings of Software Process Improvement 1997*, Barcelona, Spain, 1997.

Thomson, H. E., and Mayhew, P. "Approaches to Software Process Improvement," *Software Process. Improvement and Practice* (3:1), 1997, pp. 3-17.

van de Ven, A., and Poole, M. S. "Explaining Development and Change in Organizations," *Academy of Management Review* (20:3), 1995, pp. 510-540.

Vinter, O.; Poulsen, P.-M.; Thomsen, J. M.; and Nissen, K. "The Prevention of Errors through Experience-driven Test Efforts (PET)," ESSI Project No. 10438. Final Report. http://www.esi.es/ESSI/All/10438, 1996.

Vinter, O.; Lauesen, S.; and Pries-Heje, J. "A Methodology for Preventing Requirements Issues from Becoming Defects (PRIDE)," ESSI Project No. 21167. Final Report. http://www.esi.es/ESSI/All/21167 1998.

About the Authors

Jakob Iversen has a M.Sc. in software engineering from Aalborg University. He is currently a Ph.D. student at the Computer Science Department at Aalborg University. The thesis working title is "Data-driven Intervention in Software Process Improvement." E-mail: iversen@cs.auc.dk

Peter Axel Nielsen is currently associate professor in Information Systems at the Department of Computer Science at Aalborg University. Over the past years he has been engaged in understanding the use of methodologies in information systems development. His research interests include analysis and design techniques, object-orientation, domain modelling, and software process improvement. He is co-author of a book on object-oriented analysis and design. E-mail: pan@cs.auc.dk

Jacob Nørbjerg has a M.Sc. and a Ph.D. in computer science from the University of Copenhagen. His research interests include systems development as a work process and the design, construction, and use of information systems in organizations, particularly distributed organizations. Prior to his present assignment at the Center for Tele-Information, he worked for two years as assistant professor at the Department of Computer Science, University of Copenhagen, where he participated in an international research project on CSCW and software engineering. E-mail: jn@tele.dtu.dk