UNIVERSITY OF OSLO
Department of Informatics

A Systematic Review of Case Studies in Software Engineering

Master of Science Thesis

60 credits

Nina Elisabeth Holt

May 1st 2006

Summary

Research on technology that is to be adopted in an industrial setting must give evidence of relevance to the industry. For this, case studies are important in that they give the opportunity to test technology in realistic surroundings with all the affecting factors.

This thesis is a systematic review of 50 randomly selected articles that report case studies. The objective of the investigation is first of all to get an overview of the state of the art regarding the use of case studies in empirical software engineering. Secondly, the investigation should identify important characteristics of case studies for researchers to give careful considerations when conducting case studies.

The data collected during analysis of these 50 articles, was used to address the following issues: the extent of case studies in empirical software engineering, the quality of reporting case studies, the specification of the case study research method, what researchers call a case study, the affiliation of authors, confusion regarding research methods, and the extent of the use of multiple case studies.

The main findings of this review are:

- Close to twelve percent of the 427 papers searched, use case study as the research method.
- There are great variances in the way of reporting case study results. The general impression is that information is not clearly reported.
- Researchers are not very likely to explicitly state what kind of research method that has been used.
- Case studies are mainly used for two purposes, namely evaluative and demonstrative purposes.
    - Typical characteristics for articles with an evaluative nature are rather high response rates for the six questions in the survey, the reporting of observations of use, and most likely the use of professionals as subjects.
    - Typical characteristics for articles with a demonstrative nature are relatively low response rates for the six questions in the survey, the reporting of technology outcome, and most likely the use of authors of the articles as subjects.
- The majority of the articles with authors affiliated in research communities appear to report technology data.
- The lack of observations of use may be reminiscent of the assertion method.
- The extent of multiple case studies is 22 percent.

The following criteria for case studies in empirical software engineering were suggested: First of all, the author should specify that the research method used is the case study method. The focus in the case study should be use/evaluation of a software technology. Furthermore, the case study should test a technology in an industrial setting. Finally, the technology must be used by others than the researchers themselves (because of no manipulation), preferably by professionals.

There is a need for a specified definition of case studies in empirical software engineering. Additionally, in order to produce results that are easy for reviewers and industry to relate to, there is a need for standards for how to conduct case studies. Use of guidelines would help researchers ensure the quality of the results. Hence, guidelines for assistance through the case study process will be an important device for improving future use of this research method.

The main contribution of this review is in presenting the state of affairs and a characterization of case studies as used in empirical software engineering. Such an overview should be useful for researchers in the work of improving the case study research method. Ultimately, this review should contribute to the work of improving the use of the case study research method in empirical software engineering.

It is hoped that the findings of this research will prove valuable to empirical software engineering, whose main interest is that of investigating the interaction between technology and developers.

Acknowledgements

First of all, I would like to thank my supervisors, Dag Sjøberg and Jo Hannay, for their incredible support, contributions, discussions, guidance and encouragement. I am grateful for their continuous inspirations during the work with this thesis. Thanks to Bente Anda for useful comments, discussions and guidance. Thanks to the students and employees at Simula Research Laboratory for making such a nice work environment!

Last but not least, thanks to my family and friends for their support and encouragement during this period.

Oslo, May 2006
Nina Elisabeth Holt

Contents

List of Tables

1 Introduction

Section 1.1 presents the motivation of this master thesis. Section 1.2 accounts the intention of the research and states the research question that I have investigated. Section 1.3 gives a brief description of the research method used in this thesis. Furthermore, the contributions of the research are described in Section 1.4. The last section gives a description of how the remainder of the thesis is organized.

1.1 Motivation

In Yin's book [27, p. 17] about case studies, he states the following:

> Case study research is remarkably hard, even though case studies have traditionally been considered to be "soft" research, possibly because investigators have not followed systematic procedures.

Zelkowitz and Wallace [29] describe a case study to be an observational research method that is used for monitoring a project and collecting data over time without intervention by the researchers. This is in contrast to experiments, in which the researcher usually has control over various factors. Experiments are done when an investigator can manipulate behavior directly, precisely, and systematically [27, p. 8]. However, it is difficult to conduct experiments on realistic, large-scale projects. By using case studies, on the other hand, the researcher can study real complex projects. In case studies there is a high degree of realism, but less control. The two research strategies are complementary, and hence both are important.

Nevertheless, case studies are often looked down upon as being a weak research method:

> Although the case study is a distinctive form of empirical inquiry, many research investigators nevertheless disdain the strategy. In other words, as a research endeavor, case studies have been viewed as a less desirable form of inquiry than either experiments or surveys. Why is this?
> Perhaps the greatest concern has been over the lack of rigor of case study research. Too many times, the case study investigator has been sloppy, has not followed systematic procedures, or has allowed equivocal evidence or biased views to influence the direction of the findings and conclusions. Such lack of rigor is less likely to be present when using the other strategies — possibly because of the existence of numerous methodological texts providing investigators with specific procedures to be followed. In contrast, few if any texts (besides the present one) cover the case study method in similar fashion. [27, p. 10]

One reason for this perception may be that no standard procedure for carrying out case studies has yet been developed. This may be due to the fact that case studies have not been given as much attention as other research strategies. There are few suggestions as to procedures regarding how to conduct and report case studies, especially when it comes to case study designs and analysis [27, p. xiv].

For software engineering, the case study is a useful research method as software engineering takes place within a context. It is important not to factor out the effect of the context when validating technologies for use in industrial development [21]. However, surveys on the use of research methods in software engineering show a fairly low percentage of case studies [6, 21, 27].

In order to increase the use of case study research, the quality of accomplishment and reporting must enhance. Both Tichy *et al.* [25] and Zelkowitz & Wallace [29] conclude that the software engineering community can do a better job in reporting its results. Here, guidelines are central. Conducting a thorough review of the state of the art regarding the use of case studies is an important prerequisite for making proper quality guidelines. In general, the improvement of research methods is important to empirical software engineering. Zelkowitz and Wallace [29] say that future work must focus on enhancing researchers' ability to report on software engineering experimentation so that research can better assist industry in selecting new technology.

Kitchenham *et al*. [13] present a set of guidelines to follow when conducting research in software engineering. These guidelines are directed towards research in software engineering on a general basis. However, they identify the need for specialized guidelines for different purposes. Furthermore, specified guidelines for case study method and tool evaluation are suggested by Kitchenham *et al.* [10].

As an example of research on a specific research method, Sjøberg *et al.* [23] have contributed with efforts for making guidelines for controlled experiments in software engineering with their in-depth survey. No such work has been carried out specifically on case study research. The major research of significance is by Yin who offers procedures for designing and reporting case studies in his book *"Case Study Research Design and Methods"* [27]. However, Yin's proposal is not specifically directed towards research in empirical software engineering. Rather, it is a general approach for use in any discipline.

As the use of case studies in software engineering has not been given the attention it deserves, I would like to focus on this particular research method.

1.2 Objective

This thesis is a systematic review with the purpose of providing the state of the art regarding the use of case studies in empirical software engineering. I present an overview that characterizes what researchers call a case study in empirical software engineering. On the basis of this overview, other researchers may decide further research for improving the research method.

The objective of the investigation is first of all to get an overview of the present use of case studies in empirical software engineering. Secondly, the investigation should identify important characteristics of case studies for researchers to give careful considerations when conducting case studies.

In order to address these issues, I conducted a systematic review of 50 articles that report on case studies. The data collected during analysis of these articles, was used to answer the following research question:

RQ: *What is the state of the art regarding the use of case studies in empirical software engineering, hereunder:*

SRQ 1 *What is the extent of the use of case studies in empirical software engineering?*
SRQ 2 *What is the general impression of the quality of reporting from case studies? Is data clearly presented?*
SRQ 3 *Do researchers state the type of research method that they have used?*
SRQ 4 *What is called a case study by the authors?*
SRQ 5 *Is there a connection between what kind of data that is reported and the kind of affiliation of the author?*
SRQ 6 *Are case studies confused with other research methods?*
SRQ 7 *What is the extent of the use of multiple case studies in empirical software engineering?*

Additionally, I present a few existing guidelines for accomplishment of case studies.

1.3 Research Method

In order to identify the situation regarding the use of case studies in empirical software engineering, I carried out a systematic review of 50 randomly selected articles that report case studies. The articles were collected among the 5 453 articles scanned and analyzed by Sjøberg *et al.* in their survey of controlled experiments [23]. The sample was analyzed in depth with focus on the following six questions posed by Seaman [20]:

- Who were the subjects?
- What were they doing?
- When was data collected?
- Where did data collection take place?
- Why did they participate?
- How was data gathered?

For each article, I collected data on answers to each of these questions if an answer existed. Furthermore, I performed a count of how many articles that specified 'case study' as the research method used. Included in the data collected was the type of purpose the case studies had in articles, and the type of data reported in the articles. Additionally, the articles were classified according to author's affiliation, similar to what was done in the study of Segal *et al.* [21]. The articles were searched in order to provide an overview of the taxonomy authors use when referring to research methods. I also registered how many articles that reported multiple case studies.

1.4 Contributions

The main findings of this review are:

- Close to twelve percent of the 427 papers searched, use case study as the research method.
- There are great variances in the way of reporting case study results. The general impression is that information is not clearly reported.
- Researchers are not very likely to explicitly state what kind of research method that has been used.
- Case studies are mainly used for two purposes, namely evaluative and demonstrative purposes.
    - Typical characteristics for articles with an evaluative nature are rather high response rates for the six questions in the survey, the reporting of observations of use, and most likely the use of professionals as subjects.
    - Typical characteristics for articles with a demonstrative nature are relatively low response rates for the six questions in the survey, the reporting of technology outcome, and most likely the use of authors of the articles as subjects.
- The majority of the articles with authors affiliated in research communities appear to report technology data.
- The lack of data collection may be reminiscent of the assertion method.
- The extent of multiple case studies is 22 percent.

It is hoped that, the findings of this research will prove valuable to empirical software engineering, whose main interest is that of investigating the interaction between technology and developers. Research on technology that is to be adopted in an industrial setting must give evidence of relevance to the industry. For this, case studies are important in that they give the opportunity to test technology in realistic surroundings with all the affecting factors.

The main contribution of this review is in presenting the state of affairs and a characterization of case studies as used in empirical software engineering. Such an overview should be useful for researchers in the work of improving the case study research method. Ultimately, this review should contribute to the work of improving the use of the case study research method in empirical software engineering.

1.4 Terminology in Thesis

This section introduces terms that is used throughout the thesis.

- **Technology:** Processes, methods, techniques, languages and tools [23].

Purpose of the case studies in the articles:

- **Evaluative purpose:** Articles that report on observations of the use of a technology, including the subjects' perceptions about the technology. Subjects are most often students or professionals.

- **Demonstrative purpose:** Articles that report on the technology outcome. The authors themselves appear most frequently to be the subjects. Exemplifies use of the technology.

Types of data that is reported in the articles:

- **Observation of use:** Data about the use of a technology. This is data about actual observations of use in addition to the developers' perceptions about the technology they used.
- **Technology Outcome:** Data about a technology.

1.5 Structure

Chapter 2 presents relevant background. An overview of related work on research methods is presented in Chapter 3. The research method for this review is described in Chapter 4. Chapter 5 presents findings followed by a discussion in Chapter 6. Chapter 7 discusses the validity of this review. Finally, Chapter 8 concludes and encourages further research on the case study research method in future work.

2 Background

Empirical research is defined as research based on the scientific paradigm of observation, reflection and experimentation as a vehicle for the advancement of knowledge [17, p. 37]. Empirical studies play an important role within both theory-creating and theory-testing research [3, p. 14], and are important input to the decision-making in an improvement seeking organization [26, p. 17].

> Software engineering is a field of practice using methods and tools to solve problems where the solution is a software product. Empirical software engineering is the study of software engineering based on experiences and observations. In empirical software engineering one attempts to identify and establish a scientific approach for software engineering, which comprises of a set of research methods, theories, terminology, and a collection of experiences and observations.
>
> (Sørumgård 1997 referenced by Arisholm [3, p. 12])

Section 2.1 presents ways of how to distinguish between research methods. Section 2.2 describes common research methods. Finally, Section 2.3 presents challenges that case studies meet in empirical software engineering.

2.1 How to Distinguish Between Research Methods

Yin [27, p. 15] says that "the case study, like other research methods, is a way of investigating an empirical topic by following a set of pre-specified procedures". Other ways are, for example, experiments and surveys. Each method has particular advantages and disadvantages, depending on three conditions [27, pp. 5-9]:

- The *type of research question* posed.
- The *extent of control* an investigator has over actual behavioral events.
- The *degree of focus on contemporary* as opposed to historical phenomena.

Mohagheghi adds the following factors that can be used to distinguish the approaches [17, p. 40]:

- The ease of replication: lowest in case study and highest in experiments according to [Wohlin00].
- The risk of intervening: highest for case studies and lowest for surveys.
- Scale: experiments are "research-in-the-small", case studies are "research-in-the-typical" and surveys that try to capture a larger group are "research-in-the-large" [Kitchenham95].
- Cost: formal experiments are costly, have limited scope and are usually performed in academic environments. Industry does not have time or money to spend on experiments. [1]

[1] When it comes to costs, Simula Research Laboratory has successfully developed an approach where consultants from the industry are paid for participation [24].

2.2 Research Methods

This section describes common empirical research methods used in software engineering. Additionally, 'lessons learned' and 'assertions' are also described as they could be confused with the case study as a research method. Finally, the differences between the case study method and controlled experiments are discussed in Section 2.2.6.

Dybå [4, p. 58] gives the following main characteristics of the three main commonly used research methods in software engineering:

> So, while an experiment deliberately divorces a phenomenon from its context and a survey's ability to investigate the context is limited, the case study deliberately aims at covering the contextual conditions.

2.2.1 Case Studies – Research in the Typical

General Definitions

Yin [27, pp. 13-14] defines a case study as follows:

1. A case study is an empirical inquiry that
   - *investigates a contemporary phenomenon within its real-life context, especially when*
   - *the boundaries between phenomenon and context are not clearly evident.*

2. The case study inquiry
   - *copes with the technically distinctive situation in which there will be many more variables of interest than data points, and as one result*
   - *relies on multiple sources of evidence, with data needing to converge in a triangulating fashion, and as another result*
   - *benefits from the prior development of theoretical propositions to guide data collection and analysis.*

Yin states that case studies, like experiments, can be exploratory, descriptive or explanatory.

Yin also says that case study as a research method is favored when there is a "how" *or* "why" question and when the relevant behaviors cannot be manipulated. The contribution of case studies is through *analytical generalization (*e.g., generalization through theory, surface similarity, ruling out the irrelevancies, … [22, pp. 341-373; 27, p. 32]*)*, rather than *statistical generalization* (e.g., random sampling), where theories are expanded and generalized, although the motive of a case study may also be a simple presentation of individual cases.

Single- and multiple-case studies are two variants of case study designs. There are four types of case study designs: Single-case holistic design, single-case embedded design, multi-case holistic design and multi-case embedded design. Holistic means single-unit whereas embedded implies multiple units [27, pp. 42-45].

Prejudices Against the Case Study Research Method

Yin introduces his book [27, p. xiii] with the following claim:

> The case study has long been (and continues to be) stereotyped as a weak sibling among social science methods. Investigators who do case studies are regarded as having downgraded their academic disciplines. Case studies have similarly been denigrated as having insufficient precision (i.e. quantification), objectivity, or rigor.

There are several explanations to this perception. One reason may be the confusion of case study teaching [27, p. 10] with case study research, where case study material is being deliberately altered. This is not a part of case study research.

However, according to Yin, the greatest concern has been over the lack of rigor of case study research. This may be explained by the few, if any, texts that provide researchers with procedures to follow when conducting case studies.

Another concern about case studies can be expressed by the question: "How can you generalize from a single case?"

Yin [27, p. 10] provides the following answer:

> The short answer is that case studies, like experiments, are generalizable to theoretical propositions and not to populations or universes. In this sense, the case study, like the experiment, does not represent a "sample", and in doing a case study, your goal will be to expand and generalize theories (analytic generalization) and not to enumerate frequencies (statistical generalization).

Confusion with data collection methods like ethnography or participant-observation may be the reason for complaints about case studies in that they take too long and result in massive unreadable documents [27, p. 12].

Case Studies in Empirical Software Engineering

Like Yin, Mohagheghi [17, p. 39] categorizes case studies to be used in two types of strategies, namely qualitative and quantitative strategies. Case studies as a *qualitative strategy* explore in depth, a program, an activity or process over a period of time. Kitchenham [11] describes a qualitative case study as "A feature-based evaluation performed by someone who has used the method/tool on a real project". Case studies as a *quantitative strategy* are conducted to investigate quantitatively a single phenomenon within a specific time frame [17, p. 39]. More specifically: "An investigation of the quantitative impact of methods/tools organized as a case study" [11]. The quantitative evaluation method is based on the assumption that you can identify some measurable properties of your software product or process that you expect to change as a result of using the methods/tools you want to evaluate [11].

In [12] a case study evaluation exercise is defined to be one where a method or tool is tried out on a real project.

Furthermore, Wohlin *et al.* [26, p. 12] adds that within software engineering, case studies should not only be used to evaluate how or why certain phenomena occur, but also to evaluate the differences between, for example, two design methods. Hence, case studies are also appropriate when it comes to comparisons of technologies in order to find the best technology.

The case study's unique strength is its ability to deal with a full variety of evidence, including direct observation of the events being studied and interviews of the persons involved in the events [27, p. 8]. Dybå [4, p. 66] says that the strengths of the case study are its way of capturing "reality" in greater detail and analyzing more variables than is possible using other methods. Furthermore, the development is going to happen regardless of the needs to collect experimental data, so the only additional cost is the cost of monitoring the development and collecting this data [29].

Mohagheghi [17, p. 44] refers to the following ways, proposed by Yin, of improving the validity of case study research:

- Use multiple of sources in data collection and have key informants to review the report in composition to improve construct validity.
- Perform pattern matching (comparing an empirically based pattern with a predicted one especially for explanatory studies) and address rival explanations in data analysis to improve internal validity.
- Use theory in research design in single case studies to improve external validity.

Reporting of Case Studies

Yin emphasizes the importance of the following procedures when reporting a case study:

- Start early, before collecting and analyzing the data, to write the bibliography, methodological section, and descriptive data about the cases being studied.
- Consider case identities: real or anonymous?
- Let peers, participants and informants review the draft of the case study. Corrections made through this process will enhance the accuracy of the case study, hence increasing the construct validity of the study.

2.2.2 Experiments – Research in the Small

Case studies imply low control but high realism. Experiments on the other hand are normally done in a laboratory environment, which provides a high level of control [26, p. 9]. Experiments are preferred when an investigator can manipulate behavior directly, precisely and systematically [26, p. 14]. Subjects are assigned to different treatments at random. The objective is to manipulate one or more variables and control all other variables at fixed levels [26, p. 9].

Arisholm [3, p. 17] says that in order to impose full control, formal experiments are often small, which is a problem when you try to increase the scale from the laboratory to a real project.

Experiments sample over the variables that are being manipulated, while case studies sample from the variables representing the typical situation [17, p. 41]. As such, case studies are valuable because they incorporate qualities that an experiment cannot visualize, e.g. scale, complexity, unpredictability, and dynamism [26, p. 13].

2.2.3 Surveys – Research in the Large

In surveys, qualitative or quantitative data are often gathered by way of interviews or questionnaires. Respondents belong to a representative sample from the population being studied. The results from the survey are then analyzed to derive descriptive and explanatory conclusions and finally statistically generalized to the population from which the sample was taken [26, p. 8].

Surveys can try to deal with phenomenon and context, but their ability to investigate the context is extremely limited [27, p. 13]. Both surveys and case studies can be classified as both qualitative and quantitative. In case of a survey, the classification depends on the design of the questionnaire (which data is collected and if it is possible to apply any statistical methods). The difference between surveys and case studies is amongst others, that a survey is done in retrospect (or prior to execution of a project, based on previous experience and hence conducted in retrospect to these experiences) while a case study is done while a project is executed [26, p. 8].

2.2.4 Lessons Learned

Lessons learned is an historical method that concerns projects that have already been completed, whereas the case study is an observational method that concerns the collection of data from projects as they evolve. Lessons learned-documents examine qualitative data from completed projects; typically after the completion of a large industrial project. Such a study can be used to improve future developments [30, p. 238]. Additionally, lessons learned may indicate various trends, but cannot be used for statistically validating the results [29].

2.2.5 Assertions

Assertions are usually presented as example uses of a new technology where the developer of the technology demonstrates its value, rather than to objectively assess its relevance compared to competing technologies. It is described as *ad hoc* validation. This research method provides insufficient validation. However, it does provide basis for future experiments. Zelkowitz and Wallace [29] point out that such experimentation should be viewed as potentially biased since the goal is not to understand the difference between two treatments, but to show that one particular treatment is superior.

2.2.6 Distinctions between Case Studies and Controlled Experiments

This section presents the differences between the case study method and controlled experiments, as the two important characteristics comparison and control needs to be considered in both methods.

Comparison

Comparison is introduced in case studies conducted in software engineering. Case studies are useful in answering a "which is better" question [4, 10, 26, 27, 29]. However, comparison is above all the characteristic of controlled experiments. Therefore, it is relevant to clarify the difference between case studies and controlled experiments.

In the outset, comparisons would actually conflict the fact that the case study is characterized as an observational method with no manipulation. Nevertheless, in order to see the effectiveness of a technology, it should be possible to make comparisons against other technology. This makes it possible to find the best technology in a given context. The distinguishing factor between controlled experiments and case studies is in this case the contextual factor.

As concluded in the previous paragraph, case studies can be comparative. Multiple case studies, for instance, investigate technologies in relatively similar or varied contexts and compare these. Each case should be carefully selected so that it either predicts similar results (a literal replication) or predicts contrasting results but for predictable reasons (a theoretical replication) [27, p. 47].

Actually, even single case studies can be comparative in that the effect of a technology is studied by comparing it with earlier projects where this technology was not used. The researcher would compare results against a baseline: company baseline, sister project as baseline or apply method to a random selection of individual product components [10]. Kitchenham *et al.* [10] say that the case study by nature is comparative "contrasting the results of using one method with the results of using another". This makes sense in spirit of empirical software engineering where the objective is to find *what works best* among developers and software technology. In situations like the former, it may be a bit unclear what the new technology is compared to; i.e. what was the situation before the new technology was used. However, this may also be reality for controlled experiments where the subjects bring along their implicit understanding of how a task should be solved.

This means that case studies can be comparative without manipulation by the researcher. The comparative characteristic is OK, however not in one and the same case study because it implies too much manipulation. Comparative studies are better defined as single case studies in a multiple case study setting. An analogy can be drawn to replicated experiments [27, p. 47].

Manipulation/Control

Yin says that the case study method is favored when the relevant behavior cannot be manipulated; i.e. the degree of control the researchers have when conducting tasks in the project [27, p. 7].

In controlled experiments, the context is controlled in that selected variables are given specific treatments. Experiments sample over the variables that are being manipulated, while case studies sample from the variable representing the typical situation [17, p. 41]. If a case study in empirical software engineering would have some kind of treatment, it would be difficult to separate the case study from a controlled experiment. Thus, case studies should not have treatments. Nevertheless, case studies must to some extent involve manipulation. For example, the researcher should be allowed to ask for a particular technology to be used in an organization.

A case study is based on observations of technology (objective: to find technology that gives improvements when used in industry). However, what if a researcher wishes to test some new technology in real life and requests a company to use this technology without the interference of the researcher? Can we consider a research strategy as a case study if the researcher has some kind of initial control but after the initiation only observes the situation?

Initial control must be said to be an important element of a case study definition. If case studies only included cases where a technology is tested in industry, this would be a very inefficient research method in the sense that researchers who want to test a particular technology would have to wait until some company actually makes use of that technology. Therefore, initial control should be a part of the definition. This is supported by [4, pp. 58-59], where initial control is regarded as a part of the case study research method.

2.3 Challenges for Case Studies in Software Engineering

Anda [1, p. 15] says that: “Case studies are the most common kind of study carried out in cooperation with industry in empirical software engineering research”. In spite of this, the use of case studies in empirical software engineering meets some skepticism. This section concentrates on the main reasons for this perception.

Zelkowitz and Wallace [28] found a share of ten percent regarding the proportion of case studies in software engineering. Further, Segal *et al.* [21] found that 13 percent of the papers assessed used the case study method. Glass *et al.* [6] found that 2.2 percent reported on case studies. Ramesh *et al.* [19] conducted an analysis of 628 papers published in 13 major computer science journals where the case study as a research method only makes 0.16 percent.

According to Mohagheghi [17, p. 43], who comments on the results of Ramesh *et al.* [19], industrial case studies are rare in software engineering because there is hard access to critical information. Another reason may be that data collection may take place over a

long period. Additionally, results are difficult to generalize and harder to interpret due to the impact of context. Finally, unexpected events like project stop or changes in personnel or environment may affect data collection.

There is skepticism in the industry regarding obtaining results from experimentation that is conducted at universities. The skepticism may be due to the following concerns [8, p. 136]: Firstly, the industry does not feel an *ownership* to the research (the not-invented-here syndrome). Secondly, the environments of the laboratories at universities and industrial target environments differ, causing the feel of a certain *distance*. To counter the skepticism in industry, Arisholm *et al.* [2] present guidelines for conducting case studies based on six industrial case studies. In order to address the issue of critical information, the guidelines of Arisholm *et al.* suggest that a confidentiality agreement with the organization should be signed. The organization should also read and accept the publications before they are submitted. In this way, organizations are given control of presentation of information.

Another commonly used argument against field studies is the missing opportunity for replication [21]. Then again, as Segal *et al.* [21] emphasize, this is the reality for software engineering activities in the real world, who additionally say the following about replication:

> Validation of such studies can be based not on replication of the study but on replication of the interpretation: the question to ask is, would other researchers from the same scientific cultural tradition as the original researcher(s) and given the same data, come to the same conclusions?

Nevertheless, although a case study cannot be generalized to every possible situation, the purpose of the case study might be to explore ways of building better effort prediction models for a given type of organization. The actual prediction model based on the local effort and product data may not be valid outside the project or organization, but the results are still useful from the software organization's point of view. Thus, the fact that case studies cannot be generalized to every possible situation may not necessarily be a problem [3, p. 19].

The survey conducted by Zelkowitz *et al.* [30] provides insights into how the research and industrial communities differ in their approach toward technology innovation and technology transfer:

> In general, the methods used by the research community can be considered as exploratory. Industry, on the other hand, wants methods that work, so their techniques are more confirmatory, showing that a given method does indeed have the desired properties.

As researchers produce papers outlining the values of new technology without providing good scientific validation, industry often ignores these papers due to lack of empirical justification of the effectiveness in making their job easier.

Hence, researchers must provide sufficient evidence in order to convince the industry what actual benefits use of a technology would be. This means thorough validation and careful reporting. As the case study research method is such an important empirical research method, it needs to be standardized in order to be appreciated as a valuable research method. The rest of this thesis will therefore investigate state of the practice regarding use of case studies in empirical software engineering.

3 Related Work

A few investigations that include research on case studies in software engineering have been undertaken. This chapter summarizes these efforts. Some of these studies cover case studies as one of several experimental models. Others do not include case studies. However, these are still of relevance to this thesis, due to the structure of the studies and the characteristics that have been measured. An overview of the related work can be found in Table 1.

Sections 3.1 to 3.5 give a description of the related work. A summary of the related work is provided in Section 3.6. Identified needs for future research and a presentation of the direction of the review I have undertaken are also included in this final section.

Table 1 Surveys of Empirical Studies in Software Engineering[2]

	(Tichy *et al.* [25])	**(Zelkowitz *et al.* [28])**	**(Glass *et al.* [6])**	**(Segal *et al.* [21])**	**(Sjøberg *et al.* [23])**	**This thesis 2006**
Purpose	Compares the extent of empirical studies in computer science with other fields.	Classifies studies in SE and validates the taxonomy of empirical studies proposed by the authors.	Surveys topic, research approaches, research methods, reference disciplines and level of analysis.	Surveys topic research approaches, methods, reference disciplines and level of analysis, units of analysis and authors.	Surveys topics, subjects, tasks, environments, and internal and external validity of controlled experiments in SE.	Surveys the use of case studies in ESE.
Scope	Comp. Sci, incl. SE	SE	SE	ESE	SE	ESE
Journals	ACM (random publications), TSE, PLDI Proc., TOCS, TOPLAS	ICSE Proc, IEEE Software, TSE	IEEE Software, IST, JSS, SP&E, TOSEM, TSE	The journal Empirical Software Engineering	EASE, EMSE, ICSE, IEEE Computer, IEEE Software, ISESE, IST, JSME, JSS, METRICS, SP&E, TOSEM, TSE	EASE, EMSE, ICSE, IEEE Computer, IEEE Software, ISESE, IST, JSME, JSS, METRICS, SP&E, TOSEM, TSE
Sampling of papers	1991-1994, one to four volumes per journal, random selection of work published by ACM in 1993	All papers in 1985, 1990 and 1995	Every fifth paper in the period 1995-1999	All papers between 1997 and 2003	All papers in the period 1993-2002	50 papers randomly selected among the papers scanned and analyzed by Sjøberg *et al. [24]*
Number of investigated papers	403	612	369	119	5453 papers scanned, 103 papers analyzed in depth	427 papers scanned, 50 papers analyzed in depth

[2] This table is an extended version of Figure 1 in [23].

3.1 Experimental Evaluation in Computer Science: A Quantitative Study

Tichy *et al.* [25] conducted a survey of 400 articles motivated by their subjective impression that experimental evaluation often is neglected in computer science research.

They found that in a random sample, more than 40 percent of articles about new designs and models completely lacked experimentation. Of the journals related to software engineering, the fraction was higher; more than 50 percent lacked experimentation.

Additionally, they found that only 30 percent of computer science papers and 20 percent of software engineering papers satisfied the (rather mild) criterion of devoting one fifth or more of the space in the papers to experimental validation.

Their findings suggest that computer scientists publish relatively few papers with experimentally validated results, which appears to be a serious weakness in computer science research. Finally, they encourage researchers to produce results that are grounded in evidence.

3.2 Experimental Validation in Software Engineering

Zelkowitz and Wallace [28] conducted a survey on experimental models for validating technology. By this study, they wanted firstly, to determine how well the computer science community is succeeding at validating its theories, and secondly, to determine how computer science compares to other scientific disciplines.

They developed taxonomy for software engineering experimentation that describes the following twelve validation methods: static analysis, lessons learned, legacy data, literature search, field study, assertion, case study, project monitoring, simulation, dynamic analysis, synthetic and replicated. Additionally, a significantly amount of the papers were categorized as papers with no experimentation (papers describing a new technology that contained no experimental validations). The list was not meant to be an ultimate list, rather as a good starting point for understanding software engineering experimentation. The study examined how these approaches have been used.

Of the 612 papers assessed, where 50 were judged to be “not applicable”, 562 papers were examined. These were published in IEEE Transactions on Software Engineering, IEEE Software and the proceedings from International Conference on Software Engineering from 1985, 1990 and 1995. Each paper was classified according to the data collection method used to validate the claims in the paper. They distinguished between data used as a demonstration of concepts and true attempts at validation of the results.

Zelkowitz and Wallace state among their quantitative findings that too many papers have no experimental validation (one third of the papers) at all. However, the percentage dropped from 1985 to 1995 which seems to indicate improvement. Among the papers that did have a form of validation, they claim that too many papers used an informal (assertion) form. Researchers use lessons learned and case studies in about ten percent of

the studies, while the other techniques are used only sporadically. About five percent relied on the simulation method, while the remaining techniques were used in one to three percent of the papers. They also found that terminology is not used in a consistent manner.

The qualitative findings suggest that authors often fail to state their goals clearly or to point to the value that their method or tool adds to the experimentation process. Additionally, authors often fail to state how they validate their hypotheses and use terms very loosely.

3.3 Research in Software Engineering: An Analysis of the Literature

Glass *et al.* [6] seek to give an objective description of the state of software engineering research by examining 369 papers in six leading software engineering journals in the period 1995 to 1999. The papers were categorized according to topic, research approach, research method, reference discipline and units of analysis.

They found that SE research is diverse in topic; however as remarked by Segal *et al.* [21]:

> … though a closer look at their results shows that less than 3 % of the papers were on organizational and societal topics. It appears that the term 'broad' refers only to technical topics.

Most of the papers were placed in the category 'Systems/software concepts' (54.8 percent) where the subcategory 'methods/techniques' (18.2 percent) made the largest part.

Regarding research approach, it appears that the largest part of the papers belonged to the category 'Formulate' (55.3 percent). Only 13.8 percent were evaluative. This is consistent with Tichy *et al.* [25] who commented the lack of experimental evaluation in Computer Science publications in the early 1990s.

Findings show that the most frequent used research methods are those concerning conceptual analysis and concept implementation. The authors emphasize the surprisingly low cut of, amongst others, case/field studies.

Regarding reference disciplines, 98 percent of the papers did not have references to other fields. An interesting finding is that SE research is mostly about technical, computing-focused issues, and rarely about behavioral concerns.

Based on their results, Glass *et al.* [6] raise the questions about broadening research approaches and methods; for instance whether case and field studies would provide richer and more valuable findings for SE research or whether increasing amounts of evaluation would be beneficial, particularly in improving the rate of technology transfer in the field.

Glass *et al.* [6] encourage future research to use the classification scheme presented in their study when writing abstracts and selecting keywords: "Such a practice would aid other researchers immeasurably in assessing the relevance of published research to their own endeavors".

3.4 The Type of Evidence Produced by Empirical Software Engineers

The work of Glass *et al.* [6] did not include the journal Empirical Software Engineering, which is assumed to hold a great amount of empirical software engineering research. Because of this, Segal *et al.* [21] carried out a similar classification on papers that are published by this journal. Their paper reports on the nature of the evidence published between 1997 and 2003 in the journal of Empirical Software Engineering, using the taxonomy developed by Glass *et al.* [6], but adding 'units of analysis' and 'authors' to the classification scheme.

Investigations of the following research questions were conducted: what is the prevalence of case and field studies of software engineering practice? Is there a wide variety in the types of evidence reported in the field of empirical software engineering?

Their main findings are quoted below:

> We found that the research was somewhat narrow in topic with about half the papers focusing on measurement/metrics, review and inspection; that researchers were almost as interested in formulating as in evaluating; that hypothesis testing and laboratory experiments dominated evaluations; that research was not very likely to focus on people and extremely unlikely to refer to other disciplines.

Segal *et al.* [21] discusses their findings in the context of making empirical software engineering more relevant to practitioners.

Another interesting finding is that authors that come from research institutions clearly predominate. Segal *et al.* [21] reports that only eleven percent of the authors come from industry. Furthermore, it was found in the same study that 13 percent of the papers used case study as the research method.

Glass *et al.* [6] found that 13.8 percent of the papers featured evaluation, whereas Segal *et al.* [21] found that 53 percent of the papers in Empirical Software Engineering did the same.

3.5 A Survey of Controlled Experiments in Software Engineering

Sjøberg *et al.* [23] conducted a review of controlled experiments in software engineering. The controlled experiments were collected from nine journals and three conference proceedings from the years 1993 to 2002. Of the 5 453 articles that were read, 103 articles (1.9 percent) were found to report on a total of 113 controlled experiments.

The study focuses on technology, subjects, tasks, type of application systems, and environments in which the experiments were conducted. Additionally, data on experiment replication, and internal and external validity were also collected and discussed.

The largest categories regarding topics are software lifecycle/engineering (49 percent) and Methods/Techniques (32 percent) caused by the large number of experiments on inspection techniques (36 percent) and object-oriented design techniques (eight percent).

It was found that 87 percent of the subjects were students whereas nine percent were professionals. Actually, almost 50 percent of all subjects in software engineering are students.

They identified tasks performed by the subject according to the following categories: plan (ten percent), create (20 percent), modify (16 percent), and analyze (54 percent). Duration of task was provided in some manner in almost 80 percent of the papers. However, specific duration data pr subject was only reported in 36 percent of the experiments.

In 75 percent of the experiments, the applications were constructed for the purpose of the experiment or were student projects. Commercial applications were used by 14 percent.

Internal validity was reported in 63 percent and external validity in 69 percent of the experiments.

3.6 Summary

As we have seen in this chapter, several studies have been conducted on research methods used in software engineering. The surveys express a general need for an increase in empirical validation in addition to a more structured way of reporting research. Due to the importance of technology transfers, the case study as a research method seems to be of particular interest to the industry when choosing new technologies.

The majority of the surveys I have referred to in this chapter report on several types of research methods. Like Sjøberg *et al.* [23] however, the present study is an in-depth study of a specific research method.

A difference between this study and the studies I refer to is that I provide a state of the art regarding the use of a *specific* research method, namely case studies in empirical software engineering. I provide an overview that characterizes what researchers call case studies. The other studies survey the papers in order to provide state of the art with regards to various characteristics which are classified and quantified.

This thesis contributes to the ongoing work of improving the use of the case study research method.

4 Methodology

Empirical software engineering is described as follows by Anda [1, p. 12]:

> Empirical software engineering is the study of software engineering based on observations and experiences. […] The main goal of empirical studies is to enable understanding and to identify relationships among different factors. The studies should be conducted and reported in such a way that practitioners, who are the audience for the research, are able to understand our theories and findings in the context of their work and values.

Section 4.1 describes the research method I have used in the thesis. Section 4.2 describes how the selection of articles was identified. Finally, Section 4.3 describes how the data was collected and analyzed.

4.1 Research Method

As the purpose of this research is to describe the current practice for case studies applied in empirical software engineering, *a systematic review* was chosen as the research method. Before examining the selection of articles, I carried out literature investigations about the case study as a research method, existing proposals on how to carry out case studies, and existing surveys on research methods in software engineering.

Kitchenham [14] describes a systematic review as the "means of evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest". Furthermore, she says that the aim of systematic reviews is to present a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology. A systematic review must be undertaken in accordance with a predefined search strategy [14].

According to Kitchenham [14], the major advantage of systematic reviews is "that they provide information about the effects of some phenomenon across a wide range of settings and empirical methods".

The following is important features of systematic reviews [14]:

- Systematic reviews start by defining a review protocol that specifies the research question being addressed and the methods that will be used to perform the review.
- Systematic reviews are based on a defined search strategy that aims to detect as much of the relevant literature as possible.
- Systematic reviews document their search strategy so that readers can access its rigour and completeness.
- Systematic reviews require explicit inclusion and exclusion criteria to assess each potential primary study.

- Systematic reviews specify the information to be obtained from each primary study including quality criteria by which to evaluate each primary study.
- A systematic review is a prerequisite for quantitative meta-analysis.

The research I have conducted qualifies as both qualitative and quantitative due to the collection of large amounts of qualitative data, which I have quantified by statistical analysis. The investigation is descriptive in that it reports the state of the art for case studies in empirical software engineering. It is also normative as it suggests important concerns that should be carefully considered when conducting case studies.

4.2 Identification of Articles that Report on Case Studies

4.2.1 Target Population

In order to describe the present state of conducting and reporting case studies, it is necessary to select articles that are representative for research in the field of empirical software engineering. The material consists of data from 50 articles that were randomly collected from the collection of 5 453 articles of the twelve journals/conferences scanned and analyzed by Sjøberg *et al.* in their survey of controlled experiments [23]. The journals and conferences are considered by Sjøberg *et al.* to be "leaders in software engineering". Editorials, prefaces, article summaries, interviews, news, reviews, correspondence, discussions, comments, reader's letters, and summaries of tutorial, workshop, panels, and poster sessions were excluded from the search.

The journals are: ACM Transactions on Software Engineering Methodology (TOSEM), Empirical Software Engineering (EMSE), IEEE Computer, IEEE Software, IEEE Transactions on Software Engineering (TSE), Information and Software Technology (IST), Journal of Systems and Software (JSS), Software Maintenance and Evolution (SME), and Software: Practice and Experience (SP&E). The conferences are the International Conference on Software Engineering (ICSE), the IEEE International Symposium on Empirical Software Engineering (ISESE), and the IEE International Symposium on Software Metrics (METRICS).

4.2.2 Criteria for Inclusion

According to Kitchenham [14], a systematic review involves predefined inclusion criteria. However, it is not always possible to predefine the inclusion criteria because of the need for investigating what attributes to consider in the review. In this review, such a pre-study was conducted. The pre-study consisted of analyzing ten articles reporting case studies, where the inclusion criterion was a preliminary operational definition for case studies in empirical software engineering, derived from the general definition of Yin [27, pp. 12-15].

However, few articles satisfied the conditions of such a definition, and it turned out that this approach would not reflect the current state of affairs as to what the community call case studies. Thus, in order to provide an overview of what researchers call a case study in empirical software engineering, the working inclusion criteria for this review was that articles were included if they contained the words 'case study' or 'case studies' as descriptions for the research method followed. No general evaluation as to whether the articles are case studies according to the definition of Yin [27, p. 12-15] is done. Rather, I present the state of affairs and a characterization of case studies as used in empirical software engineering.

4.2.3 Procedure for Random Selection

This section describes the procedure for selecting the 50 articles that report on case studies for use in the analysis.

The following procedure was used in the process of identifying 50 articles:

1. Assign a unique number to each of the 5 453 articles.
2. Use MatLab in order to randomly pull 100 numbers.
3. With help of Acrobat Reader, search for the terms ‘case study’ and ‘case studies’ in the chosen 100 articles in order to get the 50 first that report on case studies.
4. Continue from step 2 until 50 articles are collected.

In total, 427 articles were PDF-searched.

4.3 Analysis of the Articles

This section describes how the articles were analyzed in order to address each of the seven sub-research questions (see Section 1.2).

The data extracted from the 50 randomly selected articles from a selection of 5 453 articles is stored in Excel tables. Additionally, a catalogue of the 427 articles in searchable PDF-format was generated. I used simple descriptive statistics on the collected data and illustrated points of interest with examples from the raw data.

The total number of articles (427 articles) that was necessary to search in order to collect the sample of 50 articles that reported on case studies was used to answer **SRQ 1** about the extent of case studies in software engineering.

In order to address **SRQ 2** regarding quality of reporting, the sample was analyzed in depth with focus on the following six questions provided by Seaman [20]:

- **Who** were the subjects?
  - Age, gender, nationality
  - Education, work experience
- **What** were they doing?
  - Job descriptions
  - Current projects
- **When** was data collected?
  - Time of day and year
  - How did it fit into their day?
- **Where** did data collection take place?
  - Physical surroundings
  - Geographical location
- **Why** did they participate?
  - Motivations, both individual and organizational
- **How** was data gathered?
  - Details of methods – recording, format, who was present, etc

For each article, I extracted data on answers to the question if an answer existed.

Furthermore, I performed a count of how many articles that specified case study as the research method used (**SRQ 3**). The articles were searched for specification of research method in titles, abstracts, keywords, and in the main bodies (explicitly stated).

SRQ 4, regarding what is called a case study by the authors, was addressed by collecting data about what purpose the case study had in the article and what type of data that was reported from the case studies.

Additionally, the articles were classified according to author's affiliation, similar to what was done by Segal *et al.* [21]. The articles were categorized as follows: *Research*, *Industry*, and *Research & Industry*. I also examined the relation between the author's affiliation and the type of data that the article reported (**SRQ 5**): *Observation of Use*, *Technology Outcome*, and articles that report both data from categories.

The articles were searched in order to provide an overview of how authors refer to research methods (i.e., taxonomy). The following terms were considered: 'Experience report', 'Lessons learned', 'Case study', 'Field study', and 'Action study'. This data was collected as a mean to answer **SRQ 6** about confusion around research methods.

I collected data about multiple case studies in order to address **SRQ 7** about the extent of the use of multiple case studies in empirical software engineering.

The articles are analyzed by me only (see Chapter 7 about threats to validity).

5 Results

This chapter presents and discusses the results of the review I conducted with the purpose of answering the research question presented in Chapter 1:

> **RQ:** *What is the state of the art regarding the use of case studies in empirical software engineering?*

The research question is further split into six sub questions that will be paid attention to in Sections 5.1 to 5.7. Finally, the findings are summarized in Section 5.8.

The following sections use articles, which have been analyzed in this review, as illustrating examples. Each article is assigned an ID, in the format A#, which will be used when referring to the articles. The data extracted during the analysis of these articles can be found in Appendix A to Q. However, the appendices will be provided on request.

5.1 Proportion of Case Studies

The results of this section are based on data collected in order to address sub-research question 1:

> **SRQ 1:** *What is the extent of the use of case studies in empirical software engineering?*

Of the 427 articles assessed, 50 were identified to report on case studies according to the selection criteria presented in Section 4.2.2. This indicates that the extent of case studies in empirical software engineering is close to twelve percent.

The correct way of providing the extent of case studies is to collect a specified number of articles which then are searched for occurrences of case studies. The answer to SRQ 1 is based on the number of articles that was necessary to search in order to make a selection of 50 articles that report case studies.

5.2 Reporting Case Studies

The results of this section are based on data collected in order to address sub-research question 2:

> **SRQ 2:** *What is the general impression of the quality of reporting from case studies? Is data clearly presented?*

This section presents the results of the analysis based on the existence and content of answers to the following six questions identified by Seaman [20]:

- **Who** were the subjects?
  - Age, gender, nationality
  - Education, work experience
- **What** were they doing?
  - Job descriptions
  - Studied projects
- **When** was data collected?
  - Time of day and year
  - How did it fit into their day?
- **Where** did data collection take place?
  - Physical surroundings
  - Geographical location
- **Why** did they participate?
  - Motivations, both individual and organizational
- **How** was data gathered?
  - Details of methods – recording, format, who was present, etc

Table 2 shows the number of articles that reported answers to each of the six questions.

Table 2 Distribution of Articles to Answers

Answer in Article	No of Articles	%	Comment
Who	36	72	
What (task, project)	50	100	32 articles provided descriptions of studied projects
When	19	38	
Where	24	48	
Why	14	28	
How	19	38	

The distribution of articles according to the number of answered questions is presented in table 3.

Table 3 Distribution of Articles to Number of Answered Questions

No of Questions Answered	No of Articles	%
6	3	6
5	9	18
4	9	18
3	7	14
2	9	18
1	5	10
0	8	16
Total	**50**	**100**

The remainder of this section focuses on findings to each type of question.

5.2.1 Subjects

Who were the subjects? The survey showed that subjects were reported in 70 percent of the articles. In five articles (ten percent), the subjects were students. 21 articles (42 percent) reported practitioners as subjects, whereas eight articles (16 percent) reported the authors as subjects. Another twelve percent reported a mix of subjects: five articles (ten percent) included authors and professionals; one article (two percent) reported authors, professionals and students as subjects. In four articles (eight percent), I could not categorize the subjects that were mentioned. Finally, six articles (twelve percent) did not report who the subjects were.

Nevertheless, there are great differences in the way the subjects are reported. Few articles inform about the subjects' characteristics. Here is an example of one of the more detailed descriptions of experience:

> The population under study was a graduate and senior level class offered by the Department of Computer Science at the University of Maryland, between September and December 1994. All students had some experience with C or C++ programming and relational databases. [A41]

Below is an example of a detailed description of age.

> The average age of the software developers included in this case study was 41 years, with a range of 34-52. Males represented 75% (6/8) and females 25% (2/8). Average education level was 18 years (2 years of post-undergraduate education). [A16]

Most often, however, subjects are barley mentioned: "The case study deals with the acquisition of COTS to support a team of requirements engineers in their activity" [A12].

Table 4 presents the types and extent of subjects that are reported in the articles.

Table 4 Number of Articles in Each Subject Category

Type of Subject	Article IDs	No of Articles	%
Students	A8, A23, A28, A41, A47	5	10
Professionals	A1, A2, A3, A4, A5, A9, A10, A12, A13, A16, A25, A27, A34, A35, A36, A37, A40, A42, A48, A29, A45	21	42
Authors	A17, A21, A22, A30, A33, A38, A46, A50	8	16
Authors/Professionals	A11, A18, A24, A43, A44	5	10
Authors/Professionals/Students	A26	1	2
Type unknown	A6, A14, A15, A20	4	8
Not reported	A7, A19, A31, A32, A39, A49	6	12
Total		**50**	**100**

The analysis with respect to subjects has uncovered a generally poor description level. This finding is supported by Sjøberg *et al.* [23] where a relatively low and arbitrary reporting on context variables was uncovered. This "is a hindrance for metastudies,

which are needed to identify which context factors influence which kinds of performance" [23].

Categories of subjects are interesting because of the important role realism plays in case studies. Presumably, case studies with a high level of students as subjects would not be as interesting as case studies with professional subjects. Sjøberg *et al.* [23] state that although there are good reasons for conducting experiments with students as subjects, they emphasize the fact that the low proportion of professionals used in software engineering experiments "reduces experimental realism, which in turn may inhibit the understanding of industrial software processes and, consequently, technology transfer from the research community to industry".

Thorough descriptions of subjects' characteristics will be necessary for the industry in order to relate to the context. Graduate students are not practitioners: they do not work in the same organizational and professional context; they are not subject to the same pressures [21], although it has been suggested that the lack of professional subjects in experimental work can possibly be solved by payment. As Sjøberg *et al.* [23] state:

> To increase the potential for sampling subjects from a well-defined population and to alleviate the problem of having few professionals as subjects (Section 6.1), the experimental software engineering community should apply new incentives, for example, paying companies directly for the hours spent on an experiment [3] or offer the companies tailored, internal courses where the course exercises can be used in experiments [27]. Payment would require that researchers include expenses for this kind of experiment in their applications to funding bodies, see further discussion in [39].

However, the necessary payment in case studies may be prohibitive, even though there are examples.

As an example of an article that does not report on the subjects that took part in the case study, article [A7] rather discusses the company of concern:

> To minimize this cumbersome process, ABB used a policy to avoid generating and sending specific patches to the selected customers. Instead, the revised products containing sets of patches were generated and delivered to all customers contracted for maintenance, to keep the customer installation consistent.

In 16 percent of the articles, the authors themselves use the method, e.g.: "We made use of the calls and called-by attributes derived in the previous section" [A17]. In this particular example, however, the authors do provide some information about *assumed* users: "The query paradigm employed in this section assumes a user who begins with a range of questions about the target software system, and yet has no knowledge about the actual internals of the system" [A17]. This case study was not conducted in a company. Only one of the articles, namely [A21], where the authors were subjects, also reported a company. The data used in testing the technology was taken from this company.

One article, [A22], only reports on subjects of the project that the authors use as data for testing the technology. However, it should have been reported on the authors' characteristics; after all, it is the authors that use the technology.

5.2.2 Tasks

This section summarizes the reporting on *job descriptions* (tasks) in addition to descriptions of *studied projects.* By 'studied projects' I mean the projects in the organizations where the case studies were conducted.

Tasks were far from explicitly stated. However, I chose to categorize all the articles from implicit information to get an impression of what kind of tasks that appears in case studies. This explains the high percentage of answers to this question in Table 1.

Sjøberg *et al.* [23] presents a set of task categories. However, case studies seem to involve other tasks than controlled experiments. Due to the diversity in tasks found in the articles, the categories were amended. The following categories were used:

- Software development
- Plan (including Project planning, Requirements analysis, and estimation),
- Create (including Design and Coding),
- Modify (including Maintenance of design and/or code),
- Analyse (including Inspection, Testing, and Document comprehension)
- Prototyping
- Implementation of systems
- Characterizing systems, teams
- Quality improvement

The most frequent tasks performed by the subjects in the case studies are related to the software development process or quality improvement. This is consistent with the findings of Segal *et al.* [21] where it was found that 33 percent of the research topics concerned software life-cycle/engineering and 19 percent had to do with measurement/metrics. Other frequently occurring tasks include design, prototyping, testing and requirements engineering. Again, however, the tasks were rarely reported explicitly.

Information about the studied projects was presented in 32 articles (64 percent). Here is one of the most descriptive examples:

> This paper is based on the system test stage of a project developing a retail banking application. The project included an upgrade of a customer information system being used by clients as a central customer, account and product database, and a complete reengineering of a retail banking system. The project scope included reengineering of the data model, technology change from IMS/DL1 to CICS/DB2, rewrite from JSP COBOL to COBOL-2 and a completely new physical design. [A47]

The example below illustrates a short description of the project in focus:

> The EPG installation coincided with the start of the project that was intended to provide the context for the EPG use. This was a medium sized project (approximately 800 h work). [A5]

5.2.3 Time Period of Data Collection or Studied Project

A relatively low share, 38 percent of the articles, report the period in time in which data collection took place, either implicitly or explicitly. This is consistent, however, with the low reporting on data collection (see Section 5.2.6 for reporting on data collection methods).

Of the 19 articles that reported time at all, ten articles present actual time of data collection. Often, in cases where time is reported, it is barely mentioned and with few details. The following example gives a good illustration: "The experimental work needed a cumulative effort of about fifteen calendar months spread over a year and a half" [A26].

Nine other articles report time that informs about when the project that is being observed took place rather than specifying when data collection took place. Here is an example:

> According to records, this was a thirteen month project, starting on 21 Nov 1997, and closing on 31 Dec 1998 (p. 43). Due to lack of labor resources, almost no work was done on the project from its initial definition until March 1998, effectively making the project ten months long (p. 43). [A48]

Table 5 Type of Time Reporting in Articles

Type of Time Reporting	Article IDs	No of Articles	%
Time Reported		**19**	**38**
Time of data collection	A2, A5, A6, A8, A21, A25, A26, A28, A35, A41	10	20
Time of project accomplishment	A1, A4, A10, A13, A18, A29, A43, A47, A48	9	18
Not Reported	A3, A7, A9, A11, A12, A14, A15, A16, A17, A19, A20, A22, A23, A24, A27, A30, A31, A32, A33, A34, A36, A37, A38, A39, A40, A42, A44, A45, A46, A49, A50	**31**	**62**
Total		**50**	**100**

The articles report poorly on time for data collection. In 62 percent of the articles, time is not reported at all.

5.2.4 Location of Data Collection

I found that 24 articles (48 percent) answered the question about where data collection took place. At first, I looked for data on physical surroundings (e.g., where the subjects were seated while being observed or whether they were working from their regular work

space or not) and geographical location. However, due to the low reporting I chose to register information about the institutions in which the case studies were conducted. Nevertheless, the following example illustrates one of the most descriptive answers that I registered: "The EPG was installed to support the guidance phase of a software process improvement effort at Allette Systems, a small Sydney-based company that focuses on web application development" [A5].

Some authors chose to restrain the identity of the organization where the case study was conducted: "We applied our selective testing method to actual development of a software functional testing support tool (FTST) in a certain company" [A15]. Generally, anonymity of companies' identities may be explained by commercial sensitivity of real projects. Case identities are further discussed in [27, pp. 157-158].

5.2.5 Motivation for Participation

Although 14 of the articles (28 percent) state the reason for why the subjects participated in the development projects, this was far from explicitly stated. In most of the cases, I have accepted as answers to this question the reason for why the company needed the technology. The example below is a typical illustration of this.

> Due to the large volume of documents a parsimonious yet effective inspection approach was necessary. Hence, the defect detection as well as the meeting-based collection activity was modified to fulfill these requirements as well as to address the inspection issues outlined above. This resulted in the non-traditional inspection implementation. [A3]

It appears to be mainly the organizational motivation for participation that is reported in the articles. The lack of stated individual motivation may bias the results of the data collection. After all, without information about subject recruitment it is not known on what premises the subjects joined.

In addition to the motivation of the participants, I found in one article the author's motivation for choosing the participants: "A reason for selecting these companies was that, we believe them to be representative for a larger category of software development organizations" [A27].

Another article explicitly stressed the fact that their approach to collecting empirical data was not perfect: "It is open to interviewer bias distorting the answers given by respondents. The people interviewed are not chosen at random; they will tend to self-select" [A35].

5.2.6 Methods for Gathering Data

Only 19 articles (38 percent) describe how data collection was carried out. Because of the varied reporting on data collection, I did not extract data on specific characteristics of gathering methods. I simply registered the answers I found that would be related to the subject matter.

The following example illustrates what I regard as a quite thoroughly detailed level of reporting on methods of data collection among the findings: "Data was collected from multiple sources from periodic surveys of users, the EPG server log, questionnaires and interviews with users and records of on-line discussions" [A5].

Moreover, each of the data collection methods from the example above is properly described. The text below on surveys, as one of the methods, illustrates the description level:

> A total of four surveys were performed at different stages of the study. Two survey instruments were used. Questionnaire 1 aimed to explore subjects' perception of the EPG, their views on benefits of the EPG, good/bad features of the EPG, most useful features, any lacking information, and any suggestions for improvements. This questionnaire was applied three times in the early stages of the investigation (week 2, 6 and 9). The purpose of repeat application was to see if people's views on the EPG and how they were using it changed over time. Questionnaire 2 was applied once after the application of questionnaire 1. The purpose of Q2 was to collect more detailed information about the benefits of the EPG technology as opposed to a paper-based guide. [A5]

Here is an example of a less detailed description: "I collected some statistics and metrics for both case studies" [A13]. This is supplemented later on with the following: "The article presents data on process and organizational issues based on interviews performed with the stakeholders".

Lack of reporting on data collection may be due to the high involvement of authors affiliated in industry (48 percent). Their preexisting knowledge about procedures used in the companies may be the cause of the low focus on data collection. Of the 31 articles where data collection reporting was absent, 14 articles have authors from research, 13 have authors from research & industry, and four have authors affiliated in industry.

Table 6 Methods for Data Collection

	Article IDs	**No of Articles**	**%**
Answers to how data was collected	A1, A2, A3, A5, A8, A10, A13, A14, A16, A21, A22, A23, A25, A26, A27, A35, A43, A45, A50	19	38
No answer provided	A4, A6, A7, A9, A11, A12, A15, A17, A18, A19, A20, A24, A28, A29, A30, A31, A32, A33, A34, A36, A37, A38, A39, A40, A41, A42, A44, A46, A47, A48, A49	31	62
Total		**50**	**100**

5.3 Specification of Case Study as Research Method

The results of this section are based on data collected in order to address the following sub-research question:

SRQ 3: *Do researchers state the type of research method that they have used?*

An interesting characteristic of the articles is whether the type of research method is specified or not. How many articles specify that the research method they have used is actually a case study? The remainder of this section presents findings regarding specification of research method expressed through title, abstract, keywords or elsewhere explicitly stated in the text.

> Because authors rarely indicated the research approach they employed explicitly in the abstract, keyword, or even in the introduction, we usually categorized the primary research approach used by examining relevant sections of the article. [6]

As we can see from Table 7[3] below, 15 of the articles (30 percent) present themselves as case studies via the title, and 21 (42 percent) via the abstract. Only two articles (four percent) introduce the term 'case study' through keywords and five articles (ten percent) state case study as the research method by the term 'research method' elsewhere in the text. Altogether, there are 32 articles (64 percent) that in some way explicitly present the research method used. Of the 50 articles, I found 18 articles (36 percent) where explicit specification of research method was absent. These articles expressed the use of a case study for instance via headings.

Eight articles specify the research method through both title and abstract. Additionally, one of these articles also states the research method elsewhere in the text. One additional article specifies research method elsewhere in the text, and also in the abstract. Research method specified through both title and keywords was found in one article.

One article actually specifies the research method indirectly: "Yin's [47] guidance on research design is used. This research was also influenced by Pettigrew's [48] work on longitudinal case studies" [A35].

Table 7 Distribution of Articles to Location of Research Method Specification

Location of Specification	No of Articles	%	Article IDs
"Case study" as part of the title	15	30	A1, A3, A4, A6, A7, A8, A11, A16, A19, A27, A28, A47, A48, A49, A50
"Case study" in the abstract	21	42	A2, A4, A5, A6, A10, A11, A13, A16, A17, A19, A20, A21, A23, A25, A26, A27, A28, A33, A39, A42, A50
"Case study" among keywords	2	4	A3, A43
Explicitly stating case study as research method	5	10	A5, A27, A34, A35, A45

[3] Note that the same article may occur in several of the categories for location of specification in Table 7.

5.4 What Authors Call a Case Study

The results of this section are based on data collected in order to address the following sub-research question:

> **SRQ 4:** *What is called a case study by the authors?*

5.4.1 Purpose of Case Study

During analysis, I discovered a pattern in the type of purpose case studies had in the articles. Most of the articles either claimed to report on case studies with an evaluative purpose or used case studies for demonstration of technologies. Several other studies, e.g. by Zelkowitz and Wallace [28], observed a similar pattern:

> As in the study by Walter Tichy, we considered a demonstration of technology via example as part of the analytical phase. The paper had to go beyond that demonstration to show that there were some conclusions about the effectiveness of the technology before we considered that the paper had an evaluative phase.

The following is an example on what I have categorized as reporting case studies with an evaluative purpose: "This paper presents a case study of the installation and use of an electronic process guide within a small-to-medium software development company" [A5].

Here is an example of an article I have categorized as reporting on case studies for a demonstrative purpose: "A simple case study illustrates the PMIF transfer format and how to use it. The PMIF is defined and used in an example" [A20].

Table 8 shows that 21 articles (42 percent) report on case studies with an evaluative purpose and 29 articles (58 percent) report on case studies with a demonstrative purpose.

The high number of case studies used for demonstrative purposes supports the finding of Zelkowitz and Wallace [28] that revealed a high number of papers without validation method, namely 1/3 of the papers:

> All too often the experiment is a weak example favoring the proposed technology over alternatives. Skeptical scientists would have to view these experiments as potentially biased.

Table 8 Distribution of Articles to Purpose of Case Study

Purpose of Case Study	Article IDs	No of Articles	%
Evaluative purpose	A1, A3, A4, A5, A6, A8, A13, A16, A22, A25, A26, A27, A28, A34, A35, A41, A42, A44, A45, A46, A50	21	42
Demonstrative purpose	A2, A7, A9, A10, A11, A12, A14, A15, A17, A18, A19, A20, A21, A23, A24, A29, A30, A31, A32, A33, A36, A37, A38, A39, A40, A43, A47, A48, A49	29	58
Total		**50**	**100**

5.4.2 Type of Data

Because of the importance of data collection in case studies, I additionally focused on the type of the data collected. I found that there are differences in the type of data that is reported in the articles. Zelkowitz and Wallace [29] found that what the research methods under study have in common is the collection of data on either the development process or the product itself:

> We tried to distinguish between data used as a demonstration of concept (which may involve some measurements as a "proof of concept", but not a full validation of the method) and a true attempt at validation of their results.

I similarly observed the presence of mainly two types of data being collected.

Table 9 shows the distribution of articles to type of data.[4] The majority, 54 percent (27 articles), of the articles fell into the type-of-data category *Technology Outcome*. 36 percent (18 articles) reported *Observations of Use*. The latter category includes actual observations of use in addition to the developers' perceptions about the technology they used. In ten percent of the articles (five articles), both data types were reported.

[4] Definitions on the two types of data are presented in Section 2.4, Chapter 2.

Table 9 Type of Data Reported

Type of Data		Article IDs	No of Articles	%
Observations of Use			**18**	**36**
	Developers' perceptions about technology	A1, A5, A13, A24, A25, A27, A28, A35	8	16
	Observations of use	A2, A3, A10, A11, A12, A26, A29, A42	8	16
	Developers' perceptions about technology/Observations of use	A34, A44	2	4
Technology Outcome (results accomplished by using the method)		A4, A6, A7, A8, A9, A14, A15, A17, A18, A19, A20, A21, A22, A23, A30, A31, A32, A33, A36, A37, A38, A40, A41, A45, A47, A49, A50	**27**	**54**
Observations of Use/ Technology Outcome			**3**	**6**
	Technology Outcome/ Developers' perceptions about technology	A46	1	2
	Technology Outcome/ Observations of use	A43, A48	2	4
Unknown		A16, A39	**2**	**4**
Total			**50**	**100**

Observational data regarding use is not present in approximately one third of the articles. Instead of presenting this kind of observational data, actual results provided by the technology are presented.

For instance, article [A33] about incremental integration testing of concurrent programs presents data on technology outcome: "The results of the first case study showed that incremental testing produces models that are significantly smaller than the unreduced models".

Although no observations of use of the technology are presented, an article may still present information about the use indirectly via actual results accomplished by using the technology. These results would show coherence between technology and developers, and give an impression of the combination's successfulness. Good results may indicate that the combination of the developers and technology was good, and hence suggest an efficient technology.

5.5 Affiliation of Authors

The results of this section are based on data collected in order to address the following sub-research question:

SRQ 5: *Is there a connection between what kind of data that is reported and the kind of affiliation of the author?*

This section presents an overview of how many authors that belong to research communities, industry communities or both. The purpose is to investigate whether there is a connection between report on data and the type of institution that the author represents. Moreover, I would like to see if there is a relation between explicit report on observational data and the affiliation of the author.

The affiliation of authors was coded with respect to the following categories: *research*, *industry*, and *research & industry*. The distribution is presented in Table 10. The proportion of articles where authors come from *research*, *industry* and *research & industry* is, respectively, 52 percent, twelve percent, and 36 percent. The proportion of authors from *research*, *industry* and *research & industry* is, respectively, 66 percent, 26 percent, and eight percent.

Table 10 Distribution of Articles to Type of Authors' Affiliation

Affiliation	Article IDs	No of Articles	%	No of Authors	%
Research	A1, A5, A11, A12, A14, A19, A21, A22, A23, A25, A26, A27, A30, A31, A32, A34, A35, A37, A38, A39, A40, A44, A45, A46, A49, A50	26	52	86	66.2
Industry	A2, A9, A13, A20, A47, A48	6	12	34	26.2
Research & Industry	A3, A4, A6, A7, A8, A10, A15, A16, A17, A18, A24, A28, A29, A33, A36, A41, A42, A43	18	36	10	7.7
Total		**50**	**100**	**130**	**100**

The number of articles from industry is consistent with that of Segal *et al.* [21]. Furthermore, like Segal *et al*., I found that authors from research communities dominate. However, their findings differ to some extent from my findings regarding articles from the categories *research* and *research & industry*. Segal *et al.* found respectively, 73 percent and 16 percent compared to 52 percent and 36 percent in my study. This may be due to the fact that Segal *et al.* considered the spectrum of research methods used in empirical software engineering, whereas I investigated the case study method exclusively. The difference may therefore be caused by the nature of the case study method with respect to an industrial setting and the industry's interests in this kind of method evaluation; hence, a higher proportion of authors from the industry.

Table 11 presents the distribution of articles to type of data that is reported and affiliation of authors.

Table 11 Distribution of Articles to Type of Data Reported and Affiliation of Authors

TYPE OF DATA REPORTED	AFFILIATION OF AUTHORS			No of Articles
	Research	Industry	Research & Industry	
Observations of Use	8 (16%)	3 (6%)	7 (14%)	18 (36%)
Technology Outcome	14 (28%)	3 (6%)	10 (20%)	27 (54%)
Observations of Use/ Technology Outcome	3 (6%)	0	0	3 (6%)
Unknown	1 (2%)	0	1 (2%)	2 (4%)
Total	**26 (52%)**	**6 (12%)**	**18 (36%)**	**50**

36 percent of the articles (18 articles) present data consisting of observations of use including developers' opinions about the technology. Of these, eight articles are written by researchers, three by authors from industry and seven by authors with relations to both communities.

In 54 percent of the articles (27 articles), data is reported which consists of technology outcome. Of these, 14 articles are written by researchers, three by authors from industry and ten by authors with relations to both communities. Six percent (three articles) present data collection on both types, all of them are written by researchers.

5.6 Confusions Regarding Research Methods

The results of this section are based on data collected in order to address the following sub-research question:

SRQ 6: *Are case studies confused with other research methods?*

The table below shows the various terms on research methods that appear in the articles.

Table 12 Use of Research Method Terminology

Research Method	Article IDs	No of Articles	%
Case study	A1-A50	50	100
Action study	A43	1	2
Experience report	A19	1	2
Field study	A42	1	2
Controlled experiment	A23, A34, A41, A45	4	8
Experiment	A13, A15, A17, A19, A21, A26, A30, A31, A46	9	18

During the analysis of the articles I found, like Sjøberg *et al.* that use of the term 'experiment' is inconsistently used in the software engineering community [23]. Nine of the articles referred to their studies as experiments even though they claimed to be reporting on case studies. This stresses the multiple meanings of this term.

One of the articles uses all three of the following terms: 'experience report', 'case study' and 'experiment'. The article starts by calling itself an experience report: "This paper is an experience report that illustrates the applicability of a viewpoint-based design method for the Web-based education (WBE) domain" [A19]. Nevertheless: "The method applicability is illustrated by a large real case study in the WBE domain. … The analysis process presented here is a large cased study that helped us to validate our viewpoint-based design method". Finally, the article uses the term experiment: "Section 4 describes the implementation and use of the derived framework to validate the experiment".

The term 'controlled experiment' was also found in some of the articles, e.g.: "The data have been collected in controlled experiments" [A23]. However, the same article states the following: "In a case study, axioms from the measurement theory will be tested, both formally and empirically".

5.7 Multiple Case Studies

This final section presents the results addressed to answer the following sub-research question:

> **SRQ 7:** *What is the extent of the use of multiple case studies in empirical software engineering?*

Table 13 shows that eleven articles (22 percent) report on multiple case studies.

Table 13 Articles Reporting on Multiple Case Studies

	Article IDs	No of Articles	%
Multiple Case Study	A13, A14, A21, A29, A32, A33, A39, A40, A42, A43, A50	11	22

5.8 Summary

This section summarizes the major findings of this review.

- Extent of case studies was close to twelve percent.
- Answers to the six questions from Seaman [20]:
  - **Who:** Although 70 percent stated who the subjects were, the description level was poor.
  - **What:** Information of tasks in the projects was reported in 64 percent of the articles.
  - **When:** 38 percent reported the time of data collection. However, the articles that reported this kind of information provided few details.
  - **Where:** 48 percent report where data was collected.
  - **Why:** As few as 28 percent stated why the subjects participated in the case studies, several even not explicitly reported either.
  - **How:** 38 percent report method for data collection.
- Specification of case study as the research method was found in 86 percent of the articles:
  - Title: 15 percent
  - Abstract: 42 percent
  - Keywords: 4 percent
  - Explicit elsewhere: 10 percent

  The remaining 14 percent express the use of the case study method for instance via headings.
- Purpose of case study reported in article:
  - Evaluative purpose: 42 percent
  - Demonstrative purpose: 58 percent
- Type of data reported:
  - Observational data: 36 percent
  - Technology data: 54 percent
  - Observational data/Technology data: 6 percent
- The lack of data collection may be reminiscent of the assertion method. Additionally, I found some mixing of terms on research methods, mainly on *experiment* and *controlled experiment*.
- Affiliation of authors:
  - The following distribution of articles was found regarding affiliation of authors:
    - Research: 52 percent
    - Industry: 12 percent
    - Research & Industry: 36 percent
  - There are more articles reporting on technology data than observational data. Interestingly, the majority of the articles (28 percent) with authors affiliated in research communities appear to report technology data.
- 22 percent report multiple case studies

6 Discussion

Section 6.1 addresses the research question from Section 1.2 based on the findings presented in Chapter 5 and issues of consideration that came up during analysis of the articles. Furthermore, a discussion of existing guidelines and specifications of case studies in empirical software engineering will be provided in Section 6.2.

6.1 State of the Art

The research question of this thesis requests an overview of the existing use of case studies in empirical software engineering:

> **RQ:** *What is the state of the art regarding the use of case studies in empirical software engineering?*

As we recall from Section 1.2, the research question is further composed by sub-questions, which will be addressed throughout the following sections. An additional issue regarding realism will be discussed in Section 6.1.8. Finally, the discussion is summarized in Section 6.1.9.

6.1.1 Proportion of Case Studies

The following sub-research question is addressed in this section:

> **SRQ 1:** *What is the extent of the use of case studies in empirical software engineering?*

Close to twelve percent (50 articles) of the 427 articles assessed were identified to report on case studies according to the selection criteria presented in Section 4.2.2. [5]

Furthermore, Zelkowitz and Wallace [28] found that close to ten percent of the papers relied on the case study method. Segal *et al.* [21] found that 13 percent of the papers examined, used the case study as the research method. This is consistent with findings from my review.

There are, however, surveys that found a lower share of case studies than what was found in this review. Glass *et al.* [6] found that only 2.2 percent of the papers were case studies. Their study revealed Conceptual analysis, Conceptual analysis/mathematical, and Concept implementation as being the most dominant research methods: “SE researchers tend to analyze and implement new concepts, and they do very little of anything else” [6]. The journals under consideration were IEEE Software, IST, JSS, SP&E, TOSEM and

[5] The correct way of providing the extent of case studies is to collect a specified number of articles which then are searched for occurrences of case studies. The answer to SRQ 1 is based on the number of articles that was necessary to search in order to make a selection of 50 articles that report case studies.

TSE. All of these journals are included in the selection of journals from which the articles of my survey were taken. Nevertheless, an explanation of the aberrancy may be the inclusion criteria. There is no information about what Glass *et al.* consider as case studies, whereas my inclusion criterion is rather broad (see Section 4.2.2).

The study of Ramesh *et al.* [19], concerning research in the related discipline of computer science, found that only 0.16 percent of the papers assessed were based on the case study method.

6.1.2 Reporting Case Studies

This section describes my general impression of extent and clarity of reporting in the articles:

> **SRQ 2:** *What is the general impression of the quality of reporting from case studies? Is data clearly presented?*

As Yin [27, p. 141] claims in the quotation below claims, reporting case study results is not easy. This is exactly the impression I had after extracting data from the articles.

> Whether serving as a finished case study or as part of a multimethod study, reporting case study results also is one of the most challenging aspects of doing case studies.

Extent of Reporting

The response level to the questions that I used for data extraction appeared to be rather low. The average response rate was only 48 percent.

The question about *who* the subjects were was the most frequently answered question among the articles. As many as 36 articles (72 percent) provided this information in some manner. When it comes to the proportion of articles that report on *what* the subjects did, answers to descriptions of studied projects were found in 32 articles (64 percent). Regarding what tasks the subjects conducted, I categorized all the articles from implicit information to get an impression of what kinds of tasks that appear in case studies. As a result, I found answers in all the articles. These are not included in Table 3. Answers to *where* the data collection took place were found in 24 articles (48 percent). However, I ended up registering where the case study was conducted due to the few answers. Questions about *when* and *how* data collection took place were answered in 19 articles (38 percent). The least answered question was about *why* the subjects participated, which had a response rate of 28 percent (14 articles).

Note that due to the low level of reporting, rather vague answers were included in the data extraction.

Distribution of answers
Only three articles provided answers to all the questions. Further, there were nine articles that answered five questions; nine articles answered four questions; seven articles that answered three questions; nine articles that answered two questions; and five articles that answered one question only. Surprisingly, however, there were eight articles where I could find no answer to neither of the questions. [6]

The findings show that there was a rather low response rate to the questions specified. This is quite surprising due to the relevance these questions have to the reader. Such information is important for the reader in order to relate to the context in which the case study was conducted and in this way find the results useful.

Clarity of the data presented in the articles
Generally speaking, it was difficult to find answers to the six questions. There was little use of descriptive headings, which would help the reader to orientate in the article. The information was not explicitly stated, but had to be found after careful inspections and evaluations of contents. The trend in the articles implies that the readers have to base their conclusion on data extracted implicitly. Hence, my general impression is that information is not clearly reported.

Zelkowitz and Wallace [29] observed that authors often fail to state how they validate their hypotheses. They had to inspect each paper carefully to determine what the authors were intending to show in the various sections called "validation" or "experimental results". Often such a section couldn't be found, so they had to determine if the presented data could be called a validation.

Indication of purpose
However, the extent of reporting may be considered as another indication of different purposes for using case studies. This review detected that for articles reporting on case studies with an evaluative purpose (21 articles), 14 articles (67 percent) provided answers to four or more questions. In comparison, of the 29 articles that reported on case studies with a demonstrative purpose, only seven articles (24 percent) provided four or more answers. Hence, a demonstrative purpose tends to have less reporting than those with an evaluative purpose.

However, low quality of reporting is not only the case for the case study method. Sjøberg *et al.* [23] conclude with the following:

> A major finding of this survey is that the reporting is often vague and unsystematic and there is often a lack of consistent terminology. The community needs guidelines that provide significant support on how to deal with the methodological and practical complexity of conducting and reporting high-quality, preferable realistic, software engineering experiments.

[6] Answers to task are not taken into consideration in the distribution of articles.

6.1.3 Specification of Case Study as Research Method

I would now like to address sub-research question 3 by paying attention to the way researchers specify what kind of research method they have used.

> **SRQ 3:** *Do researchers state the type of research method that they have used?*

When it comes to specifying case study as the research method, 32 articles (64 percent) provide this information either through abstract (21 articles; 42 percent), title (15 articles; 30 percent), explicitly elsewhere (five articles; ten percent), or keywords (two articles; four percent). [7] Of the 32 articles that specified the research method, 15 articles (47 percent) had authors affiliated in research, twelve articles (38 percent) had authors affiliated in research & industry, and five articles had authors affiliated in industry (16 percent). However, articles with authors affiliated in industry were more likely to specify research method than not specifying. Additionally, there were a greater number of articles with authors from research & industry that specified the research method than not specifying. This may indicate the importance of case studies to the industry.

Nevertheless, 18 articles out of 50 articles (36 percent) is quite a high share that does not explicitly specify what research method that has been used. Interestingly, there is a tendency of lower specification on research method in articles with a demonstrative purpose (15 of 29 articles specify; 52 percent) than articles with an evaluative purpose (seven of 21 articles specify; 33 percent).

6.1.4 What Authors Call a Case Study

In this section, the results from the survey are used in order to present an overview of what is called a case study in the literature of software engineering.

> **SRQ 4:** *What is called a case study by the authors?*

In order to address this sub-research question, I decided to collect data on what purpose the case study had in the article and what type of data the article reported. The rest of this section presents relationships between the purpose of the case study, the type of data reported in addition to subject type and reporting on methods for data collection.

Purpose of Case Study

I have found mainly two uses of the term ‘case study’:

- *Evaluative purpose:* Articles that report on observations of the use of a technology, including the subjects’ perceptions about the technology. Subjects are most often students or professionals.
- *Demonstrative purpose:* Articles that report on the technology outcome. The authors themselves appear most frequently to be the subjects. Exemplifies use of the technology.

[7] Be aware that an article may specify the research method in several of the categories.

The Case Study Used for an Evaluative Purpose

There were 21 articles that reported from case studies with an evaluative purpose.

The following figures were found regarding *subject type*: twelve articles (57 percent) had professionals; three articles (14 percent) had students; three articles (14 percent) had authors; one article (five percent) had authors/professionals; one article (five percent) had authors/professionals/students; one article (five percent) reported subject type, however the type was found to be unknown. All articles reported on subject type. This shows that professionals are the most common type of subjects (close to 70 percent) in articles with case studies of the evaluative purpose.

The *method for data collection* was specified in 13 articles (62 percent).

The Case Study Used for a Demonstrative Purpose

There were 29 articles that reported from case studies with a demonstrative purpose.

The following figures were found regarding *subject type*: nine articles (31 percent) used professionals; five articles (17 percent) used authors; seven articles (14 percent) used authors/professionals; three articles (ten percent) reported subject type, however the type was found to be unknown; two articles (seven percent) used students; and finally, six articles (21 percent) did not report subject type. Moreover, there was additionally a lack of reporting regarding where the case study was conducted (which organization) in all of these articles that did not report subject type. This may imply that the technology was tested by the authors totally or partly in more than half of the articles.

Descriptions of methods for data collection were absent in 23 articles (79 percent). This supports what is said by Zelkowitz and Wallace [29], about the absence of data collection in experiments, which frequently appears to be a fact in papers that present some new technology where “experiments” are performed to show how effective the technology is. The creator of the technology both implements and shows that it works.

There was a trend in *what kind of subjects* the two types of uses involve. As we can see, articles that report on case studies with an evaluative purpose have subject types like students and professionals, whereas articles with a demonstrative purpose are likely to involve the authors as subjects.

Finally, the data from the survey showed that articles of a demonstrative nature were less likely to report *the methods for data collection* compared to articles of an evaluative nature.

Type of Data

Researchers seek to validate hypotheses. Collection of data is necessary in order to perform such validation. What kind of data do the articles report upon? What is measured: Observations of developers using the technology or only outcome of technology?

The survey shows that mainly two types of data are reported:

- *Observations of use:* Data about the use of a technology.
- *Technology outcome:* Data about the technology.

Regarding the relation between subject type and data type, 13 of the 18 articles (72 percent) reporting on *observation of use*, used professionals as subjects, one article (six percent) used students. The remaining 22 percent that report observations of use, either involved the author somehow in subject type, the subject type was unknown or not reported at all. This shows that when the focus of reporting is directed towards the use of the technology, professionals are most frequently used as subjects.

A majority of the subjects in articles that report on *technology outcome* fall in subject categories involving the authors, unknown or not reported. Together, these categories make 63 percent (17 of 27 articles). An explanation to this may be that the technology is a prototype (or complex) so that training is needed in order to be able to use the technology. Hence, the authors test the technology themselves on data from the industry instead of using practitioners to use the technology. Reporting on observation of use may in such cases be of secondary interest to the researcher.

Similarly, Zelkowitz and Wallace [29] report the following:

> There are many examples of developers being both experimenters and subjects of study. Sometimes this happens during a preliminary test before a more formal validation of the technology's effectiveness.

A case study should present observations of use or evaluation of a technology. Several of the articles describe collection of data about a technology, but not about the use of the technology. Empirical software engineering investigates developers and technology in order to find what works better together; i.e. developers are studied as they *use* technology. A case study should include observations of use or evaluation of a technology. Several of the articles describe collection of data according to a technology, but not according to the use of the technology (e.g. [A33]). Observational data regarding use is absent in 29 articles (58 percent). Instead of presenting this kind of observational data, actual results provided by the technology are presented. In other words: Data about use of the technology is missing. Data on the outcome of the technology is reported.

Nevertheless, it is not easy to make a clear distinction between the two types of data. This is due to the fact that data collected from a technology's output may say something about the use of the technology. An article may still present information about the use of the technology even if there are no observational data presented. In these cases, it would be

indirect via actual results accomplished by using the technology; these results would show a relationship between technology and developers, and give an impression of the combination's successfulness. Good results may indicate that the developers and the technology was a good combination, and hence suggest an efficient technology. On the other hand, case studies are not benchmarks.

Trends

Interestingly, I found a relationship between *purpose of case study* and the *type of data* being reported in the articles. Typically, the majority of the 29 articles that included case studies with a demonstrative purpose report technology outcome (20 articles: 69 percent). Of the 21 articles that included case studies with an evaluative purpose, twelve articles (57 percent) present observational data.

Additionally, I found a relationship between presence of *data collection methods*, *type of data*, and *purpose of case study*. Of the 50 articles, 31 articles (62 percent) provided no information about how data was collected. Of these 31 articles, 22 articles (71 percent) reported on outcome of technology or a mix of outcome and observations. Furthermore, I categorized 18 of these 22 articles (82 percent) as having case studies with a demonstrative purpose. As we can see, there is a tendency of lack of reporting on methods for data collection in demonstrative case studies.

In comparison, only three of the 21 articles (14 percent) reporting on case studies with an evaluative purpose and technology outcome did not provide information of method of data collection.

6.1.5 Affiliation of Authors

I expected a relationship between the type of data reported and the affiliation of the authors:

> **SRQ 5:** *Is there a connection between what kind of data that is reported and the kind of affiliation of the author?*

My presumption was that articles where the outcome of technology was reported would be authored by researchers affiliated in industry. These authors hold information about development procedures in their organizations. Reporting of interest may thus fall on the actual technological outcome. If this were a reality, it would conflict the importance of reporting context to the industry.

However, the findings showed that there was no major trend when it comes to the distribution of articles according to the affiliation of authors and the type of data being reported.

I also expected a relationship between the affiliation of the authors and the reporting of data collection. I assumed that researchers were more likely to report the method used for

collecting the data. Of the 31 articles (62 percent) that did not report on method for data collection, 14 articles have authors from research, 13 have authors from research & industry, and four have authors affiliated in industry. In other words; more than half of the articles are authored by people affiliated in industry. Then again, the other half is authored by authors affiliated in research communities. Hence, there was no specific trend in affiliation of authors and reporting on method for data collection.

6.1.6 Confusions Regarding Research Methods

This section addresses the following sub-research question:

SRQ 6: *Are case studies confused with other research methods?*

The full answer to this question requires operational definitions of the case study method and other research methods. At present, there are no such definitions. Pending this, the observations of this review suggest that the lack of reporting observations of use may be reminiscent of the assertion method. There is a bias in case studies towards verification (Flyvbjerg 2004, referenced by Mohagheghi [17, p. 42]). This review detected frequent occurrences of articles where the case study seems to be used for exemplifying successful use of a technology. These are typically articles with a demonstrative purpose where data regarding technology outcome is reported.

There is a lack of sufficient data collection as evidence, and only 38 percent of the articles reported how data was collected. At least, this may indicate that the case study method is not used correctly.

I found some mixing of terms on research methods, mainly on the term 'experiment'. There were nine articles that refer to their studies as both 'case study' and 'experiment'. The meaning of the term 'experiment' may in these examples be synonymous with the term 'empirical study', as also observed by Sjøberg *et al.* [23].

Characteristics making the Difference

In order to specify what case studies are not, it is helpful to study the distinction between the case study method and controlled experiments. I will now illustrate the distinctions in terms of comparison and control (Section 2.2.6), by examples from the reviewed studies.

Comparison

Case studies can be comparative (see Section 2.2.6). Multiple case studies, for instance, investigate technologies in relatively similar or varied contexts and compare these. Article [A50] is an example of a replication of a study conducted in a different environment where the goal is to "empirically assess the object oriented design measures discussed in a literature review, and compare the results to those obtained in an analogous study using systems developed by students". As stated in the article: "In order to draw

more general conclusions and (dis)confirm the results obtained there, we now replicated the study using data collected on an industrial system developed by professionals".

Manipulation/control
Yin says that the case study method is favored when the relevant behavior cannot be manipulated; i.e. the degree of control the researchers have when conducting tasks in the project [27, p. 7].

However, case studies must to some extent involve manipulation (see Section 2.2.6). For example, the researcher should be allowed to ask for a particular technology to be used in an organization. The researcher should even be allowed to assist subjects during use. Article [A2], is shown as an example:

> Because permitting each evaluator to have hands-on access to the product was impractical, we prepared a facility in which the evaluators could observe an analyst, who acted as their representative, executing each scenario while a vendor advised the analyst of the best way to accomplish each step. The evaluators sat at a table facing the vendor and analyst so they could observe what was happening without being visible to other evaluators.

A researcher may want to test, by way of the case study method, a technology in a real setting. Article [A21] is an example of a case study where the technology is tested on real data: "For experiments, we used a set of real process data and quality ("blister") data that were collected every hour for 10 months in three different glass manufacturing lines where the glass panels for CRT TV are manufactured".

To sum up: It is reasonable that technology testing is initialized by researchers, although tested in a real setting.

6.1.7 Multiple Case Studies
This section discusses issues concerning the final sub-research question:

> **SRQ 7:** *What is the extent of the use of multiple case studies in empirical software engineering?*

The Value of Case Study Results
The following questions are frequently asked in literature [27, p.10]: Are case studies an approved way of making generalizations towards theory? Is it possible to generalize from a single case study?

Fenton and Neil [5], have the following statements about generalization:

> Collecting data from case studies and subjecting it to isolated analysis is not enough because statistics on its own does not provide scientific explanations. We need compelling and sophisticated theories that have the power to explain the empirical observations.

Dybå [4, p. 66] says that a weakness when it comes to case studies is the difficulty in generalizing, given problems of acquiring similar data from a statistically meaningful number of cases.

Kitchenham and Jones [12] express that case studies only provides limited confidence in the reliability of the evaluation. This can be handled with Yin's proposals on how to improve validity by using multiple sources of evidence, pattern matching and by using theory in research design in single case studies to improve external validity [27, p. 34].

As we have seen, a critical remark against case studies is that one cannot generalize on the basis of an individual case and therefore not contribute to scientific development. However, formal generalization is overvalued as a source of scientific development, whereas the force of example is underestimated. In analytical generalization, the researcher strives to generalize a particular set of results to some broader theory or to a broader application of a theory [17, p. 42].

Yin stresses the fact that case studies, like experiments, are generalizable to theoretical propositions and not to populations or universes. In this sense, the case study, like the experiment, does not represent a "sample", and in doing a case study, your goal will be to expand and generalize theories (analytic generalization) and not to enumerate frequencies (statistical generalization) [27, p. 10].

Kitchenham *et al.* [10] say the following:

> The results of case studies are context-dependent, but we can be more confident that a method is generally beneficial if encouraging results are reported by a number of different organizations under a number of different conditions. We can also better understand the limits of methods and tools if we get conflicting reports from different case studies.

Segal *et al.* [21]:

> An argument often made against field studies is that they cannot be replicated – but neither can a software engineering activity in the real world (one cannot dip one's toes into the same river twice!). Validation of such studies can be based not on replication of the study but on replication of the interpretations: the question to ask is, would other researchers from the same scientific cultural tradition as the original researcher(s) and given the same data, come to the same conclusions?

If you want to find out if using a technology will improve your project's software, but do not need to know if using this technology will improve everyone's software, then a formal experiment may be overkill – you can rely on a case study [10]. The results may not be valid outside the project or organization, but the results are still useful from the software organization's point of view [3, p. 19].

By replicating the case study research, generalization of case study results can be enhanced.

I found that eleven articles (22 percent) reported on multiple case studies.

6.1.8 Realism

Due to the rather high level of use of subjects other than practitioners in the case studies in addition to the context-characteristic of case studies, I find it important to discuss the issue of realism.

Case studies are supposed to investigate the use of a technology in real life where the context is an important condition when looking at the results accomplished by using the technology. This means that "you would use the case study method because you deliberately wanted to cover contextual conditions – believing that they might be highly pertinent to your phenomenon of study" [27, p. 13]. A case study is defined to be an evaluation exercise where a method or tool is tried out on a real project [12].

As we have seen, literature characterizes case studies by being carried out in realistic circumstances. However, what is meant by realism? During analysis of the articles, I found that few articles would be regarded as case studies if they were to fulfill the criteria of reporting on *a real project in industry*. Particularly, there were examples of technology studies carried out by way of pilot projects. Carrying out such projects may be abnormal to the organization and not a regular project. Would it then be correct to call it a *real* project, despite the fact that the case study was carried out in an organization? Hence, real might mean *an industrial project*. Another situation would be when students perform tasks for organizations, and perhaps not even related to a particular project. Additionally, there are cases where the authors test a technology on actual data from an organization. For instance, in [A3, A50] the researchers experiment on data collected from a real world project. Yet another issue of concern: If a case study was carried out at a university, the context would not be realistic. An exception may be, if a technology is tested in a laboratory/university, we consider this as a case study if the environment itself will use the technology in their work. The laboratory/university can in these cases be thought of as the organization. Thus, in order to include these cases, realism ought to mean *an industrial setting*.

In other words, the industrial context was quite diverse. This review found articles in where the authors tested a technology on data from an organization, articles that had a combination of practitioners and researchers who tested the technology, articles where

students tested the technology at universities, and students who tested the technology in companies.

6.1.9 Summary

The objective of empirical software engineering is to find what works best among software developers and technology. There are often hundreds of alternative technologies: How should the industry (and others who build software) judge what technologies (processes, methods, techniques, guidelines, and tools) are useful for different kinds of developer, performing different kinds of task, on different kinds of system, in different kinds of organization? Thus, research in empirical software engineering should aim to acquire general knowledge about which *technology* is useful for *whom* to conduct which (software engineering) *tasks* in which *environments* [23].

The findings detected 21 articles with such an evaluative purpose. Typical characteristics for articles with an evaluative nature are rather high response rates for the six questions in the survey, the reporting of observations of use, and most likely the use of professionals as subjects.

However, there was also another type of purpose of the case studies, namely, a demonstrative purpose. There were 29 articles with this type of purpose. Typical characteristics for articles with a demonstrative nature are relatively low response rates for the six questions in the survey, the reporting of technology outcome, and most likely the use of authors of the articles as subjects.

Segal *et al.* [21] urge researchers to scrutinize the external validity of their laboratory experiments and propose an issue to be considered: “do the results of their research really have the potential to inform the richly contextualized practice of software engineering?”. The fact that the industry seeks documentation on technology efficiency makes it crucial with careful evaluations and not just demonstrations. Such evaluation ought to include reporting of main issues regarding the research like subjects, tasks, environment, and data collection that will provide necessary details to the readers in order to understand the context in which the results (technology was used) were achieved.

Zelkowitz and Wallace [29] say: “Without a confirming experiment, why should industry select a new method or tool?”. In other words, it is not sufficient with pure demonstrations of use of the technology: “In a scientific discipline, we need to do more than simply say, ‘I tried it, and I like it’”. Thus, careful reporting should be provided when doing research in order to provide evidence of results.

6.2 Efforts to Enhance the Use of the Case Study Method

This section presents efforts to enhance the use of the case study research method. Section 6.2.1 describes existing guidelines. Section 6.2.2 discusses aspects of case studies aiming for a specification of the case study in empirical software engineering. Finally, examples of "good" case studies are presented in Section 6.2.3.

6.2.1 Existing Case Study Guidelines

The conducting of case studies is not unproblematic [3, 10]. Assistance by way of guidelines is useful for assuring quality of the results. Nevertheless, there are few agreed procedures for undertaking case studies [11]. The remainder of this section presents existing guidelines for how to carry out case studies.

The literature provided by Yin [27] is one of the more comprehensive case study literatures that exist. However, in the field of software engineering there are few proposals on thorough guidelines. Kitchenham *et al.* presents their contribution on guidance in [10]. They say that the case study method is an important research method because "case studies help industry evaluate the benefits of methods and tools and provide a cost-effective way to ensure that process changes provide the desired results". Nevertheless, what the research method does not have is "a well-understood theoretical basis". This is the motive of Kitchenham *et al.* for providing guidelines to use when designing and analyzing case studies with the aim of producing meaningful results and draw valid conclusions. The guidelines are directed towards evaluation of methods and tools, and consist of the following steps:

- Define the hypothesis
- Select the pilot projects
- Identify the method of comparison
- Minimize the effect of confounding factors
- Plan the case study
- Monitor the case study against the plan

In addition to these seven steps, they provide a checklist for how to plan a case study.

Arisholm *et al.* [2] present guidelines for conducting case studies based on six industrial case studies. However, these guidelines are mostly directed towards how to act towards the organization in which the case study is carried out. They emphasize the importance of ensuring the usefulness of the results for the organization. One of the guidelines deals with the skepticism from an industrial perspective in the matter of critical information. They suggest that a confidentiality agreement with the organization should be signed. The organization should also read and accept the publications before they are submitted. In this way, organizations are given control of presentation of information. Arisholm *et al.* also addresses pilot-study; considerations regarding who should test the technology and how this should be done; and collection of real time data as means of how to get high quality data.

According to Yin [27, p. 28], a research design should not only indicate what data are to be collected, but also describe what is to be done after the data have been collected. His advice is to conduct at least two case studies in order to have strong evidence [27, p. 53].

When it comes to *reporting*, Yin provides the most thorough guidelines. Identifying the audience for the report; developing the compositional structure; and following certain procedures are the main steps. The fact that each audience has different needs implies that several versions of a case study report may be needed. Specific for software engineering would e.g. be the nature of the evidence collected, as Segal *et al.* [21] say: "For example, quantitative evidence might be necessary to convince a manager to introduce some change in working practices; a rich case study might persuade developers to accept such a change".

Yin additionally stresses the importance of a case study database to be used for reading and storing the evidentiary base of the case study [27, pp. 101-104].

Further, Yin presents procedures in doing a case study report. The main steps include:

- when and how to start composing;
- case identities; and
- the review of the draft of case studies.

Yin [27, p. 76] claims that a guide for the case study report generally is missing in most case study plans. Moreover, the guidelines of Kitchenham *et al.* [10], only state that the results must be reported without providing a guide on how to do this. Yin [27, pp. 67-77] proposes a protocol through an example for what a report should include:

- Overview of the case study project including background project information
- Field procedures
- Case study questions reflecting the full set of concerns from the initial design to reporting in order to keep the researcher on track. An outline of the case study report:
    - *The posing of the research questions and hypothesis*
    - *A description of the research design*
    - *Apparatus*
    - *Data collection procedures*
    - *Discussion, conclusion*
    - *The presentation of the data collected*
    - *Analysis of the data*

Yin also provides descriptions of how to collect the evidence and how this should be analyzed.

In order to produce quality results which are easy for reviewers and industry to orientate, there is a need for standardizations for the use of case studies. Use of guidelines, like the ones described previous in this section (although specified for empirical software engineering) would help researchers ensure the quality of the results.

A standardization of the case study method is advantageous to several actors and concerns. First of all, the quality and validity of case study results can more easily be assessed by reviewers. Secondly, it is easier to read documents that follow a known structure and where contents are specified by metadata. This would help the field of software engineering to take advantage of previous research in a more efficient way among other things due to less effort needed in searching for and interpreting contents. Thirdly, it would help the researchers to produce sufficient evidence so that the results are put in a detailed context. Thereby, others can relate to the results.

6.2.2 Recommendations

At present there is no accurate definition of what a case study in the context of empirical software engineering actually is. Available literature on case studies provides general definitions. These are mostly directed towards social science. However, in the field of software engineering there are challenges that require additional considerations.

Because of this, the rest of this section discusses different aspects of relevance for case studies conducted in empirical software engineering. These aspects, I believe, should be taken into consideration when conducting and reporting case studies, and before a final specified definition of the case study method can be settled. Moreover, I present proposals for case study criteria, what to report, when to use the case study method, and finally a suggestion of elements to include in a future definition.

Case Study Criteria

First of all, researchers should be more specific about what research method they have used. This should be explicitly specified in the article.

I found that there is a tendency of case studies where data collection from observations of use is missing. Instead, data about the product (technology outcome) is collected in order to evaluate the technology. The question is whether we should only regard evaluation consisting of *observations of use* as proper case studies, or additionally include research that report the *technology outcome* without reporting on the actual interaction between the software engineers and the technology.

Because results are a good indicator of the successfulness of the interaction between technology and software engineers, case studies should also include research that only presents results. In this case, however, a description of the context would allow the reader to determine whether expected effects apply in his/her own organizational and cultural circumstances. Thus, it is preferable with more than pure outcome of the technology.

In order to separate the two uses of case studies, a richer vocabulary is needed in order to be more specific about how the technology has been evaluated; i.e. address more precisely the use of the case study. Different disciplines have different approaches and often use the term case study to mean different things [11].

If a technology is tested in a laboratory/university, we consider this as a case study if the environment itself will use the technology in their work. The laboratory/university can in these cases be thought of as the company. Otherwise, the case study should be conducted in an industrial organization.

Another important aspect concerns the subjects in the case studies. The industry may be skeptical to the use technology that has not been tested and shown good results. It is too expensive if the technology fails. In order for a new technology to be accepted, it must have been tested somewhere else first. A solution to this may be that preliminary testing of the technology is performed by the researchers; however, on actual data from the industry. In these cases, the researchers themselves are the subjects of the case studies.

Another situation is when it is the authors of a case study who test the technology, and further help the professionals to use the technology themselves. This may be due to the unfamiliarity of the technology among the professionals and hence the need for assistance in that the authors actually partly perform the tasks.

The issue of the use of students as subjects depends on the topic of the research. Segal *et al.* [21] say that "it is plausible that there are circumstances where laboratory experiments with students might yield results which can inform practice", where an example could be experiments concerned with individual cognition. I find this to be reasonable for case studies as well.

However, what is important is that the industry trusts the evaluation form. In a case study with an evaluative purpose, it should not be the author who studies him/herself. This is because the industry is interested in evaluations conducted in realistic environments; hence, the subjects should be practitioners from the industry testing the technology in their normal environment. The author would therefore not normally fulfill the criterion (see also Houdek in Juristo and Moreno [8]). A special situation though, would be if the author worked in the industry and tested a technology in his/her normal environments, and reported on experiences he/she had by using the technology.

The following are criteria I recommend for case studies in empirical software engineering:

- **Specification of research method:** The author must *specify* that the research method used is the case study method.
- **Focus of research:** The focus in the case study should be *use/evaluation* of a *software technology*.
- **Realism:** The case study should test a technology in an *industrial setting* (see Section 6.1.8).
- **Subjects:** The technology must be used by *others than the researchers themselves* (because of no manipulation), preferably by professionals. The author cannot study him-/herself. Nevertheless, there are exceptions as discussed earlier in this section.

Case Study Reporting

During the analysis of the articles, I detected a diverse level of reporting regarding structure, clarity, and contents.

Often, the reporting was unstructured. This is why I would like to emphasize the value of using designated sections with descriptive headings in the reporting as means to provide a structured format. Descriptive headings make it easier for the reader to orientate in the article and to make an overview of contents.

Regarding contents, in order to be valuable for further research, the reporting on case studies must provide sufficient descriptions. It is important to report the properties of the organization where the case study is conducted, due to the influence the context may have on the results. Karahasanovic [9, p. 49] says the following: "The weakness of the case study method is that it is very context biased (each development is unique)". However, I would on the contrary claim this to be one of the strengths, and indeed the issue of context bias emphasizes the importance of sufficient context descriptions.

Any matters that may influence the results should be reported. For example, the presence of the researchers conducting the case study may affect the performance and thereby the results [9, p. 49]. This is necessary information to the readers when considering the validity of the results.

Finally, as Yin [27, p. 141] recommends, researchers should start early composing portions of the case study.

These are my recommendations regarding reporting on case studies in order to provide a rich description of the context:

- **Clarity of reporting:** Reporting on the case study should be clear and structured, preferably with careful use of descriptive headings.

- **What to report:**[8]
  - **Subjects:** The number of subjects and characteristics of the subjects that performed the tasks: age, gender, nationality, education, general software engineering experience and experience specific to the tasks. [9]
  - **Tasks:** Type of tasks, duration of the tasks, and application areas of the tasks that was carried out.
  - **Motivation:** How the subjects were recruited, including motivation for participation**.**
  - **Data collection:** Systematic data collection must be presented, including what it is collected data about, how the data was collected, and how the data was analyzed. [10]
  - **When:** When the case study was conducted, and when the data collection took place.
  - **Settings:** Where the tasks were conducted; Descriptions of the institution where the case study was conducted.
  - **Validity of results:** Internal and external validity of the case study.
  - **Involvement of the researcher:** If assistance is given by the researcher during the case study, this should be reported.

Case Study Use

Case studies can be observational, descriptive, or relational [27, pp. 14-15]. In addition, case studies can be used for validation of research results. Thus, there are many uses of this research method, e.g. for understanding, explaining or demonstrating the capabilities of technologies.

However, as stated by Sjøberg [24], the goal of software engineering research is "to support the private and public software industry in developing higher quality systems with improved timeliness in a more cost-effective and predictable way". One of the main contributions to this goal is research with the purpose of evaluating and comparing technologies. Kitchenham *et al.* [10] specifically suggest case studies as particularly important for industrial evaluation of software engineering methods and tools, due to the avoidance of scale-up problems.

[8] The list is inspired by Seaman [20] and Sjøberg *et al.* [23]. Sjøberg *et al.* [23] recommend a list of elements that should be reported accurately with aim of improving the review of articles, replication of experiments, meta-analysis, and theory building.

[9] The level of details must be considered according to the number of participants. It may be sufficient with general information if the number of participants is high. However, detailed characteristics may still be collected and stored in a case study database.

[10] The analysis in this thesis has not taken into consideration reporting on how data was analyzed.

As already encouraged by Segal *et al.* [21], software engineers should consider using research methods which take account of the complexity of context, in addition to methods which factor out the effect of context, such as laboratory studies.

Although there are situations where several research methods may be appropriate, there are situations where use of certain methods is more beneficial than others. When it comes to the case study method, Yin [27, p. 9] says that this would be when "a 'how' or 'why' question is being asked about a contemporary set of events over which the investigator has little or no control", and especially when "the boundaries between phenomenon and context are not clearly evident".

Additionally, Kitchenham *et al.* [10] emphasizes that a case study is the appropriate method to use when establishing a pilot project to assess the effects of change.

Moreover, Mohagheghi [17] says that quick changes in technology make it difficult to perform before-and-after evaluations.

Further, a case study is usually preferable to formal experiments if the process changes are very wide-ranging, meaning "that the effect of the change can be assessed only at a high level because the process change represents many detailed changes throughout the development process" [10]. The effects of the change cannot be identified immediately.

Proposal for Definition

Finally, I propose the following preliminary definition of the case study research method in empirical software engineering, based on the criteria recommended in Section 6.2.2. The following is a proposal for what to include in a future definition:

Def: A *case study* in software engineering is a set of systematic observations of the use of one or more software engineering technologies (processes, methods, techniques, guidelines or tools) in an industrial setting.

Def: A *multiple case study* in software engineering is a set of case studies conducted on the use of the same technologies in several companies or in several projects within the same company.

6.2.3 Examples of "Good" Case Studies

Yin regards an exemplary case study to be significant and "complete". The latter includes explicit attention given to the boundaries of the case, collection of the evidence, and absence of certain artifactual conditions. Further, the case study must consider alternative perspectives, display sufficient evidence, and finally, be composed in an engaging manner [27, pp. 160-165].

With respect to Yin's criteria for case studies, I would say that eight of the 50 articles report on case studies in this sense.

Below are some examples of what I regard as "good" case studies among the ones reported on in the articles analyzed. They are judged according to the criteria of Yin in addition to the criteria I presented in Section 6.2.2.

Table 14 Examples of Articles reporting on Case Studies

AID	Subject Type	Purpose	Type of Data	Affiliation of Authors	No of Questions Answered
A1	Professionals	Evaluative	Observations of Use	Research	5
A3	Professionals	Evaluative	Observations of Use	Research & Industry	5
A5	Professionals	Evaluative	Observations of Use	Research	5
A10	Professionals	Demonstrative	Observations of Use	Research & Industry	6
A13	Professionals	Evaluative	Observations of Use	Industry	5
A27	Professionals	Evaluative	Observations of Use	Research	5
A35	Professionals	Evaluative	Observations of Use	Research	6
A48	Professionals	Demonstrative	Observations of Use/ Technology Outcome	Industry	5

All of these provide answers to at least five of the six questions. Article A3 and A27 lack answers to *when* data was collected. Answers to *why* is missing in A1, A5 and A13, and finally, answers to *how* data was collected is absent in A48.

It appears that articles with a low response rate to the questions of focus in this review are usually not "good" case studies. This is especially true for attributes like, subject and methods for data collection.

7 Threats to Validity

The following chapter discusses the most important threats to the validity of the results found in this review.

7.1 Choice of Journals

The selection of twelve journals and conferences is the same selection as reviewed by Sjøberg *et al.* [23], who consider this selection to be leaders in software engineering in general and empirical software engineering in particular. The selection of journals is a superset of journals chosen by the other surveys described in Chapter 3. Hence, I believe that the journals and conferences chosen constitute a proper representation of the empirical software engineering field. However, due to the procedure used for selecting articles and the fact that the size and publication frequency of these journals and conferences differ, the findings are biased toward those that publish the most papers.

7.2 Selection of Articles

The review consisted of analyzing 50 articles that reported on case studies. These articles were found among 427 randomly selected articles. The search strategy that was used in order to select these 50 articles consisted of searching in Adobe Reader 7.0 for articles that included the words 'case study' or 'case studies'. This search strategy opens the risk of a validity threat consisting of the loss of good case studies that do not refer to themselves as case studies, yet still fulfill the criteria of being one. Nevertheless, as this thesis surveyed what researchers in empirical software engineering refer to as case studies, a minimum criterion should be that the author expresses that he/she actually is reporting on a case study.

7.3 Data Extraction

During the analysis, I extracted data from 50 articles. The data provided answers to various questions of qualitative nature. Considering the lack of standards for how to report case studies in empirical software engineering, extracting answers to the six questions and furthermore classifying what kind of data that was being reported was occasionally difficult. It was not always obvious what the answers to the questions were. Hence, the data in the articles had to be interpreted. Due to the fact that the analysis was carried out by one person only, the data may include subjective interpretations of the qualitative data. A common way of addressing this validity threat is to have data extraction performed independently by several reviewers which then can be compared and discussed [14]. The possible misclassifications imply that the quantitative results may include some errors.

The reporting of data collection appeared to be rather diverse and poor. Therefore, in order to reduce overlooked data regarding data collection, I performed PDF-searches after analyzing the articles on the terms 'interview', 'questionnaire', 'observation', 'ethnographic/ethnography', 'time sheet', 'effort data', and 'record'.

7.4 PDF-Search

There may be articles that report on case studies, which were not included in the sample, because of errors in the PDF-search.

Not all of the articles that were randomly selected could be found in PDF-format. Hence, I had to manually search these papers for the occurrences of ‘case study’ and ‘case studies’. Thus, I may have overseen a few articles that included the selection criterion.

8 Conclusions

This thesis is a systematic review of how the case study research method is used in empirical software engineering research. This chapter restates the main issues of the reported work. Section 8.1 repeats the objective of the research. In Section 8.2, the main findings are presented. A summary of the discussion of the results is provided in Section 8.3. Finally, proposals for future work are stated in Section 8.4.

8.1 Objective of Research

The objective of this investigation was first of all to get an overview of the existing use of case studies in empirical software engineering. Secondly, the investigation should identify important aspects of case studies for researchers to give careful considerations when conducting and reporting of case studies.

Ultimately, the purpose is to increase the case study's status among researchers and make the profession understand the value of case studies as a research method when used in the right manner. In order to address these issues, I conducted a survey of 50 articles that report on case studies. The data collected during analysis of these articles, was used to answer the following research question:

> **RQ:** *What is the state of the art regarding the use of case studies in empirical software engineering?*

8.2 Findings

The following reports the main findings of this thesis:

- **SRQ 1:** *What is the extent of the use of case studies in empirical software engineering?*

  The extent of case studies was close to twelve percent of the 427 articles randomly selected among the 5 453 articles scanned and analyzed by Sjøberg *et al.* [23].

- **SRQ 2:** *What is the general impression of the quality of reporting from case studies? Is data clearly presented?*

  Answers to the six questions from Seaman [20]:
  - **Who:** Although 70 percent stated who the subjects were, the description level was poor.
  - **What:** Information of tasks in the projects was reported in 64 percent of the articles.
  - **When:** 38 percent reported when data collection took place. However, the articles that reported this kind of information provided few details.
  - **Where:** 48 percent reported location of data collection.
  - **Why:** As few as 28 percent stated why the subjects participated.
  - **How:** 38 percent report method for data collection.

- **SRQ 3:** *Do researchers state the type of research method that they have used?*

  Specification of case study as the research method was found in 86 percent of the articles:
  - Title: 15 percent
  - Abstract: 42 percent
  - Keywords: 4 percent
  - Explicit elsewhere: 10 percent

  The remaining 14 percent express the use of the case study method for instance via headings.

- **SRQ 4:** *What is called a case study by the authors?*

  Purpose of case study reported in article:
  - Evaluative purpose: 42 percent
  - Demonstrative purpose: 58 percent

  Type of data reported from the case study:
  - Observations of use of technology: 36 percent
  - Outcome of technology: 54 percent
  - Observations of use of technology/Outcome of technology: 6 percent

- ***SRQ 5:*** *Is there a connection between what kind of data that is reported and the kind of affiliation of the author?*

  The following distribution of articles was found regarding the kind of affiliation of authors:
  - Research: 52 percent
  - Industry: 12 percent
  - Research & Industry: 36 percent

  There are more articles reporting on technology outcome (54 percent) than observations of use (36 percent). Interestingly, 54 percent of the articles with authors affiliated in research communities appear to report technology outcome, whereas 31 percent report observations of use.

  Nevertheless, the findings showed that there was no major trend when it comes to the distribution of articles according to affiliation of author and type of data.

- **SRQ 6:** *Are case studies confused with other research methods?*

  The lack of data collection may be reminiscent of the assertion method. I found some mixing of terms on research methods, mainly on *experiment* and *controlled experiment*.

- **SRQ 7:** *What is the extent of use of multiple case studies in empirical software engineering?*

  Claims of the use of multiple case studies are found in eleven of the 50 articles (22 percent).

8.3 Discussion

The results of this thesis show that there are great variances in the quality of reporting results. Thus, this thesis supports the conclusion that the software engineering community can do a better job in reporting its results [25, 29]. In particular, there was a lack of detailed and explicit reporting of central information in most of the articles. At a minimum, information about the subject, the tasks they performed and the environment in which the tasks were conducted is necessary in order for the readers to relate to the context. Additionally, descriptions of how data was collected should be reported.

Moreover, the profession of empirical software engineering is not very likely to *explicitly* state what kind of research method that has been used. Additionally, researchers are not always consistent when referring to research methods. Authors should make a clear statement about what kind of research method that is used in the research. In fact, I suggest such specification to be made a standard part of the abstract.

As the articles were analyzed, it felt natural to categorize the articles according to what purpose the case study seemed to have in the article and what kind of data that was collected. The findings of the survey detected that case studies are mainly used for two purposes, namely for evaluative and for demonstrative purposes.

Typical characteristics for articles with an evaluative nature are rather high response rates for the six questions in the survey, the reporting of observations of technology use, and most likely the use of professionals as subjects.

Typical characteristics for articles with a demonstrative nature are relatively low response rates for the six questions in the survey, the reporting of technology outcome, and most likely the use of authors of the articles as subjects.

The lack of data collection on observations of use in case studies with an evaluative purpose may be a hindrance for the acceptance by industry of the case study as a valuable research method. This is especially important as case study results may provide evaluative documentation that the industry can base their decisions on when deciding to use some technology in their development.

As we can see, the term ‘case study’ is not only used about different things in different disciplines [27], but also within the discipline of empirical software engineering. Thus, the findings may have discovered the need for a broader terminology that can describe the different methods that go under the name of ‘case study’.

In order to remove the negative reputation from this research method, it needs to be used carefully by researchers. In this way, the quality of the evaluative aspect of case studies can be increased. In order to produce results that are easy for reviewers and industry to relate to, there is a need for standards for how to conduct case studies. Use of guidelines would help researchers ensure the quality of the results. Hence, guidelines for assistance through the case study process will be an important device for improving future use of this research method. There are few guidelines today (examples of these guidelines is

provided by Arisholm *et al.* [2], Kitchenham [10] and Yin [27]) standard procedures specified for case studies in empirical software engineering must be settled.

Thus, inspired by Seaman [20], Sjøberg *et al.* [23], Kitchenham *et al.* [10], and Yin [27], in addition to what I have observed during this work, I have proposed criteria for case studies, what to report from case studies, and when to use the case study method.

Furthermore, it is necessary to settle an accurate definition of case studies in empirical software engineering in order to clearly specify what a case study is. This thesis contributes with a proposal for such a definition, which is as follows:

> **Def:** A *case study* in software engineering is a set of systematic observations of the use of one or more software engineering technologies (processes, methods, techniques, guidelines or tools) in an industrial setting.

> **Def:** A *multiple case study* in software engineering is a set of case studies conducted on the use of the same technologies in several companies or in several projects within the same company.

I hope the results of this thesis will be useful to researchers who conduct and report case studies. However, the thesis may also be of interest to the industry where the results of the case studies need to be interpreted. Furthermore, I hope that this thesis has stressed the relevance of standardizing the case study method enough in order to encourage development of specified guidelines and future use of such guidelines when conducting case studies.

Ultimately, I would like to quote McGrath *et al.* [16, p. 109]:

> *Never throw out a method just because someone has used it badly!*

8.4 Future Work

This research has shown that there is great differences in the way case studies are being conducted and reported in empirical software engineering. Hence, there is a need for a standardized way of conducting and reporting case studies. A tailored definition of case studies in empirical software engineering is a valuable step in this work. Such a definition should be derived in order to increase the quality of future research conducted by use of the case study method. Thus, a proposal for future work is to settle a definition of case studies for use in empirical software engineering. Furthermore, the empirical software engineering community should reach a consensus on a standard way of carrying out case studies.

A proposal for future work regarding the reporting of case studies is to investigate the reporting on the validity of case studies. Additionally, the way collected data is analyzed would also be of interest.

The results of Chapter 5 show that authors in the evaluations of technologies actually are likely to test the technologies themselves in what they call a case study. It would be interesting to investigate whether there is a trend of positive or negative results of such evaluations compared with evaluations conducted by others who did not evaluate the technology themselves.

A final proposal for future work is to examine trends identified in this research in more recently published controlled SE-experiments, i.e. published after 2002. For example, it could be of interest to find out whether the proportion of case studies has increased recently and whether the reporting has followed existing guidelines.

Bibliography

1. Anda, B. Empirical Studies of Construction and Application of Use Case Models, PhD Thesis, University of Oslo, 2003.
2. Arisholm, E., Anda, B., Jørgensen, M. & Sjøberg, D.I. Guidelines on Conducting Software Process Improvement Studies in Industry. In: 22nd IRIS Conference (Information Systems Research Seminar In Scandinavia). Keuruu, Finland, pp. 87-102, 1999.
3. Arisholm, E. Empirical Assessment of Changeability in Object-Oriented Software, PhD Thesis, University of Oslo, 2001.
4. Dybå, T. Enabling Software Process Improvement: An Investigation of the Importance of Organizational Issues, PhD Thesis, NTNU Trondheim, pp. 43-66, 2001.
5. Fenton, N.E. & Neil, M. A Critique of Software Defect Prediction Models, IEEE Transactions on software engineering, Vol. 25, No. 5, 1999.
6. Glass, R.L., Vessey, I. & Ramesh, V. Research in Software Engineering: An Analysis of the Literature. Information and Software Technology, Vol. 44, No. 8, pp. 491-506, June 2002.
7. Jarvinen, P. On Research Methods. ISBN 951-97113-6-8, 1999.
8. Juristo, N. & Moreno M.A. Lecture Notes on Empirical Software Engineering. World Scientific Publishing Co. Pte. Ltd, 2003.
9. Karahasanovic, A. Supporting Application Consistency in Evolving Object-Oriented Systems by Impact Analysis and Visualisation, PhD Thesis, University of Oslo, 2002.
10. Kitchenham, B.A., Pickard, L. & Pfleeger, S.L. Case Studies for Method and Tool Evaluation. IEEE Software, Vol. 12, No. 4, pp. 52-62, July 1995.
11. Kitchenham, B.A. Evaluating Software Engineering Methods and Tool – Part 1: The Evaluation Context and Evaluation Methods. Software Engineering Notes, Vol. 21, No. 1, pp. 11-15, January 1996.
12. Kitchenham, B.A., Jones, L. Evaluating Software Engineering Methods and Tool. Software Engineering Notes, Vol. 22, No. 4, pp. 21-24, July 1997.
13. Kitchenham, B.A., Pfleeger S.L., Hoaglin D.C. & Rosenberg J. Preliminary Guidelines for Empirical Research in Software Engineering. IEEE Transactions on Software Engineering, Vol. 28, No. 8, August 2002.
14. Kitchenham, B.A. Procedures for Performing Systematic Reviews. http://www.idi.ntnu.no/emner/empse/papers/kitchenham_2004.pdf, July 2004.
15. Kraemer, K.L. The Information Systems Research Challenge: Survey Research Methods. Boston, Harvard Business School, 1993.
16. McGrath, Martin & Kulka. Judgement Calls in Research.
17. Mohagheghi, P. The Impact of Software Reuse and Incremental Development on the Quality of Large Systems, Doctoral Theses at NTNU 2004:95.
18. Perry, D.E., Sim, S.E. & Easterbrook, S.M. Case Studies for Software Engineers. Proceedings of the 26th International Conference on Software Engineering, 2004.

19. Ramesh, V., Glass, R.L. & Vessey, I. Research in Computer Science: An Empirical Study. The Journal of Systems and Software, Vol. 70, No. 1-2, pp. 165–176, February 2004.
20. Seaman, C. Guidelines for Qualitative Research and Publication. http://attend.it.uts.edu.au/isern2005/slides/Carolyn.ppt, 2005.
21. Segal, J., Grinyer, A. & Sharp, H. The type of evidence produced by empirical software engineers. 2005.
22. Shadish, W.R., Cook, T.D. & Campbell, D.T. Experimental and Quasi-Experimental Designs for Generalized Causal Inference. Houghton Mifflin Company, 2002.
23. Sjøberg, D.I.K., Hannay, J.E., Hansen, O., Kampenes, V.B., Karahasanovic, A., Liborg, N.-K. & Rekdal, A.C. A Survey of Controlled Experiments in Software Engineering. IEEE Transactions on Software Engineering, Vol. 31, No. 9, pp. 1-21, September 2005.
24. Sjøberg, D.I.K. The Simula Approach to Experimentation in Software Engineering. http://www.simula.no/departments/engineering/publications/Sjoberg.2006.1/, 2006.
25. Tichy, W.F., Lukowicz, P., Prechelt, L. & Heinz, E.A. Experimental Evaluation in Computer Science: A Quantitative Study. Journals of Systems and Software, Vol. 28, No. 1, pp. 9-18, January 1995.
26. Wohlin, C., Runeson, P., Høst, M., Ohlsson, M.C., Regnell, B. & Wesslen, A. Experimentation in Software Engineering: An Introduction. Published by John Wiley \& Sons Inc., Kluwer Academic Publishers, 1999.
27. Yin, R. K. Case Study Research Design and Methods. Sage Publications, 2003.
28. Zelkowitz, M.V. & Wallace, D. Experimental Validation in Software Engineering. J. Information and Software Technology, Vol. 39, No. 11, pp. 735-743, 1997.
29. Zelkowitz, M.V. & Wallace, D. Experimental Models for Validating Computer Technology. IEEE Computer, 1998.
30. Zelkowitz, M.V., Wallace D.R. & Binkley, D.W. Lecture Notes on Empirical Software Engineering, published by World Scientific, editor: Juristo, N. & Moreno, A.M., Vol. 12, pp. 229-263, 2003.