Guidelines on Conducting Software Process Improvement Studies in Industry

Erik Arisholm[a], Bente Anda[a], Magne Jørgensen[a,b] and Dag I.K. Sjøberg[a]
{erikar,bentea,magnej,dagsj}@ifi.uio.no
Department of Informatics, University of Oslo[a]
Storebrand ASA, Norway[b]

Abstract

Initiating and conducting empirical case studies in industry are non-trivial tasks. There are many pitfalls, and many researchers have failed in their research. For example, there is the risk that the results are not particularly useful for the IT organization. As a result, the IT organization will be reluctant to cooperate with the researchers again. The purpose of this paper is to share the authors' experience and lessons learnt in conducting empirical case studies within software process improvement in cooperation with industry. In particular, we focus on the potential conflicts between the organizational goals and the research goals. Our experiences are summarized in terms of guidelines that we believe are useful for researchers in this area.

Keywords: IS Research Issues, Software Engineering, Case Study

BRT Keywords: IB03, F, AI0102

Introduction

Research on software process improvement can reuse many guidelines and methods from the natural sciences, but is in some aspects different. It is more a "research of the artificial" (Simon 1969), i.e., a study of the interface between an artifact (human created thing - software) and its environment (programmers, users, organization, etc.). In the IT industry, controlled experiments cannot easily be carried out; case studies are frequently the only practical way to get knowledge enabling a systematic improvement of the software development processes. However, initiating and conducting such case studies in industry are non-trivial tasks. There are many pitfalls, and many researchers have failed in their research. Although there exist theoretical frameworks for how to carry out process improvement studies (Kitchenham 1990, Zelkowitz and Wallace 1998), we believe that there is a need for sharing practical case study research experience within the software research community.

This paper presents the experiences from six case studies with emphasis on the potential conflicts between the organizational goals and the research goals.

Our experiences are summarized in terms of practical guidelines, for example:

- Extra effort is required to establish confidence and a positive cooperative environment if there are large geographic distances, many organizational layers, etc.
- Sensitive information must be handled confidentially.
- Intermediate results should be presented frequently to ensure that the organization

understands the motivation, goals and potential organizational benefits of the research.

The remainder of this paper is organized as follows. The next section gives a brief overview of research methodologies applicable to software process improvement. Then we describe the case studies with emphasis on cooperation challenges and experiences and the case-study research guidelines generalized from these experiences. Conclusions are stated in the last section.

Research Methods in Software Process Improvement

Researchers use an organization as a laboratory for performing research. They promote and support the organizations' change processes and participate in the process evaluation. Software process improvement may consist of the following activities:

- Describe and formalize the software process.
- Identify and characterize the technology (methods and tools) under study and its environment.
- Evaluate the software process or the use of existing methods or tools.
- Identify new requirements for the software process, methods or tools.
- Improve existing software process, methods or tools.

The following sections outline research approaches used in software process improvement activities. Focus is primarily on the case study as this is the method used in the studies described in the next section. The action research approach is included because research in software process improvement is concerned with promoting and supporting change. Case studies in software process improvement will therefore often include aspects of action research.

Approaches to Case Study Research

A *case study* allows in-depth understanding of one particular case or development project. The case study research is considered particularly suited when the object is the study of information systems in organizations (Braa and Vidgen 1999). Reality can be captured in detail and many variables can be analyzed, but there are problems with generalization due to lack of control, different interpretations and possible effects caused by the intervention of the researcher. Several case studies will together form a broader picture from which both researchers and industry can draw knowledge.

A variant of the case study is the multi-case where a random selection is drawn from one type of project. This represents a larger population than one single case, and it enables the use of statistical methods to test hypotheses. A potential weakness is that it may be difficult to ensure that the individual projects are "similar" enough.

Action Research

A case study where there is (almost) no intervention from the researcher can be classified as an *observation*. A case study characterized by planned and deliberate changes to the organization under study can be classified as *action researc*h (Argyris and Schön 1991). Action research aims to (1) contribute to solving practical problems of an organization and (2) achieve scientific results by joint collaboration between an organization and the researcher (Rapoport 1970).

Controlled Experiments

If sufficient resources are available, and when it is possible to control the relevant variables in the environment, a (controlled) experiment is the best method to ensure a high degree of precision and generality. This approach is characterized by the manipulation of independent variables that are believed to cause changes in the process, and a systematic assessment of the dependent variables.

In software process improvement, we would prefer to conduct experiments in real organizations (field experiments). Unfortunately, because it is difficult to find the needed resources and collaborating organizations, experiments are often conducted in an artificial setting (laboratory experiments). Moreover, the often limited prior knowledge about the object under study and its context makes it difficult to identify the independent variables.

Case Studies in Software Process Improvement

The following sections describe the experiences from six case studies, with emphasis on the potential conflicts between the organizational goals and the research goals.

An overview of the case studies is given in the table on the next page.
The entries in the table should be interpreted as described below.
Researcher:

- Position - position held by the researcher at the time the study was conducted. Experience - number of years experience with empirical research in software engineering.
- Relation with organization - whether the researcher was employed by the organization in which the study was conducted or not.
- Location - whether the researcher performed the study from within the organization or was located outside of it.

Organization:

- Involvement - how much the organization was/seemed willing to invest in the research project, both in terms of effort and money.

Viewpoint:

- Organization - the study is described by a member of the organization.
- Researcher - the study is described by the researcher.

Well-defined goals:

- Organization - the organization had clearly defined goals for the study.
- Researcher - the researcher had clearly defined goals for the study.

Expected benefits:

- Organization - to what extent did the organization expect results which could lead to improvements in its software development processes.
- Researcher - to what extent did the researcher expect interesting and useful results that could be published and lead to further research.

Costs:
Rough estimates of the costs involved in the study as estimated either by the organization or the researcher, depending on the viewpoint.

Usefulness:
Results as perceived either by the organization or the researcher, depending on the viewpoint.

	CASE 1	**CASE 2**	**CASE 3**	**CASE 4**	**CASE 5**	**CASE 6**
Researcher:						
Position	Professional	M.Sc.stud.	M.Sc.stud.	Ph.D. stud.	Ph.D. stud.	Ph.D. stud.
Experience	3 years	None	None	None	None	None
Relation with organization.	Semi-internal	External	External	External	External	External
Location	Remote	Remote	On site	Remote	Remote	On site
Organization:						
Involvement	Medium/ Low	Medium/ Low	High	Medium/ Low	Low	Low
Viewpoint:						
Organization	Yes	Yes	Yes	No	No	No
Researcher	Yes	No	No	Yes	Yes	Yes
Well-defined goals:						
Organization	Medium	Medium	Medium	High	None	None
Researcher	High	Medium	High	High	Low	Low
Expected benefits:						
Organization	Medium	Medium	High	Medium	Low	Low
Researcher	High	Medium	High	Medium/ High	Medium	Medium
Costs:						
Organization	100 hrs	30 hrs	200 hrs	40 hrs	8 hrs	50 hrs
Researcher	800 hrs	200 hrs	1000 hrs	300 hrs	100 hrs	500 hrs
Usefulness:						
Organization	Medium	Very low	High	Medium	None	Medium/ Low
Researcher	High	Medium	High	Medium/ High	Low	High
Software customer	-	-	-	Medium	None	Low

Table 1: Case Study Overview

CASE 1

Goals

The research goal was to study the impact of software CASE tools on development and maintenance productivity, and to develop a method for comparing CASE tools. These goals were the organization's goals, but while the research emphasis was on the development of a method, the studied organization was more interested in the actual productivity comparison, i.e. a conflict in focus of the study. The organization was, in addition, interested in a comparison with other similar organizations (benchmarking).

Research context

The research organization and the software house belonged to the same "mother" company.

Methodological Characteristics

Study of four CASE (Computer Aided Software Engineering) tools and eleven development and maintenance projects in the organization. Information based on tool information, historical data and in-depth interviews with the CASE tool experts to better understand the measured results.

Risk Items

- There was a need for interviews of and support from developers and managers.
- The usefulness for the organization was not obvious. The initiative came from the research department. Although, we got permission to study the productivity of the CASE tools it was not obvious that we would get real support.
- If the development teams believed that they would be evaluated (and not the CASE tools), they could manipulate the data they gave to us to get better productivity.

Initiation of Study

We started the study by informally contacting "key personnel" (but not the management) in the software house to get a better understanding of the availability of data and the current management focus, including the short and long term goals of the organization. This knowledge was important when presenting the research project to the managers and led to a change in the focus of the research project (to a higher focus on productivity measurement to be used in the estimation models and to compare productivity with other organizations). In addition, we had identified that most of the data we needed was already collected (but not systematically used by anyone!). Our argumentation and "goodwill", for this reason, got much stronger.

The management accepted our research project with no large remarks. Unfortunately, we got no senior manager to "sponsor" our research project, which we interpreted as a sign of low management interest in what we did, i.e. no real support. This low interest was, in our case, not a major problem for the research, but we believe that our work would have been significantly more useful for the organization if the management had been more involved.

Data Collection

The data collection went without large problems. The developers answered our questions and supported the collection of data, with very few incidents where the developers had a negative attitude towards our work. We did not interpret this as a high interest in what we did, but that the developers were polite and supported a management decision. We told the developers that they would be asked to quality assure the data before anything was published, but received no positive or negative reaction on this information.

Only one of the measured projects got high attention: The measurement of a pilot project using a recently selected CASE tool. This CASE tool was intended to be used for the development of "enterprise applications", i.e. large, business critical applications and the top management wanted us to measure the productivity of the first pilot project using this CASE tool. Probably, to assure that the choice of this CASE tool was successful. Many managers and developers had been involved in this decision, and it was an investment of several million NOK. To our surprise, we measured a very low productivity. We could of course not conclude that this expensive CASE tool lead to low productivity from only one project. On the other hand, the developers in this project were experienced, and the application complexity was low. In a way, we were caught in a trap. The large investment in the new CASE tool was not possible to reverse, and a high publicity on a potential low productivity (we would need more projects to be more confident) could create a lot of "political" discussions disturbing our research. To avoid problems for the rest of the data collection, we kept this results secret ("data has not been sufficiently analyzed, yet") as long as the data collection period lasted. In addition, we decided that this project would not be a part of the research report, but would be treated separately and with very much care. Otherwise, we believe this particular project measurement would have taken very much of our energy and effort.

Data Analysis and Quality Assurance

When the data collection was finished, we summarized the results for each CASE tool and asked 1-2 experts from each CASE tool to comment on our findings and the measured data. Some comments were received, which we included in the report. Although we asked the developers to quality assure the data, we found that the only way we could feel confident in the data quality was to control the data collection process or to discuss the measured values person-to-person with the developers involved. As expected, the measured pilot project using the new CASE tool created some heated debate, and was used (and to some degree misused) to criticize the way we measured productivity. This debate "ended" with a wish for more measurement of projects using this CASE tool.

Presentation of Results

There was no single presentation to spread the results. Instead, there were several smaller presentations "tailor made" for different development teams and management. A research report was written and distributed to the managers and the developers participating in the data collection. These "tailor made" presentations made it possible to focus on only the CASE tools relevant for a particular development team.

The productivity data was made available to the "estimation team". They contributed in the decision to remove one of the CASE tools, and they enabled an improvement of the estimation models. There has been no significant use of the method for comparing CASE tools by the organization.

A paper version of the report was presented at a conference (Jørgensen *et al.* 1995).

Lessons Learnt

- Knowledge about the organization was a very important factor to increase the probability for a successful research project. In particular, this knowledge should have an impact of the study design and the presentation to the management to get permission.
- A "what's in it for me"-analysis is a very good predictor for level of support.
- We were naïve regarding the usefulness of our study as perceived by the developers and overestimated, thus, the level of developer support. To get quality assurance of collected data, we had to control the collection process and carry out detailed interviews with the involved developers.
- Try to eliminate "political issues" from the research, e.g. to be part of a "play" where the goal is to show that certain decisions are correct/incorrect.
- Very sensitive data should be considered left out of the study if the data is not essential for the research results and the inclusion of the data put strong restrictions on the publication of the results.

- Involvement by the organization itself is no precondition for successful research, but, probably, for successful use of the results by the organization.
- The presentation of the results should be tailor made towards different roles and needs, and answer "what's important in this study for me".
- Try to find "pockets of opportunity" for use of the results, e.g. productivity data to improve the estimation model when the research goal is to create a method for comparing CASE tools. If possible, create a WIN-WIN situation for all roles.

CASE 2

Goals

Both the researcher's goal and the organization's goal were to improve the software estimation model. The IT organization felt, however, that the real goal of the researcher was to write a report.

Research context

The researcher was external to the organization and had no previous relations with it.

Risk Items

- The research project had to be completed within approx. 3 months.
- At the start, researcher had no knowledge about the organization or the estimation routines used by the organization.
- The contact persons in the organization had little experience cooperating with external researchers.
- The researcher was located far from the organization. The organization and the researcher were communicating through sporadic meetings, mail and telephone.
- The research goal was not clearly defined (due to the short time frame the goal was in practice formulated "do as much as possible").

Methodological Characteristics

The study was a case study of the estimation processes of an organization. Interviews and historical data were the information sources.

Initiation of Study

The researcher was recommended by another researcher (who had contacts in the organization). The project was started with a kick-off meeting where mainly the organization presented their processes and goals. The researcher was, at this point, very undecided/unclear about his research goal.

Data Analysis and Quality Assurance

The data was given to the researcher as a number of spreadsheets and documents. The researcher obviously needed more context to use the data meaningfully, but did not ask for this information. The organization did on the other hand not follow up this lack of information asking.

Presentation of Results

The results were only presented in a report and were, from the organization's viewpoint, not of any value.

Lessons learnt

- A risk analysis in the start of a risk project should be carried out in cooperation between the organization and the researcher. If such a risk analysis had been carried out, the organization would probably have been more aware of the high risks due to the short time frame and the need for the researcher to understand the organization.
- A "contract" where the organization and the researcher agreed on the expectations to each other would have been very useful. In this case, probably, both the researcher and the organization were disappointed regarding the other party.
- If a researcher does not regularly ask questions to better understand the organization and the collected data, this is a sign of high risk of low quality research and thus less useful for the organization.

CASE 3

Goals

Improving the user interface of a Lotus Notes based software process handbook (ongoing project in its final phase). The organization wants the user interface to be improved as much as possible within the available time frame, while the researcher has to focus on a scientific (empirical and analytical) argumentation of the impacts of the changes he proposes.

Research context

The study was conducted at a large Norwegian software house. The researcher was external to the company.

Risk Items

- The organization had to invest much effort and money (hardware, office, etc.) to make the research meaningful. If key personnel would not cooperate, the research would not succeed. (Luckily, the CEO had the opinion that "if we let him do this project, we will treat him as an internal".)
- The researcher had to invest a lot of effort in understanding the processes and organization roles. If not, the study would be of little use for the organization.
- The organization had specific needs (improving the user interface of the process handbook), while the research aimed for more general results. The output of the research project had to fulfill both these needs.

Methodological Characteristics

An in-depth case study of the processes of one organization. Experiments on different types of user interfaces using students.

Initiation of Study

The group leader of the "software improvement team" and the researcher agreed on goals and the milestones. The researcher got an office close to the persons he would cooperate with. A large part of the research work is integrated into the goals of the team, which ensure that the results will be used by the organization.

Data Analysis and Quality Assurance

First, two preliminary studies were carried out (interviews and observations of process handbook users). Then, an experiment has been carried out to test the proposed changes in the user interface. The research has achieved a high level of goodwill and permissions in the organization through several actions:

- The researcher supported a "maturity study" of the organization by carrying out the analysis (which was only slightly relevant for his work).
- He has regularly been presenting his ideas to the involved persons.
- He has gained an in-depth understanding of the process handbook and the organization (in the beginning some of the people were a bit tired of his silly questions, but very soon they found it useful to discuss user interface issues with the researchers).
- He presented general guidelines about how to carry out user interface test to the developers on a seminar.

The work is regarded as successful from both viewpoints. The organization has decided to implement the proposed changes in the software process handbook user interface, and the researcher has collected a lot of interesting data to be analysed.

Presentation of Results

The intermediate results of the study have regularly been presented to the process owners. A research paper is planned.

Lessons Learnt

- In studies where an in-depth understanding of the organizations and its processes is important, let the researcher be close to the persons he/she will cooperate with.
- The researcher will gain a lot of goodwill taking part in some of the organizations regular activities. This goodwill can be very important when demanding a lot of effort from the organization.
- Regular presentations and discussion with the involved increases the support - at least if there is something of interest to the organization.
- If possible, integrate the research plans in the organizational plans.

CASE 4

Goals

- Process evaluation, with focus on the cost of implementing changes during evolutionary development of object-oriented software.
- Evaluate a measurement framework for software changeability assessment.

Research context

The software development organization was responsible for the development of an in-house software system for the software customer. The project was organized in several project phases (product increments) with more functionality delivered in each phase. Researcher had the role of a process evaluator and was primarily associated with the software development organization.

Methodological Characteristics

An in-depth case study of the development process in one organization. The case study collected and analyzed both quantitative data (process and product measurements) and qualitative data (interviews and informal discussions with developers and management). The second project phase will include aspects of action research.

Risk Items

- Software customer could resist that time and money was spent on "non-productive" process evaluation activities.
- Software developers could resist the measurement of cost and productivity.
- Low commitment from the developers could reduce quality of data.
- The product was a strategic and competitive means for the software customer, and data sensitivity and confidentiality could become a problem from a research perspective.
- The software customer could decide to terminate the project after the initial phase, resulting in much wasted effort for the researchers.

Initiation of Study

This study was initiated by having informal discussions with the management of the development organization to determine whether the given software development project could be useful "pilot project" for the evaluation and improvement of an in-house, evolutionary development process called Genova (Arisholm *et al.* 1998). While the development organization was formally responsible for the pilot project, most of the practical work was done by the Ph.D. student (the researcher) who wanted to use the case study in his Ph.D. research. The next phase involved a meeting with the software customer where the researcher and his supervisor presented an overall motivation and goal of the study to the project management of the development organization and the software customer. During this meeting, we primarily focused on the potential benefits for the software customer (independent evaluation of the project and the software process). It was agreed that any effort specific to the process evaluation (the case study) should be reported separately from the regular development activities and that such efforts should not incur costs for the software customer.

A plan was made for how data should be collected and presented. The case study was organized in two phases that corresponded to the two project phases (product increments to be delivered). During the first phase, the primary purpose of data collection and analysis was to identify potential improvements in both the research process (process instrumentation, data collection and analysis) and the development process. The second project phase would implement the suggested process improvements and evaluate the results.

Data Collection

Initially, the plan was that the researcher would actively monitor the project during software development. This would ensure that data-collection issues could be resolved early and hence ensure better quality data. However, due primarily to unanticipated technical problems during the first project phase, the developers and project management were very "busy" and paid little attention to the case study. It became difficult for the researcher to get the necessary time and resources to observe the development project in detail. This resulted in that the data collection had to be based on historical process and product data collected from the configuration management database, weekly status reports and timesheets once the first phase of the project was finished. Consequently, we were unable to adjust the instrumentation of the process during this first phase of the project (e.g., improving reporting procedures and configuration management guidelines). This in turn reduced the quality and usefulness of the initial data.

Data Analysis and Quality Assurance

Quality assurance was performed by means of discussions of the data with the software development organization. Interviews with key resources are also planned, but time pressure has made progress slower than expected. Important lessons were learnt that will be applied to improve data collection and analysis during the second phase of the study.

Presentation of Results

A workshop paper (Arisholm and Sjøberg 1999) was published based on the results from the first phase of the project. The results are interesting at least from a research perspective, and they generated many new ideas for future research. Care had to be taken to ensure that sensitive information was not made public, and that both the software development organization and the software customer had approved the content of the paper before it was published. In addition, a report is currently in preparation by the researcher that will summarize the experiences and results from the first project phase. This report will describe the process evaluation and suggestions for process improvements to be implemented in the next project phase. These suggested process improvements are intended to be beneficial both to the software customer and to the software development organization. The report will also contain explicit recommendations to improve future data collection.

Lessons Learnt

- Historical data may be useful, but "real-time" monitoring and data collection would have enabled the adjustment of the process instrumentation such that more useful and higher quality data could have been collected.
- Initial meetings with focus on the potential usefulness for the development organization and the software customer resulted in high initial commitment and interest from all parties.
- High involvement from the researcher resulted in high initial commitment from the development organization and software customer. Unfortunately, unanticipated technical problems resulted in significant time-pressure during the project. Therefore, the case study work still received low priority.
- It is likely that the case study would have received higher priority if the research goals were better integrated with the organizational plans and goals. In this project, however, this was difficult to achieve since the development project essentially was financed by the software customer who could not be expected to fund additional research related work activities.

- To improve the data collection and data analysis process, it was useful to divide the case study in phases where the first phase could be more exploratory in nature. This means that subsequent phases can incorporate important lessons learnt during the initial data collection and data analysis.

CASE 5

Goals

The description and formalization of a legacy system development process. This was intended as the first step of an initiative to improve software maintenance processes for legacy systems in the software development organization.

Research context

The software development organization participated in the development of an in-house software system for the software customer together with participants from the customer organization

Methodological Characteristics

The study can be classified as a typical case study where one instance of a legacy process was to be studied in detail. There were problems with control as described below. If the study had been completed there could have been problems with generalizing this legacy process for further use.

Risk Items

- The development project was located at the software customer approximately 600 km. from where the researcher worked. Hence, personal contact would be limited.
- A software system developed by a third party software vendor was used as a basis for the development. Consequently, there were three parties involved in the development project and they would all be affected by the research project to some extent.
- The product was a strategic and competitive means for the software customer. The development project could therefore reveal sensitive information to the researcher.
- The software customer could resist that time and money was spent on ”non-productive” process description activities.

Initiation of Study

The study was initiated by having informal discussions with the management of the software development organization to determine whether the proposed research project could be useful for the organization, and whether the proposed development project could be of interest for the researcher. The planned result from the case study might have given better knowledge about how an existing software system can be used in the development of a new software system. This could lead to new business opportunities for the software development organization.

A manager from the software development organization, who was involved in the actual development project, then obtained permission from the project manager, employed by the software customer, to pursue the research project. The project manager received no thorough presentation of the research project or possible benefits for his organization. It is consequently likely that he accepted the project because of a longlasting relationship with the software development organization, and not because he thought it would produce interesting results.

The next phase involved a meeting between the researcher, the project members from the development organization and the software customer. At the meeting the researcher presented an overall motivation and goal of the study, and the project members gave a brief description of the project. The project members were concerned about the possibility that the research project might lead to the publication of sensitive information about the organization. They were reassured by the fact that all details would be anonymous and that the goal of the research process was software improvement, which was not the software customers area of business.

Data Collection

The researcher should get access to all relevant information about the project, partly from documentation and partly from interviews with project members. An initial process model should then be defined from this material, and this process model should be corrected and eventually verified by the project members.

Due to the difference in locations, a set of questions was sent by mail to one member of the development project. The questions were forwarded to the project manager. He decided that answering them could reveale sensitive information about the software customer and third party software vendor that might complicate their relationship. He said that he did not believe that a confidentiality agreement would be sufficient to prevent these negative effects, and consequently did not want to reveal enough information about the development project to make the research project feasible. The research project was stopped.

When the project manager accepted the research project, the development project was on schedule and apparently successful. At the time he received with the questions, we believe that the development project had met unanticipated difficulties, and that this made the project manager reluctant to allow an outsider access to the project.

Lessons Learnt

- The fact that three organizations (the software development organization for which the study was to be conducted, the software customer and the third party vendor) were involved is probably one of the main reasons why this research project failed. The development organization and the researcher would probably benefit from the research project. To some extent, this could also have been the case for the software customer. However, confidentiality was considered more important than the positive effects that could be expected.
- The necessary involvement and backing from the organization(s) must be obtained before the initiation of the research project. It is important that both (all) parties understand the possible implications of the project. In this case, the importance of these aspects was probably underestimated. To prevent such problems, all parties should meet before the initiation of a research project and discuss both research goal and organizational goals thoroughly.
- The difference in locations made personal contact between the researcher and the project manager and members difficult. The project manager and the researcher never met. This might be one reason why the project manager did not trust the researcher with sensitive information. The researcher could have considered staying near the development project for the duration of the research project to avoid problems due to

the difference in location.

CASE 6

Goals

- Identification of the requirements of a change management tool (in particular, consistency checking and impact analysis).
- Study of the effect of using such a tool.

Research context

The software development organization was responsible for the development of a health management system currently being used in more than 20 hospitals. The continuous change in requirements made it difficult to keep track of all the needed consequential changes in the system implementation.

Risk Items

- The research was initiated by a consultant who was the supervisor of a Ph.D. student who actually carried out the research, that is, the research was not initiated by the company itself.
- The developers could resist revealing problems of change management as this could give the impression of unsatisfactory job performance.
- The developers could be reluctant to spend time on the research as this could not be prioritized to hard deadlines.

Methodological Characteristics

The study was both a case study and action research. The first part involved measuring the extent of evolution in all the software being developed in the health management project. These measurements were collected automatically outside working-hours with only the researcher involved. Hence, the working environment of the employees was not affected by this measurement collection.

The second part of the study was action research in that it involved developing a tool to help the employees keeping control over all the changes to their software and thus improve their development process.

Initiation of Study

This study was initiated by a consultant who experienced great problems in keeping track of all components and their dependencies when things had to be changed. It was agreed with the company management that a researcher (a Ph.D. student of the consultant) should stay at the company for two-three months to investigate the problems, implement a change management tool and evaluate it. The consultant and the managing director made an agreement that the researcher should work for free, but travel and housing expenses should be covered by the company. The manager of the health management project should be the contact person of the researcher and should introduce him to relevant project members.

Data Collection

As it was expected that the developers were busy, it was already from the beginning planned that most data should be collected automatically. Since it took 6-7 weeks to design and implement the data collection tool, the researcher had to make judgements from observations and informal discussions with the developers during the initial phase. The evaluation of the change management tool was supplemented with interviews and responses at demonstrations. However, the developers showed little interest in using the change management tool, and did only to a minor extent contribute to the evaluation of the tool.

One reason for the reluctance by the developers to use the tool was probably that they did not like that a tool made by an outsider, a foreign Ph.D. student, should analyze all their software, and detect bugs and inconsistencies caused by themselves. Another reason might have been that they were not convinced that it was worthwhile to spend time on the tool when short-term deadlines had to be met. If they did not see any personal gains, why should they be co-operative?

When the researcher left the company after the first period, the data collection tool was set up to be run automatically every night to study the extent and pattern of software evolution over time.

Data Analysis and Quality Assurance

When the researcher returned to the company after one year, it turned out that much data was lost due to changes in the environment that made the data collection tool stop working. The tool had to be changed to cope with new compilers, DBMS versions, etc. before it could work again. Hence, the measurements of evolution could not be based on regular intervals as originally planned, but trends could still be observed.

Presentation of Results

A journal paper (Sjøberg 1993) was published based on the results from the evolution study. The project manager read the paper and accepted it, as he felt it contained no sensitive information. The evaluation of the tool was presented as being based on only anecdotal evidence. The reluctance by the developers to take part in the research was not mentioned in the paper.

Lessons Learnt

- When introducing a tool that automatically checks the quality of software, one should ask: Who should use the tool? How should the working process be organized to benefit as much as possible from the tool? How should the project management motivate and encourage active use of the tool?
- It is particularly important that inexperienced and immature developers find bugs and inconsistencies by themselves before the software is released. The only purpose of the tool should be to improve the quality of the software; a negative attitude may be created if it is felt that the tool is used for individual monitoring purposes, e.g. by the project management. Therefore, establishing confidence between the developers and the researchers is crucial. To have gained more confidence in this study, the researcher should probably have been more social and paid more interest in the daily work of the developers that was not the focus of the research.
- Automatic data collection must be checked regularly to determine whether it is

working. Even if the data collection software has been properly tested, changes in the environments may make it fail.

Case Study Research Guidelines

This section describes guidelines that are a result of the lessons learnt in the six case studies presented in the previous section.

High-risk items

- Avoid large geographic distance between researcher and organization. Frequent, direct contact may alleviate communication problems and subsequent conflicts.
- Keep the number of organizational "layers" between the researcher and the studied organization to a minimum.
- Ensure necessary involvement and backing from the organization(s) before initiating the research project.
- A negative attitude may be created if a developer feels that the study can be used for individual monitoring purposes. Ensuring that this is not the case may improve the cooperation from the developers.

How to get inside?

- Know the organization (its goals, focus, earlier research on that organization, etc.).
- First presentation/discussion is essential and should focus on a realistic assessment of the usefulness of the research for the organization (ask yourself, what would make you as a manager to say "yes" to this project).
- Use your personal network.
- Organizations may consider regular contact with students as a good opportunity for recruitment.
- Results from the case study may have a marketing effect for the organization if presented in the right media. Capitalize on this factor during initiation and planning, and follow up after the results are available.
- Agree on plans (but be open to redirect goals and scope).
- Discuss expectations with each other.
- Discuss major risk factors.

How to get high quality data?

- An initial "pilot-study" might be useful to assess and reduce risks, improve the data collection process and focus the research goal.
- When introducing a tool that automatically checks the quality of software, one should ask: Who should use the tool? How should the working process be organized to benefit as much as possible from the tool? How should the project management motivate and encourage active use of the tool?
- Whenever possible, collect "real time" data. Historical data may have lower quality and validity.

How to present the results?

- Sensitive data should be considered left out of the study.

- Sign a confidentiality agreement with the organization. Let the organization read and accept publications before submission.
- Intermediate results should be presented frequently to ensure that the organization understands the motivation, goals and potential organizational benefits from the research. Write and present reports in addition to scientific papers.

Conclusions

There is no doubt that carrying out case studies in industry is a difficult task. Consequently, we should share practical case study experience within the information systems and software engineering communities. In this paper, we have presented the experiences and lessons learnt from six case studies with emphasis on cooperation challenges and the potential conflicts between the organizational goals and the research goals. We believe that the resulting guidelines might be useful for both Ph.D. students and experienced researchers.

Acknowledgements

We gratefully acknowledge the support from the companies in which we carried out the case studies. Our research is funded by The Research Council of Norway through the industry-project SPIQ (Software Process Improvement for better Quality).

References

Argyris, C. and Schön, D.A. (1991). Participatory Action Research and Action Science Compared. In *Participatory Action Research*, Sage, Newbury Park, CA.

Arisholm, E., Benestad, H. C., Skandsen, J. and Fredhall, H. (1998). Incorporating Rapid User Interface Prototyping in Object-Oriented Analysis and Design with Genova. In *Proceedings of NWPER'98 Nordic Workshop on Programming Environment Research* (Eds, Mughal, K. A. and Opdahl, A. L.), Sweden, pp. 155-161.

Arisholm, E. and Sjøberg, D. (1999). Empirical Assessment of Changeability Decay in Object-Oriented Software. In *ICSE'99 Workshop on Empirical Studies of Software Development and Evolution*, Los Angeles.

Braa, K. and Vidgen, R. (1999). Interpretation, intervention, and reduction in the organizational laboratory: a framework for in-context information system research. To be published in *Accounting, Management and Information Technologies*, Elsevier Science Ltd.

Jørgensen, M. (1995). Empirical evaluation of CASE Tool efficiency. In *Sixth Int. Conf. On Applications of Software Measurement*, Orlando, pp. 207-230.

Jørgensen, M., Bygdås, S. S. and Lunde, T. (1995).Empirical evaluation of CASE tools. In *Norwegian Informatics Conference*, pp. 71-82.

Kitchenham, B. (1990). Software Metrics: Measurement for Software Process Improvement, *Blackwell,* ISBN 0-534-08010-3.

Rapoport, R.N. (1970). Three Dilemmas in Action Research. In *Human Relations*, (23:4).

Simon, H. (1969). The science of the artificial. *The MIT press*. Cambridge, MA.

Sjøberg, D. (1993). Quantifying Schema Evolution. In *Information and Software Technology,* Vol. 35, No. 1, pp. 35-44.

Zelkowitz, M.V. and Wallace D.R. (1998). Experimental models for validating technology, In *IEEE Computer*, May 1998, pp. 23-31.