Chapter 10
The Management of University–Industry Collaborations Involving Empirical Studies of Software Engineering

Timothy C. Lethbridge, Steve Lyon, and Peter Perry

Abstract In this chapter we will discuss some of the pragmatic considerations that we believe university researchers and companies should consider when establishing collaborative software engineering research projects; in particular, those involving empirical studies of software engineers. The chapter is illustrated using as a case study a research collaboration in which the authors are involved. We enumerate the costs, benefits, risks and risk-reducing factors that can have an impact on all the parties involved in the collaboration (the company, the faculty members and the graduate student researchers). Understanding this information is needed to help justify the research in the first place, and to manage it effectively. We then discuss many of the activities that will be needed to plan and manage the project, including such issues as attracting students, handling intellectual property, obtaining ethical approval and interacting with participants. The main objective of the chapter is to provoke some thoughts in the minds of those planning empirical research projects in software engineering.

1. Introduction

Most software engineering tools and techniques are aimed at reducing cost, speeding development and/or increasing software quality – all in the context of the pervasive complexity and rapid change one finds in industrial software projects. Researchers must conduct empirical studies in industrial settings in order to properly understand the complexities of commercial software products and processes, and to evaluate new ideas. This paper presents lessons we have learned through a university-industry research collaboration in which the authors participated. The objective of the paper is to help guide others who are considering embarking on similar endeavors.

Empirical studies in companies can take many forms; the discussion in this paper does not presuppose one form in particular. Studies will most often investigate software engineering processes, but may also assess the usefulness of various technologies

F. Shull et al. (eds.), *Guide to Advanced Empirical Software Engineering*.

that software engineers use or develop. Some empirical studies, e.g. learning how much of a typical project's duration or effort is devoted to a certain activity, could stand on their own: Their conclusions would be used for general decision-making. Other empirical studies might enable the researchers to form hypotheses about, or validate, their own research ideas. Examples of the latter include novel testing techniques or programming languages.

Empirical studies can use a variety of techniques ranging from questionnaire-based surveys, structured interviews and observation sessions to controlled experiments (Lethbridge et al., 2005; Sjøberg et al., 2005). Almost all these techniques involve people as research participants. Traditionally students have performed this role, but as emphasized above, it is often essential to use industrial employees in order to obtain accurate and relevant answers to many research questions.

Researchers in empirical studies can take on the role of the indifferent outsider, observing and measuring what goes on in the company. Or they can take on a more participatory role, seeking to improve the industrial environment by conducting *action research* (Potts, 2003; Baskerville and Wood-Harper, 1996; Checkland, 1991; Dittrich, 2002).

Conducting empirical studies in software companies is not easy. In this chapter we will focus on how to plan and manage such projects; we will look at how to justify such projects, find participants and staff, deal with the competing interests of the researchers and company managers, as well as various other issues. Additional challenges, discussed elsewhere in this book, arise from the need to conduct good science. The latter challenges include establishing adequate experimental controls, choosing appropriate metrics, and properly analyzing the resulting data.

Software engineering researchers are normally not trained in management. As more of them recognize the imperative to conduct empirical studies in industry, we expect increasing interest in learning from the experiences of others. In this chapter we present a set of issues that researchers need to consider, illustrated by the case study of a research project in which the authors collaborated.

The authors represent both industry and academia and have each conducted research with several different partners. The academic author has also worked in industry. The issues raised in this chapter are therefore derived from a variety of experiences.

There is some existing literature about industry-university collaboration. Conradi et al. (2003) discuss experiences in Norway in which several small and medium enterprises (SMEs) and several universities jointly worked on process improvement research. Some of the lessons-learned they present are similar to the ones we present here, although our experiences relate more to individual performance improvement rather than company process improvement. Beckman et al. (1997) and Mead et al. (1999) provide some suggestions about another type of industry-university collaboration – working together to design and deliver educational programs. Arisholm et al. (1999) provide a series of small case studies about industrial collaborations, each with their own lessons learned. Finally,

Rombach and Achatz (2007) summarize a variety of issues regarding research collaborations.

In the next section we give a brief overview of the research project that will serve as the case study. We then enumerate the benefits of university-industry research projects and the factors that can lower risks. Following this we discuss the costs and the risks themselves. We conclude by presenting a set of considerations that industrial and university researchers should consider as they plan their projects.

2. An Example Research Project: The Mitel – University of Ottawa CSER Collaboration

We will illustrate this chapter with examples from our own experiences as University of Ottawa researchers and Mitel managers conducting collaborative research. These results are personal reflections gathered from brainstorming ourselves about what worked, and how we could have conducted our research better.

Mitel is a medium-sized telecommunications company, best known for its PBX hardware and software. As with all telecommunications software, the Mitel systems are very large.

In 1995 the Mitel managers (the second and third authors of this paper) approached University of Ottawa researchers with a general research problem: How to reduce the cost of maintenance of a large software system. As is normally the case when starting such projects, we had particular ideas we wished to test. We believed that one of the biggest difficulties faced by the engineers was an inability to visualize the system's design, due to its complexity and the sheer magnitude of its code and documentation. In earlier research, the first author had developed a knowledge base management system (KBMS) (Lethbridge, 1994) and believed that if we modeled the Mitel system using this KBMS we would be able to help Mitel engineers to understand their system better. Such a KBMS model was expected to be especially helpful in enabling new design staff members to learn the Mitel system, and become productive more quickly.

Since we wanted to apply good scientific method, we decided that an important part of the research would be to study software engineers and their product (Singer and Lethbridge, 1998). The objective of this was to better capture the nature of the problem that the KBMS was supposed to solve, and to develop hypotheses that we would later seek to confirm. Before long, we noticed several patterns in the work of the engineers. In particular, they were spending a large amount of effort searching code, and they were having significant difficulty manipulating and organizing the results of their searches. They were thus finding it hard to effectively use this information. As a result we changed our research direction considerably and focused on designing a tool to solve these immediate and pressing problems. Investigating the KBMS ideas dropped to a lower priority.

In 1996, Mitel joined the Consortium for Software Engineering Research (CSER, www.cser.ca,), and the research project grew to encompass studies of various features that might be appropriate in a software exploration environment. The tool that we developed, *TkSee* (Lethbridge and Anquetil, 1997), saw continuous voluntary use by Mitel engineers from the date it was introduced (1996) until several years after the project concluded in 2002. It also served as a test environment for several aspects of the research. In the rest of this chapter, we will refer to this work as the Mitel-CSER project.

Research on the Mitel-CSER project used many approaches: To gather data from software engineers we measured their use of tools, interviewed them, asked them to draw pictures describing their views of the architecture of some software, and shadowed them. We developed a new shadowing technique called *Synchronized Shadowing*, and a new approach to analyzing the large amount of data that results – representing *work patterns* using use-case maps (www.usecasemaps.org). We have conducted usability studies (Herrera, 1999) to ensure our tool is usable. We believe that if the tool has poor usability, this would negatively impact user acceptance, hence we would not be able to tell if its core functionality was useful or not. We also developed techniques for analyzing Mitel software (Somé and Lethbridge, 1998) that are used to build the databases that TkSee uses.

The research involved the academics immersing themselves in the industrial environment – not to the extent of actually working on Mitel products, but rather through being on the premises and actively trying to solve problems faced by the developers. We therefore followed the research paradigm suggested by Potts (2003), in which one 'intertwines research and industry intervention'.

Both the academics and the company benefited from the research. Mitel was pleased with the impact of the tool, and the academics were able to produce many publications, (e.g. Anquetil and Lethbridge, 2003; Anquetil and Lethbridge, 1999; Sayyad Shirabad et al., 2003; Lethbridge and Singer, 2001; Liu and Lethbridge, 2002; Somé and Lethbridge, 1998).

However, there have also been several difficulties that turned the research into a good case study. Most notably, it has not been easy to motivate graduate students and others on the research team to embrace techniques that involve studying work practices and software usability. It has also not been easy to strike a balance between conducting well-designed and focused research on the one hand, and solving difficult-to-characterize industrial problems on the other hand. We sometimes spent excessive effort developing software of sufficient quality so that it can be actually used by the engineers – necessary so we can determine if our ideas are valid. We similarly had difficulty attracting a large enough population of users to scientifically validate our ideas, although several Mitel users have used TkSee extensively.

The Mitel-CSER research project is considered successful despite these difficulties. We hope our accumulated lessons-learned as presented in this chapter will be of value to others who embark on similar research.

3. The Benefits of University-Industry Software Engineering Empirical Studies

In this section and the next we will enumerate the positive and negative sides of empirical software engineering research projects involving companies and university research groups. Before starting any such project we believe it is important to attempt to quantify these factors. The information may be used to help 'sell' research projects to either the company or the researchers, to plan such projects and to manage risk.

In what follows we separately enumerate the benefits to the company, to faculty members and to students involved in the research. These are summarized in Table 1 While many of these benefits might be self-evident, the parties may not necessarily

Table 1 Benefits of industry–company research collaborations

Category of benefit	Benefit type	Typical amount of benefit (impact * probability of occurrence)
To the company		
Direct benefits	• New or improved technology or product	Medium
	• Data and knowledge useful for decision making	High
	• Patents	Low
Indirect benefits	• Potential employees for company	Medium
	• Ideas and expertise of researchers	High
	• Public relations	Medium
Factors lowering risk of research	• Graduate students are often top achievers	Medium
	• Researchers have a personal stake in success	Medium
	• Low cost compared to in-house research	High
	• Government matching funds and tax incentives	High
To researchers		
Direct benefits	• Funding	High
	• Interesting and challenging problems and data	High
	• Test-bed for ideas	High
Indirect benefits	• Exposure to the 'real world': Provides valid and relevant knowledge, consulting and networking.	High
To the public		
Indirect benefits	• Advancement of state-of-the art and state-of-the-practice	High

think of all of them. We believe that systematically analyzing these factors, quantitatively if possible, should be done more frequently when research projects are planned. Knowing the potential benefits we can, a) balance them with the costs to decide whether the project (or an aspect of it) is worth doing and attract adequate funding, and b) make sure we actively work to realize the benefits.

3.1. Potential Benefits to the Company

Benefits to the company fall into three categories: Direct benefits, indirect benefits and risk-reducing factors. The direct benefits are what immediately spring to mind, and result from success of the research. However the indirect benefits might be of considerable value too. The risk-lowering factors are considered as a separate category of 'positive' factors that make it worthwhile doing the research in conjunction with universities as opposed to in-house.

3.1.1. Direct benefits

The most obvious direct benefit to the company is new or improved technology (processes, techniques and tools) and products. Empirical software engineering research does not itself normally directly create such improvements, but provides data and knowledge useful for making management or design decisions.

For example, in the Mitel-CSER project our studies of software engineers gave us design ideas and led to changes in research focus. Similarly, our studies of usability told us what tool improvements were necessary. We used data from an empirical study to develop the TkSee tool, which in turn reduced the elapsed time some new employees took to learn about Mitel software. In fact the training time for designers new to the product was typically halved, and this provided the most readily quantifiable benefit of the project. It is important to note that this kind of benefit requires management of technology transfer, an issue discussed by Zelkowitz (1995) and Pfleeger (1999).

Technology transfer involves taking an idea from laboratory prototype to permanent use of a mature product within a company or industry as a whole. One of the issues often faced is establishing the appropriate intellectual property framework to do this – for us, this was not a challenge because we had a well-written collaborative research agreement from the start, which anticipated close interaction with the company and had clauses clearly describing IP rights. We did, however find three *practical* technology transfer issues challenging: Firstly we needed to make our research software usable enough so that it could be used in daily practice; in other words we had to approach 'product quality'. We were able to achieve this by following rigorous usability engineering techniques, such as usability studies. The second challenge was integrating TkSee with the corporate tools and data infrastructure. Our database needed regular builds, and our server

needed to be maintained. We were able to train a Mitel staff person to do this, however, from time to time that person was unavailable, causing some down time. The third challenge was spreading the use of the tool from one focused team to the wider organization or industry as a whole. Although we attempted to do this, we never had any 'takers' beyond the original team. We were not able to make the extra investment of time and effort to broaden the technology transfer. We had quite a lot of requests from outside Mitel to obtain TkSee, but we found it hard to service these requests, since setting up the tool required a lot of time-consuming configuration.

Another possible direct benefit of empirical studies is intellectual property: Such studies might uncover data that could provide competitive advantage or a patentable invention.

3.1.2. Indirect benefits

In today's employment environment, where people with appropriate skills are often hard to find, an important indirect benefit of research collaborations is the exposure to the company of potential highly-skilled employees. Graduate students can learn a considerable amount about the company during their research and develop a desire to work there. It is important, however, for companies to actively recruit such students (as they approach the completion of their degree) in order to realize this benefit – in the Mitel-CSER project we learned this lesson only after the first few years.

A related indirect benefit to the company is exposure to academic researchers who can provide expertise and fresh ideas; this can be achieved through formal presentations or informal discussions. Faculty members will also absorb corporate know-how and the corporate needs for future stills; they will thus be in a better position to educate future employees.

A final indirect benefit is the public relations value resulting from the joint publication of research results.

3.1.3. Risk-lowering factors

Research can be conducted using in-house employees instead of university researchers. In many cases, however, the specialized expertise is not available, and both the uncertainty of the outcome, and the cost of the research are too high for the industrial agenda. There are several benefits from using university researchers: Graduate students tend to be talented individuals with the latest knowledge. They have a personal stake in the project's success and direct power over its success due to their need to complete a thesis – their main reward, graduation, does not come until success is achieved. Graduate students are also paid relatively little, seeing their work as an investment in themselves. Added to this is the benefit of the guidance of experienced faculty members.

Faculty members are also personally motivated to succeed in the research due to their need to publish papers, although this can be a double-edged sword as we will discuss later. Furthermore a faculty member's time may be at least partly 'free' to the company.

Finally, government matching funds that cover part of the cost to the universities and tax incentives for industrial research all reduce the risk to the company.

The lists of direct and indirect benefits are similar to the benefits of industrial collaboration reported by Conradi et al. (2003). Conradi et al. also discuss benefits to individual participants, but don't discuss the risk-lowering factors.

3.2. Potential Benefits to the Faculty Members, Graduate Students, and the Public

Significant benefits also accrue to faculty members and graduate students. Both categories of academic researchers directly benefit from significant amounts of funding for their work, interesting intellectual problems and data to work with, and a test-bed for their ideas. Indirect benefits include exposure to the 'real world'; the knowledge researchers acquire is likely to help the researchers improve other aspects of their research as well as their teaching. Opportunities for networking and consulting will also likely arise: Faculty members might find potential graduate students or other collaborators in the companies, while students might receive job offers.

Finally, as mentioned at the beginning of the chapter, there is one important public benefit to empirical studies in industry: They are necessary to properly understand the complexities of software engineering, and thus advance the state-of-the-practice, resulting in better and cheaper software-intensive products and services in most parts of our society.

4. The Drawbacks of University-Industry Software Engineering Empirical Studies

In this section, we present the drawbacks of university–company collaborations for empirical software engineering research. These factors should be balanced against the benefits discussed in the last section. Awareness of these factors can also suggest ways to manage and reduce them. Table 2 provides a summary.

We divide the sets of drawbacks into those that primarily affect the companies, those that affect the faculty members and graduate students, and those that affect the success of the project as a whole (impacting everybody who is interested in the results).

We also divide the drawbacks into costs and risks. Costs are factors that can be estimated directly, while risks are uncertainty factors for which one can estimate their probability of occurrence and their impact on costs and benefits if they occur.

Table 2 Drawbacks of industry–company research collaborations

Category of drawback	Drawback type	Typical amount of drawback (impact * probability of occurrence)
To the company		
Costs	• Cash funding	Varies from none to medium
	• Consumption of employee time	Varies, normally medium
	• Office space and equipment	Normally low
Risk factors	• Different definitions of success (bottom line for industry vs. scientific results and publication for researchers)	Medium if the company has defined the problem; otherwise low
	• Unknown consumption of employee time	Low to medium
	• Inappropriate release of intellectual property	Normally low for empirical studies
To researchers		
Costs	• Constrained research freedom	High if the company has defined the problem; otherwise low
	• Excess consumption of time	Moderate to high, depending on experience of researchers and research design
Risk factor	• Company-initiated cancellation	Varies from low to high depending on corporate priorities and rapport between researchers and the company
To the project as a whole		
Risk factors	• Different perceptions of the problem	High if the company has defined to the problem for researchers solve; otherwise low
	• Failure to staff the project with sufficient numbers of skilled researchers	Medium
	• Unknown skill level of researchers, including their ability to estimate the required effort	Varies from low to high depending on experience of researchers
	• Failure to find or keep adequate numbers of participants	Varies from low to high; depending on effort needed, management support, and other factors
	• Inconclusive or non-useful results	Low, but higher when the objective is to validate a hypothesis

Note that some projects are initiated by researchers while others are initiated by companies who have an active need to solve to a problem. Some risks are considerably higher in the latter case.

4.1. Potential Drawbacks to the Company

The costs to the company of participation in research projects with universities include direct cash funding of the research, consumption of employee and management time as well as office space, equipment and other supplies devoted to the research. For empirical studies, the time of research participants may be the greatest cost.

The following are risk factors that add uncertainty to the costs and benefits; these are listed starting with the most significant. Note that we enumerate risks to success of the project *as a whole* later in this section.

4.1.1. Different definitions of success

Unless a project is very small and the company is purely expecting indirect benefits (see Sect. 3), then the company will expect some concrete result that will ultimately impact their bottom line. Researchers, on the other hand usually have completely different motivations for participating, the main one being publishing results. This cultural conflict is explored in more detail by Zelkowitz et al. (1998).

This fundamental difference of interest can lead, in the worst case, to researchers not paying any attention to the needs of the company. Normally, with well-intentioned researchers, the impact is more subtle: The researchers might be stressed about their thesis deadlines, paper deadlines or other academic requirements and give priority to them. Or the researchers might deviate from a project plan that interests the company because they find interesting side-problems that will more readily result in publishable results.

This difference of interest is probably the biggest risk factor to companies, and thus must be carefully managed. In the Mitel-CSER project, this risk factor had a major impact – many graduate students wanted to direct their theses to topics that related to, but were not directly central to, the original project plan. The faculty member directing the project was also in the process of achieving tenure and so spent considerable time writing papers – sometimes leaving the project plan to languish at a lower priority for long periods.

4.1.2. Unknown consumption of employee time

In some empirical projects, such as those involving completing surveys, this is not a high risk. However for observational studies or those that involve open-ended investigation the risk is higher.

4.1.3. Difficulty controlling release of intellectual property

Companies tend to worry that publication of research results might cause them to lose competitive advantage. Some also have concerns about source code or design

information getting into the hands of competitors. If these issues are discussed during project planning (see Sect. 5), these risks can be minimized.

4.2. Potential Drawbacks to Faculty Members and Graduate Students

There are two clear costs to the academic researchers of collaborating with industry.

The first cost occurs when there are constraints placed on the freedom of researchers to follow their interests. Software engineering is a very rich domain with many potential problems and much data to be gathered. This richness, however, means that some problems will be considerably more interesting and easy to publish about than others. When working on an industrially-sponsored project, the researcher has a responsibility to the company and cannot readily sidetrack to pursue ideas that might prove more publishable.

The second cost to the researchers is the substantial amount of human resources that empirical studies take. Planning and managing an industrial research project can take far more time than many types of work that can be done on campus and with groups of students as participants.

A risk factor with big potential consequences to the researchers is that the company will undergo some form of reorganization or reprioritization, and cancel the research in progress. The academic author has experienced this several times. In fact, subsequent to the time when this paper was initially written, the Mitel-CSER project itself was cancelled, just after an agreement had been reached to continue it. The reason was simply a high-level decision from the corporate executives to cut all possible costs, including all external research.

A contingency plan for such situations is to work with two or three different companies on the same research problem, however this can be excessively time consuming and may not be possible if the companies are competitors. In case of project cancellation, all may not be lost. The data gathered so far can be reported as preliminary results, and can serve as a point of departure for a new study, or it can be combined with data in a later study. A sliver lining from a cancellation is that the researchers then are freer to work with other companies, where they may gain fresh perspectives. Indeed, we were able to replicate some of our work in IBM, who we later worked with, lending increased confidence to our conclusions.

4.3. Risks to the Research as a Whole

The following risk factors are typical of empirical studies at present. They can impact the ability to obtain useful results, or even to complete the project, and therefore affect both parties (although they only affect the company if it is sponsoring the project because it has a problem to solve).

4.3.1 Different perceptions of the problem

Academics without much experience in industry may have very different notions about what software engineering involves and what are the real problems. On the other hand, industry managers tend to vary widely in the software engineering knowledge they possess. This can lead to difficulty communicating, and misunderstandings about the problem that is to be tackled. This issue is very much related to classic difficulties in requirements analysis where, due to inadequate communication and preconceived ideas, customers have one perception of the problem and software engineers another.

4.3.2 Failure to staff project with sufficient numbers of skilled researchers

Empirical research has not customarily been widely performed in the software engineering community, and for some people lacks a certain 'respect' or is considered to be 'soft'. The Mitel-CSER project has certainly suffered from this phenomenon; we have on occasion tried to convince graduate students to become interested in such studies and have found that they don't see it as 'real' engineering. Empirical studies of usability, as performed by human factors experts, are seen to be part of an entirely different culture. For these reasons, it is hard for the project leaders to attract researchers (graduate students, postdoctoral researchers and faculty) who have expertise and interest. Hopefully this book will make a difference.

In addition to having questionable interestingness or respect, empirical projects also often generate profuse volumes of data, which is very time-consuming to analyze. This acts as a deterrent to software engineering researchers who are used to solving engineering problems. In the Mitel-CSER project, we attempted to use administrative assistants to transcribe tapes in interviews, however this failed because the interviews used so much technical jargon that the transcribers could not adequately understand them.

4.3.3 Unknown skill level of researchers

Even if staff can be found, conducting empirical studies is a skill in which not many software engineering researchers have been trained – something this book hopes to alleviate. Therefore the students, and even faculty, may well be on a learning curve and may make mistakes. Of particular importance is the ability of the researchers to estimate how much time empirical studies will take; our own lack of experience meant that this we severely underestimated when we developed our project plan.

4.3.4 Failure to find or keep adequate numbers of participants

It is common for researchers to get a low response rate to surveys; we conducted one mail-out survey as part of our research and obtained only a 2% response rate.

Within companies, it may be possible to interest participants in observational or interview-oriented studies, but it may be very hard to get enough people to use a specific piece of software as part of their work, or to follow a certain methodology. In addition, participants may leave the team or company, or withdraw from the study for personal reasons. In the Mitel-CSER project, we have suffered from all of these difficulties to a considerable extent, although we have been lucky to have a large enough pool from which to draw new participants.

4.3.5 Inconclusive or non-useful results

No research is guaranteed success, otherwise it wouldn't be research. However in software engineering there tends to be a perception that any engineering problem can be solved given enough work. Questions subjected to empirical studies, however, are often not answered by ingenuity, but rather by analysis of data. There might not be enough data for statistical significance, or there might be too many extraneous variables or methodological errors detected that the results are not meaningful. See Trochim (2007) for excellent coverage of threats to validity. Another point to consider is that an otherwise successful study needs to be well-cited, and 'find its place' in the scientific literature if it is to be truly useful. A study will be more likely to have impact if it uses similar measurement scales and methods as other studies of a similar type. Williams et al. (2005) discuss this in more details.

For companies, an answer to a research question might not require 95% confidence. They may be able to base a decision on a 70% probability of something occurring. Also a company may be satisfied with empirical studies that are simply seeking to gather observations and trends. Success criteria therefore need to be separately defined for both parties in a research collaboration.

In the Mitel-CSER project, neither of our two main empirical studies involved controlled experiments. In one (Herrera, 1999) we explored techniques for conducting usability studies, and in the other (Singer and Lethbridge, 1998) we gathered data in order to generate work patterns. Both studies had largely qualitative outcomes, generating tools or tools improvements, and lessons that could be used in subsequent research. A key sign of success for the company was that the tools we developed were useful to them. The key indicator of success for the researchers was that we were able to publish a significant number of papers.

5. Planning Empirical Studies Projects

In this section we discuss the set of issues that need to be discussed and made part of the project plan as a company-industry empirical research project is established. These include: Justifying the project in the first place, issues that must be agreed between the parties, obtaining ethics approval, staffing the project, working with participants, and analyzing the data.

Table 3 Checklist of activities that should be part of the planning and management process of industry–university collaborations involving empirical studies

Activity	Involves or decided by
• Decision: To use university researchers or in-house employees (refer to Tables 1 and 2 for decision-making information)	Company
• Attracting companies	Researchers
• Decision: Level and type of commitment (finances, resources, timetable, deliverables)	Negotiated
• Decision: How on-going management and risk management will be handled?	Negotiated
• Decision: What is the research focus, what are the goals and what are the research questions?	Negotiated, but may be largely determined by either party
• Decision: What participants will be available and when?	Negotiated
• Decision: What information must be confidential?	Negotiated
• Decision: How will publication of results be handled?	Negotiated
• Decision: Who owns intellectual property?	Negotiated
• Obtain ethics approval	Researchers
• Find researcher team members and train them	Researchers
• Plan the details of work with participants	Researchers
• Plan for data analysis	Researchers
• Evaluate the risks and manage changes	Both parties

A checklist of the activities that should be performed during project planning is presented in Table 3.

5.1. If the Company is Considering Initiating Research: Should it Use University Researchers or Corporate Employees?

As discussed in Sects. 3 and 4, there are many benefits that companies can obtain by involving university researchers, but there are also various risks. If the company is initiating the research, it must first decide whether to instead use its own employees for the research. A university research team will normally involve one or more faculty members and at least the same number of graduate students; since the faculty members' time is split divided among several tasks (teaching, administration and other research), the bulk of the research is often performed by graduate students, under the direction of the faculty members.

The main benefits to using university researchers are that they are a valuable pool of talent, and cost less than in-house employees. University researchers often also have very specific knowledge and research skills that cannot be found inside the companies. The cost of this talent might be so low compared to the potential benefits that very little further analysis is needed. In many countries, graduate students are paid significantly less than company employees. Faculty members might be paid consulting fees for

some of the research, but they tend to spend much additional time on the research that is just part of their normal university duties, paid by the university.

On the other hand if the research is of the type where the company absolutely must have a rapid answer to a question, then there is a high risk in involving graduate students who are prone to take their time completing courses and might want to focus their thesis on another topic. Furthermore, an advantage of using corporate employees is that they tend to have a greater knowledge of the company's products, needs and environment.

In summary, there is no single answer to whether it is better to perform research in-house or involve university researchers: the decision depends on the type of research to be done. In-house employees can work full time and may focus better on the problem, but are normally much more expensive and may lack specific expertise in the area of the research.

5.2. If the University is Considering Initiating: How does it Make Contacts with Companies?

The biggest practical problem in studying work practices is obtaining a good sample of participants. If a university researcher is initiating the project, it might be possible in some cases to conduct a study using participants who are solicited individually (for example they might be asked to fill out a questionnaire on the web on their own time). However, it is usually necessary to work with teams within a company. Hence, participation needs to be obtained from the management of one or more companies.

Finding suitable organizations is the first hurdle. While many researchers or their institutions may have a few companies that are their perennial 'contacts' in industry, empirical researchers should give thought to involving companies of several different types to avoid introducing bias. The companies most likely to be willing to participate are those already involved in research – particularly medium to large companies whose primary business is software or computer products. Much harder to penetrate are companies in other industries that develop specialized software or in-house software, for example, banking and health care. In the past, we have experienced considerable frustration finding suitable managers to contact. Our only advice is that unbiased research often requires considerable effort of this type. We were lucky with the Mitel-University of Ottawa collaboration since both parties sought out each other.

When the university researchers are the ones seeking the contacts, two levels of management must be convinced to participate: Higher management must agree to the involvement of the company as a whole, while first-level managers must agree to the involvement of their teams. In both cases, obtaining and maintaining commitment can be hard. Management will naturally be concerned about the costs of the research, particularly in terms of time. Researchers have to effectively, but realistically, show that there are benefits to the company, which can balance the costs. The costs and benefits presented in Sects. 3 and 4 can be used to make a case.

It is easier to make a case to a company when establishing a long-term relationship. We have found companies are more open to empirical studies when other members of the research team are tackling the company's engineering problems (perhaps using data from the empirical studies).

5.3. Key Success Factors: Mutual Understanding in a Co-operative Relationship

Empirical studies of software engineering involve people studying people. The fundamental requirement for a successful research project relationship is that the two parties, the company based software engineers, and the academic researchers get to know and trust each other. A strong positive social relationship of mutual respect and trust must be established and maintained between the company based manager and the principal researcher. As usual in social matters it really helps if people like each other. This relationship takes time to establish and it may take many meetings spread out over several weeks to develop mutual understanding of the research problem and opportunities for solutions. As Conradi et al. (2003) say, it is important that the researchers have a, 'humble attitude … towards the situation of the practitioners'.

This dialogue must culminate in a research plan that is mutually acceptable to both parties. Since longer-range research work will always play second fiddle to the immediate product development needs of the company, it is vital that the company manager be personally fully committed to supporting and carrying through the project. He or she must see the value and want to carry it forward in order to accept the hindrance to his day-to-day work. At the same time, expectations must be carefully managed. Overly enthusiastic research promises or commitments of company time can lead to fractured relations and harm the project. The project should have a time frame that anticipates research results corresponding to the normal steps in progress of a thesis.

It is also vital that each party understands and respects the agenda and imperatives of the other. This understanding should develop as the dialogue between the two project leaders goes on.

5.4. Issues that Must Be Agreed Between the Company and the Researchers

Once a company has established its willingness to participate, it is important to reach agreement on a number of issues. The formality of the agreements will vary with the size and duration of the research. A very large project requires more detailed negotiations than a small one, particularly if financial support is involved. A company will be interested in the project, but be more willing to participate if its

managers were given a presentation about empirical software engineering and the proposed methods. In such cases, the researcher should treat educating the organization as part of the negotiation process, so they can proceed as partners in the endeavor.

The following are areas where we believe agreements should be established to help ensure the project's success. In the Mitel-CSER project, some of these items were included in a written agreement, but most were just tacit agreements that evolved over time. If we were starting again, we would probably prefer to write down more details, although there is always the danger that developing a more detailed formal agreement (which might have to be approved by lawyers) would cause inflexibility and possibly lengthy delays, thus potentially causing more harm than good.

5.4.1 Level and type of commitment to the project

The first point of mutual agreement should be the level and type of commitment of both parties to the project. Questions to answer are: What is the project's expected duration? How much support (e.g. space, time, equipment) is expected from the company? What kind of results or specific deliverables, if any, are expected from the researchers? Agreement on these issues often forms the basis for agreement on other issues below.

For our project, Mitel has provided financing since its inception, with NSERC (a Canadian Government funding agency) subsequently matching both cash and in-kind contributions. Mitel also provides office space and equipment, although the distance to the company and lack of direct-enough public transportation has meant that graduate students have usually preferred to work on campus. The faculty member has on average spent one morning a week at the company, although at the peak of the research he tended to spent several consecutive days there. Over 80% of the faculty member's research time has been spent analyzing data and writing up the results, an activity not performed at the company site. We have found it important to communicate with the company frequently during these latter activities to ensure that long absences are not interpreted as delays in the research.

In the first year of the project, we established a very ambitious timetable for the research, which later proved to be unrealistic. Project plans developed in subsequent years were somewhat more accurate, but we still had an unrealistic schedule. This was because we did not sufficiently allow for the fact that it might be hard to find appropriate graduate students, that they are delayed by courses, comprehensive exams, and other activities, and that they receive and often accept tempting job offers and therefore drop to part-time status. The main problem with finding graduate students is that most entering graduate students want to create new software, not perform empirical studies. The delays from courses and exams arise because the graduate students feel they have to give 100% of their time to these activities to maximize their marks. We overcame these problems to some extent by hiring

people as research associates – such people have a stronger contractual obligation than graduate students who are merely 'supported' in their studies. Unfortunately market conditions make hiring skilled research associates difficult.

The only real deliverables that Mitel expects are features periodically added to the tool, and regular reports about progress. However our plans were always rather ambiguous regarding the level of quality expected, and we rarely met our target dates (the whole software industry, of course, tends to have this problem). One problem we faced was students and research associates implementing just enough software to test their ideas, but not making the quality of the software high enough so that Mitel could use it on a regular basis after their studies were complete. As discussed earlier, the core TkSee tool was made highly usable, but add-on features created for specific student studies were often never used seriously by Mitel employees.

5.4.2 The decision-making and management process

Since empirical research projects, especially long-duration ones, rarely proceed as initially planned, there need to be agreements about how changes to plans will be made. We believe that an active-risk management approach is needed: At the initial stages, the risks (see Sect. 4) need to be identified and their magnitude estimated. When researchers and company managers meet on a semi-regular basis, both progress and the risk profile should be informally reviewed and changes to the plans agreed.

Risk management was something with which the university researchers had little experience at the start of the Mitel-CSER project. As the project progressed and deviated from the original plans (albeit in parallel with significant success), we did not do a good job of ensuring that both parties clearly understood the reasons for the deviations. University researchers may well be able to learn from the managerial expertise of the company in this regard, just as the company can learn from the technological expertise of the researchers.

At one point we went too far in the opposite direction by regularly updating a detailed project plan. That turned out to be far too time-consuming with not enough benefit. We now believe the kind of regular management needed should involve update and discussion of a very brief progress chart, and a short list of successes, problems and risks.

5.4.3 Access to participants

Both sides need to agree on how many employees will participate in the study and how much time is required from each employee. Sometimes an organization will find it difficult to provide the personnel required by the ideal research design and some compromise may be necessary.

In our project, Mitel agreed to a certain number of employee-hours per year to be devoted to our project, but we did not accurately monitor this, and likely used

somewhat less time than budgeted. A key point for Mitel was that before the researchers initiated meetings with employees, they would check with management to see who was busy with 'critical' or 'deadline' work, and avoid these employees until they had more time.

5.4.4 Confidentiality of data

Some data needs to be kept confidential for corporate reasons; for example a company may not allow highly sensitive information such as source code or defect logs to be taken off-site. Data about individuals needs to be kept confidential for ethical reasons – we will discuss this further below. Data that are not confidential for either of the above reasons can serve as the basis for discussions of the next point, publication of results.

We had to negotiate with Mitel regarding the confidentiality of certain data that revealed aspects of their software's design that needed to be kept a trade secret. We were not able to take Mitel's source code out of company premises: This proved useful in some ways because it encouraged grad students to spend time at the company. However it was also quite inconvenient at times.

5.4.5 Publication of results

It is difficult to predict which results will be sufficiently interesting to publish, particularly before data collection has begun. Understandably, companies are reluctant to give blanket approval to disclosure of information. One solution is to set some ground rules at the beginning, and deal with publications on a case-by-case basis. Although this approach adds a step to the process of writing a paper, it has the benefit of providing researcher with an opportunity to verify their observations and conclusions.

On our case, our papers are reviewed for publication by the company at the same time that peer review occurs. Officially, Mitel could have asked to approve them before initial submission, however we established a good working relationship so that we did not need to be so rigid: Mitel told us the kinds of things they didn't want made public and we wrote in a style that accounted for Mitel's desires. At the same time Mitel recognized that academics often have very short lead-times to submit papers. They never rejected any papers, although they requested a few changes.

Another decision to be made is whether or not to identify the organization in the publication. A company may want its contributions acknowledged, or it may not want to be associated with 'negative' findings. Also, it may not be possible to publish the identity of the company without compromising the anonymity of the participants. This question can be dealt with in using the same approach described above for results. Realizing that anyone could find out from various sources that funded our research, we realized it would have been pointless to not mention Mitel's name. In some paper, the company employees also took a personal stake by becoming authors.

5.4.6 Other intellectual property issues

In addition to publication of results and protecting trade secrets, the two parties need to agree on what will happen if a patentable invention should arise from the research. Achieving agreement in this area can be very time consuming. The degree of sensitivity on the part of the company will depend on whether research results could provide functionality central to their products. In the case of the Mitel project, the benefits accrue to design efficiency. For these to be most valuable they need to be incorporated in commercially available tools and so Mitel has little concern about patents in this case. On the other hand another member of CSER is a software tools company and it has a much greater interest.

The formal CSER agreement acknowledges inventions as belonging to the inventors. Members have a free license to use any tools and techniques that arise from the research within their individual businesses. If they wish to sell products incorporating any CSER inventions then they must separately negotiate a license with the inventor.

A final comment regarding the co-operation of companies: One should keep in mind the possibility of a long-term relationship with the company. After going through the effort of establishing a relationship it will likely be useful to extend it either by performing a series of different studies, each building on the previous, or by performing longitudinal studies where software engineers are followed over many years.

5.5. Obtaining Approval of the Research Ethics Board

It is now considered essential in most countries that any research project involving human subjects should be scrutinized by a Research Ethics Board (REB) before the project gets underway. This is something that social scientists and medical researchers now take for granted, but which is not widely known in engineering. Even projects involving simple questionnaires need to be evaluated.

Research ethics are the subject of Chap. 12 of this book. There are many issues which are particularly important to industrial empirical studies, such as ensuring that management doesn't influence the freedom of participants to not participate or to withdraw, and doesn't see the raw data. Rather than presenting details about the ethical issues themselves here, we will briefly list some points relevant to the management of the ethics approval process.

The most important management issues for the empirical software engineering researcher to do are:

- Become familiar with the REB process at their institution.
- Plan the project with sufficient care that no ethical guidelines are violated. This means writing a proposal document in considerable detail so as to be convincing to the REB – something that might be more time-consuming than anticipated. The most important parts of such a document are the research protocol itself and the informed consent form that must be signed by all participants.

- Plan the project with sufficient time to allow the REB to make its decision, with allowances for possible required changes and resubmission. REBs very often nit-pick about details of proposals.
- Do not start any studies involving people until approval is received.

Long-term projects where the research is opportunistic in the sense that individual studies are planned on an on-going basis, may have to repeat this approval process.

In the early days of the Mitel-CSER project we conducted the work without REB approval out of ignorance, and because there was no formal mechanism for such approval within engineering. That was later rectified; at the same time Canadian research ethics guidelines have been strengthened and harmonized.

5.6. Staffing the Project and Training Researchers

Company-industry empirical research projects will normally involve graduate students and perhaps postdoctoral fellows. As mentioned in Sect. 4, an important difficulty such projects will face is attracting interested researchers.

One technique that may work is involving researchers from the social sciences as collaborators. Many anthropologists and psychologists have developed an interest in, and expertise in, software engineering processes. Such people would not be able to solve engineering problems, and may have a weaker understanding of what they are observing than engineers, but they should know more than the average engineer about human behaviour, work practices, study methodologies and ethics. The work of course is not lessened, but graduate students in these disciplines might be more motivated to perform the detailed data analysis gathered from human subjects involved in empirical studies.

In the Mitel-CSER project we have been fortunate to work with Janice Singer, a scientist at the National Research Council who has a Ph.D. in psychology and has also worked in software development. Our research group has also involved graduate students in psychology from time to time.

It is essential for the entire research team to practice and refine the research methodology before taking it on the road, otherwise many mistakes will be made and data will be lost. Researchers unfamiliar with the techniques discussed in this book will be surprised about how many difficulties can arise. For example the wording of questions must be thoroughly tested to remove ambiguities. Also the process of setting up cameras, recording, transcribing, and coding should be well rehearsed.

In addition to understanding empirical study techniques, researchers should normally spend considerable time in learning about their company. An understanding of corporate culture needs to be established so researchers can effectively interact with the participants and correctly interpret data. The researchers need a basic understanding of key aspects of the participant's work, such as the problem domain, the business context for the application, and the tools and process they are

using. Some of this knowledge can be gained during the study itself, but we have found it more effective to have a learning phase in advance of the study.

5.7. Working with Corporate Employees and Managers

After establishing a research relationship with the company, obtaining ethics approval and training the research staff, the next step is to establish relationships with individual participants. Whether potential participants are willing to participate depends on several factors:

- The type of research: Being watched is of more concern to most people than, for example filling out a survey. Also, long-term or time-consuming research might attract fewer participants.

5.7.1 Whether the participants perceive management to be supportive

We have found it essential that management be enthusiastic about the research and make this clear to their employees. Enthusiasm assures employees that they are not at risk of being penalized for not getting their 'regular' job done while taking time out to participate in the research. Since our research continued for a long period of time, and many employees came and went during this period, management periodically arranged meetings with the employees at which the researchers presented a status report and sought input. However, for ethical reasons, managers should make it clear that participation is completely optional and they are not ordering people to participate.

A technique that we find useful is to use two consent forms. One is signed by the manager, consenting to the participation of his or her staff and assuring them that there will not be any management interference or impact whether or not they participate. A copy of this is given to the participants along with their own consent form.

Whether the participant perceives some benefit to participation: Some participants will enjoy taking time away from their daily work; others may be interested in the research for its own sake or because they feel they may gain something from the results. In our research we always tried to make it clear to employees that we were trying to develop tools that would be helpful to them. It was a concern when our work took longer than expected that some participants might feel let down.

The personality and beliefs of the participants: We have found some employees are more willing to participate than others. In fact, we have had situations where participants actively dissuade us by saying that the work they are doing would not be interesting enough for us to study. Leaving out such people might bias the research, so we tried to encourage the employees to participate while continuing to assure them it was optional.

Empirical research in companies can be mentally intense for researcher and participant. In order to get the most out of the work, the pace should not be rushed. Plenty

of flexibility should be built into the day's schedule and no more than two sessions should be held in any day.

It is also important to understand that software engineers follow a development cycle. This means that they are doing different activities at different times. Finding what software engineers do during design and coding does not necessarily reflect what they do during bug-fixing or requirements gathering. Therefore, data collection has to focus on one aspect of the development cycle, or must extend over several time points to get an overall view of software engineering work.

Another consideration is software engineers' time constraints. Researchers need to find, to the greatest extent possible, data collection methods that do not affect the software engineers' productivity. Unfortunately, it is not always possible to gather key information unobtrusively. When a time commitment is required from software engineers, researchers need to make sure that they get the largest possible return for that time.

5.8. Maintaining the Relationship

Maintaining an industrial research relationship takes continued work. Some of the tactics we suggest are the following:

- Ensure all researchers (both faculty and students) have a regular presence in the company premises, whether or not they are actively conducting studies. The mere fact of being there, working on papers, theses, etc. shows a commitment. Participating in company meetings social events can also help to solidify the relationship.
- Report regularly on research progress, perhaps once every month. Even if not much has happened (as is often the case when academics are in the midst of teaching courses, and working on other matters), at least find something to say.
- Offer to give presentations on various topics. These could include updating employees on the status of the research, or giving a lecture on some topic that might simply be interesting to the company. The company will therefore reap value-added in terms of expertise that they can use to further justify continuing the relationship.

5.9. Planning for Data Analysis

Data analysis is probably the most time-consuming phase of most empirical studies. We will not discuss techniques here, since that is the topic of other chapters. However, we wish to point out that it should, where possible, be carefully planned at the project's start.

6. Concluding Remarks

In this chapter we have discussed many of the issues we have faced when managing university-industry empirical studies of software engineering. Our goal in presenting this information is to present the lessons we have learned, and hence to provide guidance for others undertaking similar studies for the first time. The issues discussed, such as the benefits and drawbacks to be considered, establishing contact with organizations and participants, staffing, and obtaining ethical approval, can be made to work more smoothly through effective planning. We also strongly believe in on-going evaluation and change management of the project as it progresses, particularly considering the risk factors we identified.

Acknowledgements This work was supported by the Consortium for Software Engineering Research (CSER) and the Natural Sciences and Engineering Research Council of Canada (NSERC). We would like to thank Janice Singer of the National Research Council who participated in many of the discussions as we planned our research, and thus contributed many of the ideas in this chapter. We would also like to thank anonymous reviewers for their valuable suggestions.

References

Anquetil, N. and Lethbridge, T.C. (2003), A Comparative Study of Clustering Algorithms and Abstract Representations for Software Remodularization, *IEE Proceedings – Software*, pp. 185–201. Winner of the Mather Premium award.

Anquetil, N. and Lethbridge, T.C. (1999), Recovering Software Architecture from the Names of Source Files, *Journal of Software Maintenance: Research and Practice*, 11, pp. 201–221.

Arisholm, E., Anda, B., Jørgensen, M., and Sjøberg, D.I.K. (1999), Guidelines on Conducting Software Process Improvement Studies in Industry, *22nd IRIS Conference (Information Systems Research Seminar in Scandinavia)*, Timo K. Kakola (ed.), Computer Science and information Systems Reports, Technical Reports TR-21, University of Jyvaskyla, Keuruu, Finland, pp. 87–102, 7–10.

Baskerville R.L. and Wood-Harper, A.T. (1996), A Critical Perspective on Action Research as a Method for Information Systems Research, *Journal of Information Technology*, (11), pp. 235–246.

Beckman, K., Khajenoori, S., Coulter, N., and Mead, N. (1997), Collaborations: Closing the Industry-Academia Gap, *IEEE Software*, 14(6), pp. 49–57.

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

Conradi, R., Dybå, T., Sjøberg, D.I.K., and Ulsund, T. (2003), Lessons Learned and Recommendations from Two Large Norwegian SPI Programmes, *9th European Workshop on Software Process Technology (EWSPT 2003)*, Helsinki, Finland 1–2 September, Lecture Notes in Computer Science 2786, Springer-Verlag, pp. 32–45.

Dittrich, Y. (2002), Doing Empirical Research on Software Development: Finding a Path Between Understanding, Intervention, and Method Development, *Social Thinking: Software Practice*, MIT Press, Cambridge MA, pp. 243–262.

Herrera, F. (1999), *A Usability Study of the TkSee Software Exploration Tool*, *M.Sc. Thesis in Computer Science*, University of Ottawa, http://www.site.uottawa.ca/~tcl/gradtheses/

Lethbridge, T.C., Sim, S., and Singer, J. (2005), Studying Software Engineers: Data Collection Methods for Software Field Studies, *Empirical Software Engineering*, 10(3), pp. 311–341.

Lethbridge, T.C. and Singer, J. (2001), Experiences Conducting Studies of the Work Practices of Software Engineers, in *Advances in Software Engineering: Comprehension, Evaluation and Evolution*, H. Erdogmus and O. Tanir (eds.), Springer-Verlag, ISBN 0–387–95109–1, pp. 53–76.

Lethbridge, T.C. (1994), *Practical Techniques for Organizing and Measuring Knowledge*, *Ph.D. Thesis*, University of Ottawa, http://www.site.uottawa.ca/~tcl/thesis_html/thesis_ToC.html

Lethbridge, T.C. and Anquetil, N. (1997), *Architecture of a Source Code Exploration Tool: A Software Engineering Case Study*, University of Ottawa, Computer Science Technical Report TR-97–07.

Liu, H. and Lethbridge, T.C. (2002), Intelligent Search Methods for Software Maintenance, *Information Systems Frontiers*, 4(4), pp. 409–423.

Mead, N., Beckman, K., Lawrence, J., O'Mary, G., Parish, C., Perla, U., and Walker, H. (1999), Industry/University Collaborations: Different Perspectives Heighten Mutual Opportunities, *Journal of Systems and Software*, 49, pp. 155–162.

Pfleeger, S.L. (1999), Understanding and Improving Technology Transfer in Software Engineering, *Journal of Systems and Software*, 47, pp. 111–124.

Potts, C. (2003), Software-Engineering Research Revisited, *IEEE Software*, 10(56), pp.19–28.

Rombach, D. and Achatz, R. (2007), Research Collaborations between Academia and Industry, *Future of Software Engineering 2007*, L. Briand and A. Wolf (eds.), ICSE 2007, IEEE-CS Press, 2007, pp. 29–36.

Sayyad Shirabad, J., Lethbridge, T.C., and Matwin, S. (2003), Mining the Maintenance History of a Legacy Software System, *International Conference on Software Maintenance (ICSM),* Amsterdam, IEEE Computer Society, pp. 95–104.

Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovi , A., Liborg, N.-K., and Rekdal, A.C. (2005) A Survey of Controlled Experiments in Software Engineering, *IEEE Transactions on Software Engineering*, 31(9), pp. 733–753.

Singer, J. and Lethbridge T. (1998), Studying Work Practices to Assist Tool Design in Software Engineering, *6th IEEE International Workshop on Program Comprehension*. A longer version appears as: University of Ottawa, Computer Science Technical Report TR-97–08, Italy, pp. 173–179.

Somé, S.S. and Lethbridge T. (1998), Parsing Minimizing when Extracting information from Code in the Presence of Conditional Compilation, *6th IEEE International Workshop on Program Comprehension*. A longer version appears as University of Ottawa Computer Science Technical Report TR-98–01, Italy, June, pp. 118–125.

Trochim, W.M.K. (2007), *Research Methods Knowledge Base: Introduction to Validity*, http:// www.socialresearchmethods. net/kb/introval.php, visited April 13, 2007

Williams, L., Layman, L., and Abrahamsson, P. (2005), Establishing the Essential Components of a Technology-Dependent Framework: A Strawman Framework for Industrial Case Study-Based Research, *Workshop on Evidence-Based Software Engineering at the International Conference on Software Engineering (ICSE)* 2005, St. Louis.

Zelkowitz, M. (1995), *Assessing Software Engineering Technology Transfer within NASA, NASA/ GSFC Technical Report, NASA-RPT-003–95, NASA/GSFC, January*. http://www.cs.umd.edu/ users/mvz/pub/assessment.ps

Zelkowitz, M., Wallace, D., and Binkley, D. (1998), *Culture Conflicts in Software Engineering Technology Transfer*, University of Maryland Technical Report. http://www.cs.umd.edu/users/ mvz/pub/expsurvey.pdf.