Empir Software Eng (2009) 14:131–164
DOI 10.1007/s10664-008-9102-8

Guidelines for conducting and reporting case study research in software engineering

Per Runeson · Martin Höst

Published online: 19 December 2008

Editor: Dag Sjoberg

Abstract Case study is a suitable research methodology for software engineering research since it studies contemporary phenomena in its natural context. However, the understanding of what constitutes a case study varies, and hence the quality of the resulting studies. This paper aims at providing an introduction to case study methodology and guidelines for researchers conducting case studies and readers studying reports of such studies. The content is based on the authors' own experience from conducting and reading case studies. The terminology and guidelines are compiled from different methodology handbooks in other research domains, in particular social science and information systems, and adapted to the needs in software engineering. We present recommended practices for software engineering case studies as well as empirically derived and evaluated checklists for researchers and readers of case study research.

Keywords Case study · Research methodology · Checklists · Guidelines

1 Introduction

The acceptance of empirical studies in software engineering and their contributions to increasing knowledge is continuously growing. The analytical research paradigm is not sufficient for investigating complex real life issues, involving humans and their interactions with technology. However, the overall share of empirical studies is negligibly small in computer science research; Sjøberg et al. (2005), found 103 experiments in 5,453 articles Ramesh et al. (2004) and identified less than 2% experiments with human subjects, and only 0.16% field studies among 628 articles. Further, existing work on empirical research

P. Runeson (✉) · M. Höst
Department Computer Science, Lund University, Box 118, SE-221 00 Lund, Sweden
e-mail: per.runeson@cs.lth.se

M. Höst
e-mail: martin.host@cs.lth.se

methodology in software engineering has a strong focus on experimental research; the earliest by Moher and Schneider (1981), Basili et al. (1986), the first methodology handbook by Wohlin et al. (2000), and promoted by Tichy (1998). All have a tendency towards quantitative approaches, although also qualitative approaches are discussed during the later years, e.g. by Seaman (1999). There exist guidelines for experiments' conduct (Kitchenham et al. 2002; Wohlin et al. 2000) and reporting (Jedlitschka and Pfahl 2005), measurements (Basili and Weiss 1984; Fenton and Pfleeger 1996; van Solingen and Berghout 1999), and systematic reviews (Kitchenham 2007), while only little is written on case studies in software engineering (Höst and Runeson 2007; Kitchenham et al. 1995; Wohlin et al. 2003) and qualitative methods (Dittrich 2007; Seaman 1999; Sim et al. 2001). Recently, a comprehensive view of empirical research issues for software engineering has been presented, edited by Shull et al. (2008).

The term "case study" appears every now and then in the title of software engineering research papers. However, the presented studies range from very ambitious and well organized studies in the field, to small toy examples that claim to be case studies. Additionally, there are different taxonomies used to classify research. The term case study is used in parallel with terms like field study and observational study, each focusing on a particular aspect of the research methodology. For example, Lethbridge et al. use *field studies* as the most general term (Lethbridge et al. 2005), while Easterbrook et al. (2008) call *case studies* one of five "classes of research methods". Zelkowitz and Wallace propose a terminology that is somewhat different from what is used in other fields, and categorize project monitoring, case study and field study as *observational methods* (Zelkowitz and Wallace 1998). This plethora of terms causes confusion and problems when trying to aggregate multiple empirical studies.

The case study methodology is well suited for many kinds of software engineering research, as the objects of study are contemporary phenomena, which are hard to study in isolation. Case studies do not generate the same results on e.g. causal relationships as controlled experiments do, but they provide deeper understanding of the phenomena under study. As they are different from analytical and controlled empirical studies, case studies have been criticized for being of less value, impossible to generalize from, being biased by researchers etc. This critique can be met by applying proper research methodology practices as well as reconsidering that knowledge is more than statistical significance (Flyvbjerg 2007; Lee 1989). However, the research community has to learn more about the case study methodology in order to review and judge it properly.

Case study methodology handbooks are superfluously available in e.g. social sciences (Robson 2002; Stake 1995; Yin 2003) which literature also has been used in software engineering. In the field of information systems (IS) research, the case study methodology is also much more mature than in software engineering. For example, Benbasat et al. provide a brief overview of case study research in information systems (Benbasat et al. 1987), Lee analyzes case studies from a positivistic perspective (Lee 1989) and Klein and Myers do the same from an interpretive perspective (Klein and Myers 1999).

It is relevant to raise the question: what is specific for software engineering that motivates specialized research methodology? In addition to the specifics of the examples, the characteristics of software engineering objects of study are different from social science and also to some extent from information systems. The study objects are 1) private corporations or units of public agencies *developing* software rather than public agencies or private corporations *using* software systems; 2) project oriented rather than line or function oriented; and 3) the studied work is advanced engineering work conducted by highly educated people rather than routine work. Additionally, the software engineering research community has a pragmatic and result-oriented view on research methodology, rather than a philosophical stand, as noticed by Seaman (1999).

The purpose of this paper is to provide guidance for the researcher conducting case studies, for reviewers of case study manuscripts and for readers of case study papers. It is synthesized from general methodology handbooks, mainly from the social science field, as well as literature from the information systems field, and adapted to software engineering needs. Existing literature on software engineering case studies is of course included as well. The underlying analysis is done by structuring the information according to a general case study research process (presented in Section 2.4). Where different recommendations or terms appear, the ones considered most suited for the software engineering domain are selected, based on the authors' experience on conducting case studies and reading case study reports. Links to data sources are given by regular references. Specifically, checklists for researchers and readers are derived through a systematic analysis of existing checklists (Höst and Runeson 2007), and later evaluated by PhD students as well as by members of the International Software Engineering Research Network and updated accordingly.

This paper does not provide absolute statements for what is considered a "good" case study in software engineering. Rather it focuses on a set of issues that all contribute to the quality of the research. The minimum requirement for each issue must be judged in its context, and will most probably evolve over time. This is similar to the principles by Klein and Myers for IS case studies (Klein and Myers 1999), "it is incumbent upon authors, reviewers, and exercise their judgment and discretion in deciding whether, how and which of the principles should be applied". We do neither assess the current status of case study research in software engineering. This is worth a study on its own, similar to the systematic review on experiments by Sjøberg et al. (2005). Further, examples are used both to illustrate good practices and lack thereof.

This paper is outlined as follows. We first define a set of terms in the field of empirical research, which we use throughout the paper (Section 2.1), set case study research into the context of other research methodologies (Section 2.2) and discuss the motivations for software engineering case studies (Section 2.3). We define a case study research process (Section 2.4) and terminology (Section 2.5), which are used for the rest of the paper. Section 3 discusses the design of a case study and planning for data collection. Section 4 describes the process of data collection. In Section 5 issues on data analysis are treated, and reporting is discussed in Section 6. Section 7 discusses reading and reviewing case study report, and Section 8 summarizes the paper. Checklists for conducting and reading case study research are linked to each step in the case study process, and summarized in the Appendix.

Throughout the paper, we use three different case study examples to illustrate the methods. The examples are selected from the authors' publications, representing a variety of approaches within case study research. They illustrate solutions or identify problems in case study research, i.e. are not always compliant with the guidelines in this paper. The examples are presented in a format like this and they are denoted study XP, RE and QA after their research area on agile methods (extreme programming), requirements engineering and quality assurance, respectively. More information about the studies can be found in the original publications (Karlström and Runeson 2005; 2006*) (XP), (Regnell et al.* 2001*) (RE), and (Andersson and Runeson* 2007a, b*) (QA).*

2 Background and Definition of Concepts

2.1 Research Methodology

In order to set the scope for the type of empirical studies we address in this paper, we put case studies into the context of other research methodologies and refer to general definitions of the term case study according to Robson (2002), Yin (2003) and Benbasat et al. (1987) respectively.

The three definitions agree on that case study is an empirical method aimed at *investigating contemporary phenomena in their context*. Robson calls it a research strategy and stresses the use of *multiple sources of evidence*, Yin denotes it an inquiry and remarks that *the boundary between the phenomenon and its context may be unclear*, while Benbasat et al. make the definitions somewhat more specific, mentioning *information gathering from few entities* (people, groups, organizations), and the *lack of experimental control*.

There are three other major research methodologies which are related to case studies:

- Survey, which is the "collection of standardized information from a specific population, or some sample from one, usually, but not necessarily by means of a questionnaire or interview" (Robson 2002).
- Experiment, or controlled experiment, which is characterized by "measuring the effects of manipulating one variable on another variable" (Robson 2002) and that "subjects are assigned to treatments by random."(Wohlin et al. 2000). Quasi-experiments are similar to controlled experiments, except that subjects are not randomly assigned to treatments. Quasi-experiments conducted in an industry setting may have many characteristics in common with case studies.
- Action research, with its purpose to "influence or change some aspect of whatever is the focus of the research" (Robson 2002), is closely related to case study. More strictly, a case study is purely observational while action research is focused on and involved in the change process. In software process improvement (Dittrich et al. 2008; Iversen et al. 2004) and technology transfer studies (Gorschek et al. 2006), the research method should be characterized as action research. However, when studying the effects of a change, e.g. in pre- and post-event studies, we classify the methodology as case study. In IS, where action research is widely used, there is a discussion on finding the balance between action and research, see e.g. (Avison et al. 2001; Baskerville and Wood-Harper 1996). For the research part of action research, these guidelines apply as well.

Easterbrook et al. (2008) also count ethnographic studies among the major research methodologies. We prefer to consider ethnographic studies as a specialized type of case studies with focus on cultural practices (Easterbrook et al. 2008) or long duration studies with large amounts of participant-observer data (Klein and Myers 1999). Zelkowitz and Wallace define four different "observational methods" in software engineering (Zelkowitz and Wallace 1998); *project monitoring, case study, assertion* and *field study*. Our guidelines apply to all these, except assertion which is not considered a proper research method. In general, the borderline between the types of study is not always distinct. We prefer to see *project monitoring* as a part of a case study and *field studies* as multiple case studies. Robson summarizes his view, which seems functional in software engineering as well: "Many flexible design studies, although not explicitly labeled as such, can be usefully viewed as case studies." (Robson 2002) p 185.

Finally, a case study may contain elements of other research methods, e.g. a survey may be conducted within a case study, literature search often precede a case study and archival analyses may be a part of its data collection. Ethnographic methods, like interviews and observations are mostly used for data collection in case studies.

2.2 Characteristics of Research Methodologies

Different research methodologies serve different purposes; one type of research methodology does not fit all purposes. We distinguish between four types of purposes for research based on Robson's (2002) classification:

- Exploratory—finding out what is happening, seeking new insights and generating ideas and hypotheses for new research.
- Descriptive—portraying a situation or phenomenon.
- Explanatory—seeking an explanation of a situation or a problem, mostly but not necessary in the form of a causal relationship.[1]
- Improving—trying to improve a certain aspect of the studied phenomenon.[2]

Case study methodology was originally used primarily for exploratory purposes, and some researchers still limit case studies for this purpose, as discussed by Flyvbjerg (2007). However, it is also used for descriptive purposes, if the generality of the situation or phenomenon is of secondary importance. Case studies may be used for explanatory purposes, e.g. in interrupted time series design (pre- and post-event studies) although the isolation of factors may be a problem. This involves testing of existing theories in confirmatory studies. Finally, as indicated above, case studies in the software engineering discipline often take an improvement approach, similar to action research; see e.g. the QA study (Andersson and Runeson 2007b).

Klein and Myers define three types of case study depending on the research perspective, positivist, critical and interpretive (Klein and Myers 1999). A *positivist* case study searches evidence for formal propositions, measures variables, tests hypotheses and draws inferences from a sample to a stated population, i.e. is close to the natural science research model (Lee 1989) and related to Robson's explanatory category. A *critical* case study aims at social critique and at being emancipatory, i.e. identifying different forms of social, cultural and political domination that may hinder human ability. Improving case studies may have a character of being critical. An *interpretive* case study attempts to understand phenomena through the participants' interpretation of their context, which is similar to Robson's exploratory and descriptive types. Software engineering case studies tend to lean towards a positivist perspective, especially for explanatory type studies.

Conducting research on real world issues implies a trade-off between level of control and degree of realism. The realistic situation is often complex and non-deterministic, which hinders the understanding of what is happening, especially for studies with explanatory purposes. On the other hand, increasing the control reduces the degree of realism, sometimes leading to the real influential factors being set outside the scope of the study.

[1] Easterbrook et al. distinguish between exploratory and confirmatory case studies. We interpret Robson's explanatory category being closely related to Easterbrook's confirmatory category.

[2] Robson denotes this category "emancipatory" in the social science context, while improvement is our adaptation to an engineering context.

Case studies are by definition conducted in real world settings, and thus have a high degree of realism, mostly at the expense of the level of control.

The data collected in an empirical study may be quantitative or qualitative. Quantitative data involves numbers and classes, while qualitative data involves words, descriptions, pictures, diagrams etc. Quantitative data is analyzed using statistics, while qualitative data is analyzed using categorization and sorting. Case studies tend mostly to be based on qualitative data, as these provide a richer and deeper description. However, a combination of qualitative and quantitative data often provides better understanding of the studied phenomenon (Seaman 1999), i.e. what is sometimes called "mixed methods" (Robson 2002).

The research process may be characterized as fixed or flexible according to Anastas and MacDonald (1994) and Robson (2002). In a fixed design process, all parameters are defined at the launch of the study, while in a flexible design process key parameters of the study may be changed during the course of the study. Case studies are typically flexible design studies, while experiments and surveys are fixed design studies. Other literature use the terms quantitative and qualitative *design* studies, for fixed and flexible design studies respectively. We prefer to adhere to the fixed/flexible terminology since it reduces the risk for confusion that a study with qualitative design may collect both qualitative and quantitative data. Otherwise it may be unclear whether the term qualitative refers to the data or the design of the study,

Triangulation is important to increase the precision of empirical research. Triangulation means taking different angles towards the studied object and thus providing a broader picture. The need for triangulation is obvious when relying primarily on qualitative data, which is broader and richer, but less precise than quantitative data. However, it is relevant also for quantitative data, e.g. to compensate for measurement or modeling errors. Four different types of triangulation may be applied (Stake 1995):

- Data (source) triangulation—using more than one data source or collecting the same data at different occasions.
- Observer triangulation—using more than one observer in the study.
- Methodological triangulation—combining different types of data collection methods, e.g. qualitative and quantitative methods.
- Theory triangulation—using alternative theories or viewpoints.

Table 1 shows an overview of the primary characteristics of the above discussed research methodologies

Yin adds specifically to the characteristics of a case study that it (Yin 2003):

- "copes with the technically distinctive situation in which there will be many more variables 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."

Table 1 Overview of research methodology characteristics

Methodology	Primary objective	Primary data	Design
Survey	Descriptive	Quantitative	Fixed
Case study	Exploratory	Qualitative	Flexible
Experiment	Explanatory	Quantitative	Fixed
Action research	Improving	Qualitative	Flexible

Hence, a case study will never provide conclusions with statistical significance. On the contrary, many different kinds of evidence, figures, statements, documents, are linked together to support a strong and relevant conclusion.

Perry et al. define similar criteria for a case study (Perry et al. 2005). It is expected that a case study:

- "Has research questions set out from the beginning of the study
- Data is collected in a planned and consistent manner
- Inferences are made from the data to answer the research question
- Explores a phenomenon, or produces an explanation, description, or causal analysis of it
- Threats to validity are addressed in a systematic way."

In summary, the key characteristics of a case study are that 1) it is of flexible type, coping with the complex and dynamic characteristics of real world phenomena, like software engineering, 2) its conclusions are based on a clear chain of evidence, whether qualitative or quantitative, collected from multiple sources in a planned and consistent manner, and 3) it adds to existing knowledge by being based on previously established theory, if such exist, or by building theory.

2.3 Why Case Studies in Software Engineering?

Case studies are commonly used in areas like psychology, sociology, political science, social work, business, and community planning (e.g. Yin 2003). In these areas case studies are conducted with objectives to increase knowledge about individuals, groups, and organizations, and about social, political, and related phenomena. It is therefore reasonable to compare the area of software engineering to those areas where case study research is common, and to compare the research objectives in software engineering to the objectives of case study research in other areas.

The area of software engineering involves development, operation, and maintenance of software and related artifacts, e.g. (Jedlitschka and Pfahl 2005). Research on software engineering is to a large extent aimed at investigating how this development, operation, and maintenance are conducted by software engineers and other stakeholders under different conditions. Software development is carried out by individuals, groups and organizations, and social and political questions are of importance for this development. That is, software engineering is a multidisciplinary area involving areas where case studies normally are conducted. This means that many research questions in software engineering are suitable for case study research.

The definition of case study in Section 2.1 focuses on studying phenomena in their context, especially when the boundary between the phenomenon and its context is unclear. This is particularly true in software engineering. Experimentation in software engineering has clearly shown, e.g. when trying to replicate studies, that there are many factors impacting on the outcome of a software engineering activity (Shull et al. 2002). Case studies offer an approach which does not need a strict boundary between the studied object and its environment; perhaps the key to understanding is in the interaction between the two?

2.4 Case Study Research Process

When conducting a case study, there are five major process steps to be walked through:

1. Case study design: objectives are defined and the case study is planned.
2. Preparation for data collection: procedures and protocols for data collection are defined.

3. Collecting evidence: execution with data collection on the studied case.
4. Analysis of collected data
5. Reporting

This process is almost the same for any kind of empirical study; compare e.g. to the processes proposed by Wohlin et al. (2000) and Kitchenham et al. (2002). However, as case study methodology is a flexible design strategy, there is a significant amount of iteration over the steps (Andersson and Runeson 2007b). The data collection and analysis may be conducted incrementally. If insufficient data is collected for the analysis, more data collection may be planned etc. However, there is a limit to the flexibility; the case study should have specific objectives set out from the beginning. If the objectives change, it is a new case study rather than a change to the existing one, though this is a matter of judgment as all other classifications. Eisenhardt adds two steps between 4 and 5 above in her process for building theories from case study research (Eisenhardt 1989) a) shaping hypotheses and b) enfolding literature, while the rest except for terminological variations are the same as above.

2.5 Definitions

In this paper, we use the following terminology. The overall *objective* is a statement of what is expected to be achieved in the case study. Others may use goals, aims or purposes as synonyms or hyponyms for objective. The objective is refined into a set of *research questions*, which are to be answered through the case study analysis. A case may be based on a *software engineering theory*. It is beyond the scope of this article to discuss in detail what is meant by a theory. However, Sjøberg et al., describe a framework for theories including constructs of interest, relations between constructs, explanations to the relations, and scope of the theory (Sjøberg et al. 2008). With this way of describing theories, software engineering theories include at least one construct from software engineering. A research question may be related to a *hypothesis* (sometimes called a proposition (Yin 2003)), i.e. a supposed explanation for an aspect of the phenomenon under study. Hypotheses may alternatively be generated from the case study for further research. The *case* is referred to as the object of the study (e.g. a project), and it contains one or more *units of analysis* (e.g. subprojects). *Data* is collected from the *subjects* of the study, i.e. those providing the information. Data may be *quantitative* (numbers, measurements) or *qualitative* (words, descriptions). A case study *protocol* defines the detailed procedures for collection and analysis of the raw data, sometimes called *field procedures*.

The guidelines for conducting case studies presented below are organized according to this process. Section 3 is about setting up goals for the case study and preparing for data collection, Section 4 discusses collection of data, Section 5 discusses data analysis and Section 6 provides some guidelines for reporting.

3 Case Study Design and Planning

3.1 Defining a Case

Case study research is of flexible type, as mentioned before. This does not mean planning is unnecessary. On the contrary, good planning for a case study is crucial for its success. There are several issues that need to be planned, such as what methods to use for data collection, what departments of an organization to visit, what documents to read, which persons to interview, how often interviews should be conducted, etc. These plans can be formulated in a case study protocol, see Section 3.2.

A plan for a case study should at least contain the following elements (Robson 2002):

- Objective—what to achieve?
- The case—what is studied?
- Theory—frame of reference
- Research questions—what to know?
- Methods—how to collect data?
- Selection strategy—where to seek data?

The objective of the study may be, for example, exploratory, descriptive, explanatory, or improving. The objective is naturally more generally formulated and less precise than in fixed research designs. The objective is initially more like a focus point which evolves during the study. The research questions state what is needed to know in order to fulfill the objective of the study. Similar to the objective, the research questions evolve during the study and are narrowed to specific research questions during the study iterations (Andersson and Runeson 2007b).

The case may in general be virtually anything which is a "contemporary phenomenon in its real-life context" (Yin 2003). In software engineering, the case may be a software development project, which is the most straightforward choice. It may alternatively be an individual, a group of people, a process, a product, a policy, a role in the organization, an event, a technology, etc. The project, individual, group etc. may also constitute a unit of analysis within a case. In the information systems field, the case may be "individuals, groups…or an entire organization. Alternatively, the unit of analysis may be a specific project or decision"(Benbasat et al. 1987). Studies on "toy programs" or similarly are of course excluded due to its lack of real-life context. Yin (2003) distinguishes between *holistic case studies*, where the case is studied as a whole, and *embedded case studies* where multiple units of analysis are studied within a case, see Fig. 1. Whether to define a study consisting of two cases as holistic or embedded depends on what we define as the context and research goals. In our XP example, two projects are studied in two different companies in two different application domains, both using agile practices (Karlström and Runeson 2006). The projects may be considered two units of analysis in an embedded case study if the context is software companies in general and the research goal is to study agile practices. On the contrary, if the context is considered being the specific company or application domain, they have to be seen as two separate holistic cases. Benbasat et al. comment on a specific case study, "Even though this study appeared to be a single-case, embedded unit analysis, it could be considered a multiple-case design, due to the centralized nature of the sites." (Benbasat et al. 1987).

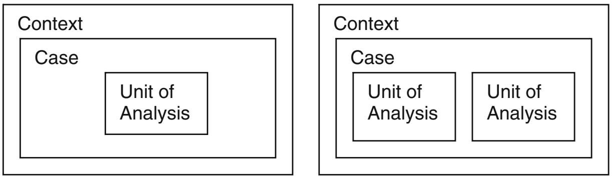


Fig. 1 Holistic case study (*left*) and embedded case study (*right*)

Using theories to develop the research direction is not well established in the software engineering field, as concluded in a systematic review on the topic (Hannay et al. 2007; Shull and Feldman 2008). However, defining the frame of reference of the study makes the context of the case study research clear, and helps both those conducting the research and those reviewing the results of it. As theories are underdeveloped in software engineering, the frame of reference may alternatively be expressed in terms of the viewpoint taken in the research and the background of the researchers. Grounded theory case studies naturally have no specified theory (Corbin and Strauss 2008).

The principal decisions on methods for data collection are defined at design time for the case study, although detailed decisions on data collection procedures are taken later. Lethbridge et al. (2005) define three categories of methods: direct (e.g. interviews), indirect (e.g. tool instrumentation) and independent (e.g. documentation analysis). These are further elaborated in Section 4.

In case studies, the case and the units of analysis should be selected intentionally. This is in contrast to surveys and experiments, where subjects are sampled from a population to which the results are intended to be generalized. The purpose of the selection may be to study a case that is expected to be “typical”, “critical”, “revelatory” or “unique” in some respect (Benbasat et al. 1987), and the case is selected accordingly. Flyvbjerg defines four variants of information-oriented case study selections: “extreme/deviant”, “maximum variation”, “critical” and “paradigmatic” (Flyvbjerg 2007). In a comparative case study, the units of analysis must be selected to have the variation in properties that the study intends to compare. However, in practice, many cases are selected based on availability (Benbasat et al. 1987) as is the case for many experiments (Sjøberg et al. 2005).

Case selection is particularly important when replicating case studies. A case study may be *literally replicated*, i.e. the case is selected to predict similar results, or it is *theoretically replicated*, i.e. the case is selected to predict contrasting results for predictable reasons (Yin 2003).

There were different objectives of the three example cases. The objective of study XP was to investigate how an agile process can coexist with a stage-gate management organization. The objective of study RE was to evaluate a method for prioritization of requirements, and the objective of study QA was to find quantitative prediction models and procedures for defect data.

Study XP is considered an embedded case study with two units of analysis from two different companies, although it might be seen as two holistic case studies, as denoted above. RE is a holistic case study with one unit of analysis, while QA is an embedded case study in one company with three different projects as units of analysis. All the companies were selected based on existing academia-industry relations, while the units of analysis were selected to fit the specific case study purposes.

Concerning the frame of reference, no explicit theories are referred to in studies XP and RE. However, the investigated approaches are based on existing methods that, to some extent, already have been investigated. Earlier studies thereby affected the designs of the studies. Study QA was partly a replication, which means that the original study formed a frame of reference from which theories on, for example, the Pareto principle and fault persistence between test phases were used when hypotheses were defined.

Data were primarily collected using interviews in the XP case. In the RE case, questionnaires constituted the major source of data, while in the QA case, defect metrics from a company was the major data source.

3.2 Case Study Protocol

The case study protocol is a container for the design decisions on the case study as well as *field procedures* for its carrying through. The protocol is a continuously changed document that is updated when the plans for the case study are changed.

There are several reasons for keeping an updated version of a case study protocol. Firstly, it serves as a guide when conducting the data collection, and in that way prevents the researcher from missing to collect data that were planned to be collected. Secondly, the processes of formulating the protocol makes the research concrete in the planning phase, which may help the researcher to decide what data sources to use and what questions to ask. Thirdly, other researchers and relevant people may review it in order to give feedback on the plans. Feedback on the protocol from other researchers can, for example, lower the risk of missing relevant data sources, interview questions or roles to include in the research and to assure the relation between research questions and interview questions. Finally, it can serve as a log or diary where all conducted data collection and analysis is recorded together with change decisions based on the flexible nature of the research. This can be an important source of information when the case study later on is reported. In order to keep track of changes during the research project, the protocol should be kept under some form of version control.

Pervan and Maimbo propose an outline of a case study protocol, which is summarized in Table 2. As the proposal shows, the protocol is quite detailed to support a well structured research approach.

Case study protocols cannot be published in extenso since they contain confidential information. However, parts of the protocol can be published, such as interview instruments, which is the case in study XP. In study QA, a logbook was kept which documents the iterations of the case study. A condensed version of the logbook is shown below as published (Andersson and Runeson 2007b), which shows seven case study cycles, indicating the evolutionary characteristic of the case study.

#	Goals and scope	Data collection and filtering	Analysis and presentation	Interpretation and improvement
1	*Simulation model*	*Process models* *Time reports* *Failure reports*	*Build simulation model*	*Too complex approach - not completed*
2	*Exploratory*	*Failure reports project 1, 2*	*Distribution of detection activities over time*	*Response on specific events in each project*
3	*Exploratory*	*Failure reports per feature group*	*Distribution of detection activities per feature group*	*Motivation for distribution*
4	*Confirmatory*	*Failure reports project 3*	*Same as in cycle 2 and 3*	*Sufficient fit for practical use*
5	*Explanatory*	*Qualitative data on feature groups*	*Characteristics of feature groups*	*Root cause analysis on causes and suggestions for each group* *Subset of failure reports*
6	*Explanatory (prediction)*	*All failure reports*	*Prediction of defect content with simple model*	*Use of prediction model to improve planning*
7	*Explanatory (prediction)*	*All failure reports* *Time data*	*Software reliability growth models*	*Use of prediction model*

Table 2 Outline of case study protocol according to Pervan and Maimbo (2005)

Section	Content
Preamble	Contains information about the purpose of the protocol, guidelines for data and document storage, publication.
General	Provides a brief overview of the research project and the case research method
Procedures	Detailed description of the procedures for conducting each case, including down-to-earth details on contacts and timing.
Research instrument(s)	Interview guides, questionnaires etc. to be used to ensure consistent data collection.
Data analysis guidelines	Detailed description of data analysis procedures, including data schemas, priori codes etc.
Appendix A	Template letter to invite participants.

3.3 Ethical Considerations

At design time of a case study, ethical considerations must be made (Singer and Vinson 2002). Even though a research study first and foremost is built on trust between the researcher and the case (Amschler Andrews and Pradhan 2001), explicit measures must be taken to prevent problems. In software engineering, case studies often include dealing with confidential information in an organization. If it is not clear from the beginning how this kind of information is handled and who is responsible for accepting what information to publish, there may be problems later on. Key ethical factors include:

- Informed consent
- Review board approval
- Confidentiality
- Handling of sensitive results
- Inducements
- Feedback

Subjects and organizations must explicitly agree to participate in the case study, i.e. give informed consent. In some countries, this is even legally required. It may be tempting for the researcher to collect data e.g. through indirect or independent data collection methods, without asking for consent. However, the ethical standards must be maintained for the long term trust in software engineering research.

Legislation of research ethics differs between countries and continents. In many countries it is mandatory to have the study proposal reviewed and accepted with respect to ethical issues (Seaman 1999) by a review board or a similar function at a university. In other countries, there are no such rules. Even if there are no such rules, it is recommended that the case study protocol is reviewed by colleagues to help avoiding pitfalls.

Consent agreements are preferably handled through a form or contract between the researchers and the individual participant, see e.g. Robson (2002) for an example. In an empirical study conduced by the authors of this paper, the following information were included in this kind of form:

- Names of researchers and contact information.
- Purpose of empirical study.
- Procedures used in the empirical study, i.e. a short description of what the participant should do during the study and what steps the researcher will carry out during these activities.

- A text clearly stating that the participation is voluntary, and that collected data will be anonymous.
- A list of known risks.
- A list of benefits for the participants, in this case for example experience from using a new technique and feedback effectiveness.
- A description of how confidentiality will be assured. This includes a description of how collected material will be coded and identified in the study.
- Information about approvals from review board.
- Date and signatures from participant and researchers.

If the researchers intend to use the data for other, not yet defined purposes, this should be signed separately to allow participants to choose if their contribution is for the current study only, or for possible future studies.

Issues on confidentiality and publication should also be regulated in a contract between the researcher and the studied organization. However, not only can information be sensitive when leaking outside a company. Data collected from and opinions stated by individual employees may be sensitive if presented e.g. to their managers (Singer and Vinson 2002). The researchers must have the right to keep their integrity and adhere to agreed procedures in this kind of cases. Companies may not know academic practices for publication and dissemination, and must hence be explicitly informed about those. From a publication point of view, the relevant data to publish is rarely sensitive to the company since data may be made anonymous. However, it is important to remember that it is not always sufficient to remove names of companies or individuals. They may be identified by their characteristics if they are selected from a small set of people or companies.

Results may be sensitive to a company, e.g. by revealing deficiencies in their software engineering practices, or if their product comes out last in a comparison (Amschler Andrews and Pradhan 2001). The chance that this may occur must be discussed upfront and made clear to the participants of the case study. In case violations of the law are identified during the case study, these must be reported, even though "whistle-blowers" rarely are rewarded.

The inducements for individuals and organizations to participate in a case study vary, but there are always some kinds of incentives, tangible or intangible. It is preferable to make the inducements explicit, i.e. specify what the incentives are for the participants. Thereby the inducement's role in threatening the validity of the study may also be analyzed.

Giving feedback to the participants of a study is critical for the long term trust and for the validity of the research. Firstly, transcript of interviews and observations should be sent back to the participants to enable correction of raw data. Secondly, analyses should be presented to them in order to maintain their trust in the research. Participants must not necessarily agree in the outcome of the analysis, but feeding back the analysis results increases the validity of the study.

In all three example studies issues of confidentiality were handled through Non-Disclosure Agreements and general project cooperation agreements between the companies and the university, lasting longer than one case study. These agreements state that the university researchers are obliged to have publications approved by representatives of the companies before they are published, and that raw data must not be spread to any but those signing the contract. The researchers are not obliged to report their sources of facts to management, unless it is found that a law is violated.

In order to ensure that interviewees were not cited wrongly, it was agreed that the transcribed interviews were sent back to them for review in the XP study. In the beginning of each interview, interviewees were informed about their rights in the study. In study QA, feedback meetings for analysis and interpretation were explicitly a part of the methodology ((Andersson and Runeson 2007b) Fig. 1). *When negotiating publication*

of data, we were explicitly told that raw numbers of defects could not be published, but percentages over phases could, which was acceptable for the research purposes.
All the three studies were conducted in Sweden, where only studies in medicine are explicitly regulated by law; hence there was no approval of the studies by a review board beforehand.

3.4 Checklist

The checklist items for case study design are shown in Table 3.

4 Collecting Data

4.1 Different Data Sources

There are several different sources of information that can be used in a case study. It is important to use several data sources in a case study in order to limit the effects of one interpretation of one single data source. If the same conclusion can be drawn from several sources of information, i.e. triangulation (Section 2.2), this conclusion is stronger than a conclusion based a single source. In a case study it is also important to take into account viewpoints of different roles, and to investigate differences, for example, between different projects and products. Commonly, conclusions are drawn by analyzing differences between data sources.

According to Lethbridge et al. (2005) data collection techniques can be divided into three levels:

- First degree: Direct methods means that the researcher is in direct contact with the subjects and collect data in real time. This is the case with, for example interviews, focus groups, Delphi surveys (Dalkey and Helmer 1963), and observations with "think aloud protocols".
- Second degree: Indirect methods where the researcher directly collects raw data without actually interacting with the subjects during the data collection. This approach is, for example taken in Software Project Telemetry (Johnson et al. 2005) where the usage of software engineering tools is automatically monitored, and observed through video recording.

Table 3 Case study design checklist items

1. What is the case and its units of analysis?
2. Are clear objectives, preliminary research questions, hypotheses (if any) defined in advance?
3. Is the theoretical basis—relation to existing literature or other cases—defined?
4. Are the authors' intentions with the research made clear?
5. Is the case adequately defined (size, domain, process, subjects…)?
6. Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors using the proposed design?
7. Does the design involve data from multiple sources (data triangulation), using multiple methods (method triangulation)?
8. Is there a rationale behind the selection of subjects, roles, artifacts, viewpoints, etc.?
9. Is the specified case relevant to validly address the research questions (construct validity)?
10. Is the integrity of individuals/organizations taken into account?

- Third degree: Independent analysis of work artifacts where already available and sometimes compiled data is used. This is for example the case when documents such as requirements specifications and failure reports from an organization are analyzed or when data from organizational databases such as time accounting is analyzed.

First degree methods are mostly more expensive to apply than second or third degree methods, since they require significant effort both from the researcher and the subjects. An advantage of first and second degree methods is that the researcher can to a large extent exactly control what data is collected, how it is collected, in what form the data is collected, which the context is etc. Third degree methods are mostly less expensive, but they do not offer the same control to the researcher; hence the quality of the data is not under control either, neither regarding the original data quality nor its use for the case study purpose. In many cases the researcher must, to some extent, base the details of the data collection on what data is available. For third degree methods it should also be noticed that the data has been collected and recorded for another purpose than that of the research study, contrary to general metrics guidelines (van Solingen and Berghout 1999). It is not certain that requirements on data validity and completeness were the same when the data was collected as they are in the research study.

In Sections 4.2–4.5 we discuss specific data collection methods, where we have found interviews, observations, archival data and metrics being applicable to software engineering case studies (Benbasat et al. 1987; Yin 2003).

In study XP data is collected mainly through interviews, i.e. a first degree method. The evaluation of a proposed method in study RE involves filling out a form for prioritization of requirements. These forms were an important data source, i.e. a second order method. In study QA stored data in the form defect reporting metrics were used as a major source of data, i.e. a third degree method. All studies also included one or several feedback steps where the organizations gave feedback on the results, i.e. a first-degree data collection method. These data were complemented with second or third degree data, e.g. process models were used in studies XP and QA.

4.2 Interviews

Data collection through interviews is important in case studies. In interview-based data collection, the researcher asks a series of questions to a set of subjects about the areas of interest in the case study. In most cases one interview is conducted with every single subject, but it is possible to conduct group-interviews. The dialogue between the researcher and the subject(s) is guided by a set of interview questions.

The interview questions are based on the topic of interest in the case study. That is, the interview questions are based on the formulated research questions (but they are of course not formulated in the same way). Questions can be *open*, i.e. allowing and inviting a broad range of answers and issues from the interviewed subject, or *closed* offering a limited set of alternative answers.

Interviews can, for example, be divided into *unstructured*, *semi-structured* and *fully structured* interviews (Robson 2002). In an unstructured interview, the interview questions are formulated as general concerns and interests from the researcher. In this case the interview conversation will develop based on the interest of the subject and the researcher. In a fully structured interview all questions are planned in advance and all questions are asked in the same order as in the plan. In many ways, a fully structured interview is similar to a questionnaire-based survey. In a semi-structured interview, questions are planned, but they are not necessarily asked in the same order as they are listed. The development of the conversation in the interview can decide which order the different questions are handled, and the researcher can use the list of questions to be certain that all questions are handled. Additionally, semi-structured interviews allow for improvisation and exploration of the studied objects. Semi-structured interviews are common in case studies. The different types of interviews are summarized in Table 4.

An interview session may be divided into a number of phases. First the researcher presents the objectives of the interview and the case study, and explains how the data from the interview will be used. Then a set of introductory questions are asked about the background etc. of the subject, which are relatively simple to answer. After the introduction comes the main interview questions, which take up the largest part of the interview. If the interview contains personal and maybe sensitive questions, e.g. concerning economy, opinions about colleagues, why things went wrong, or questions related to the interviewees own competence (Hove and Anda 2005), special care must be taken. In this situation it is important that the interviewee is ensured confidentiality and that the interviewee trusts the interviewer. It is not recommended to start the interview with these questions or to introduce them before a climate of trust has been obtained. It is recommended that the major findings are summarized by the researcher towards the end of the interview, in order to get feedback and avoid misunderstandings.

Interview sessions can be structured according to three general principles, as outlined in Fig. 2 (Caroline Seaman, personal communication). The funnel model begins with open questions and moves towards more specific ones. The pyramid model begins with specific ones, and opens the questions during the course of the interview. The time-glass model begins with open questions, straightens the structure in the middle and opens up again towards the end of the interview.

During the interview sessions it is recommended to record the discussion in a suitable audio or video format. Even if notes are taken, it is in many cases hard to record all details, and it is impossible to know what is important to record during the interview. Possibly a dedicated and trained scribe may capture sufficient detail in real-time, but the recording should at least be done as a backup (Hove and Anda 2005). When the interview has been recorded it needs to be transcribed into text before it is analyzed. This is a time consuming task, but in many cases new insights are made during the transcription, and it is therefore not recommended that this task is conducted by anyone else than the researcher. In some cases it may be advantageous to have the transcripts reviewed by the interview subject. In this way questions about what was actually said can be sorted out, and the interview subject has the chance to point out if she does not agree with the interpretation of what was said or if she simply has changed her mind and wants to rephrase any part of the answers.

During the planning phase of an interview study it is decided whom to interview. Due to the qualitative nature of the case study it is recommended to select subjects based on

Table 4 Overview of interviews

	Unstructured	Semi-structured	Fully structured
Typical focus	How individuals qualitatively experience the phenomenon	How individuals qualitatively and quantitatively experience the phenomenon	Researcher seeks to find relations between constructs
Interview questions	Interview guide with areas to focus on	Mix of open and closed questions	Closed questions
Objective	Exploratory	Descriptive and explanatory	Descriptive and explanatory

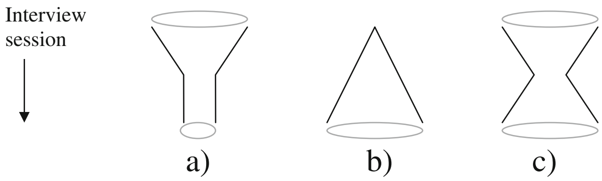


Fig. 2 General principles for interview sessions. **a** funnel, **b** pyramid, and **c** time-glass

differences instead of trying to replicate similarities, as discussed in Section 3.1. This means that it is good to try to involve different roles, personalities, etc in the interview. The number of interviewees has to be decided during the study. One criterion for when sufficient interviews are conducted is “saturation”, i.e. when no new information or viewpoint is gained from new subjects (Corbin and Strauss 2008).

Interviews were conducted in study XP. The researchers had an initial hypothesis about potential problems of combining agile methods with a traditional stage-gate model. However no details about this were known and the hypotheses were not detailed with respect to this. Hence a semi-structured approach was chosen, which supports the combination of exploratory and explanatory type of case study. An interview guide was developed, based on knowledge of agile and stage-gate models, together with the hypotheses of the study. The interviews were semi-structure, where the structure was given in terms of topics, which we wanted to cover and approximate time budget for each topic, see (Karlström and Runeson 2006) “Appendix A”. Relevant people to interview were identified in cooperation with the involved organizations. All interviewed persons were promised that only anonymous data would be presented externally and internally in the organization. Two researchers conducted most of the interviews together, which were audio recorded, and later transcribed. The interviewers also took notes on what they spontaneously found relevant.

4.3 Observations

Observations can be conducted in order to investigate how a certain task is conducted by software engineers. This is a first or second degree method according to the classification in Section 4.1. There are many different approaches for observation. One approach is to monitor a group of software engineers with a video recorder and later on analyze the recording, for example through protocol analysis (Owen et al. 2006; von Mayrhauser and Vans 1996). Another alternative is to apply a “think aloud” protocol, where the researcher are repeatedly asking questions like “What is your strategy?” and “What are you thinking?” to remind the subjects to think aloud. This can be combined with recording of audio and keystrokes as proposed e.g. by Wallace et al. (2002). Observations in meetings is another type, where meeting attendants interact with each other, and thus generate information about the studied object. An alternative approach is presented by Karahasanović et al. (2005) where a tool for sampling is used to obtain data and feedback from the participants.

Approaches for observations can be divided into high or low interaction of the researcher and high or low awareness of the subjects of being observed, see Table 5.

Table 5 Different approaches to observations.

	High awareness of being observed	Low awareness of being observed
High degree of interaction by the researcher	Case 1	Case 2
Low degree of interaction by the researcher	Case 3	Case 4

Observations according to case 1 or case 2 are typically conducted in action research or classical ethnographic studies where the researcher is part of the team, and not only seen as a researcher by the other team members. The difference between case 1 and case 2 is that in case 1 the researcher is seen as an "observing participant" by the other subjects, while she is more seen as a "normal participant" in case 2. In case 3 the researcher is seen only as a researcher. The approaches for observation typically include observations with first degree data collection techniques, such as a "think aloud" protocol as described above. In case 4 the subjects are typically observed with a second degree technique such as video recording (sometimes called video ethnography).

An advantage of observations is that they may provide a deep understanding of the phenomenon that is studied. Further, it is particularly relevant to use observations, where it is suspected that there is a deviation between an "official" view of matters and the "real" case (Robinson et al. 2007). It should however be noted that it produces a substantial amount of data which makes the analysis time consuming.

In the three example studies no extensive observations, e.g. through video recording or think-aloud procedures, were conducted. In a study, related to the XP study, Sharp and Robinson use observations (Sharp and Robinson 2004*). The observer spent 1 week with an XP team, taking part in everyday activities, including pair programming, i.e. an approach like Case 1 above. Data collected consisted of field notes, audio recordings of meetings and discussions, photographs and copies of artifacts.*

4.4 Archival Data

Archival data refers to, for example, meeting minutes, documents from different development phases, organizational charts, financial records, and previously collected measurements in an organization. Benbasat et al. (1987) and Yin (2003) distinguish between documentation and archival records, while we treat them together and see the borderline rather between qualitative data (minutes, documents, charts) and quantitative data (records, metrics), the latter discussed in Section 4.5.

Archival data is a third degree type of data that can be collected in a case study. For this type of data a configuration management tool is an important source, since it enables the collection of a number of different documents and different versions of documents. As for other third degree data sources it is important to keep in mind that the documents were not originally developed with the intention to provide data to research in a case study. A document may, for example, include parts that are mandatory according to an organizational template but of lower interest for the project, which may affect the quality of that part. It should also be noted that it is possible that some information that is needed by the researcher may be missing, which means that archival data analysis must be combined with other data collection techniques, e.g. surveys, in order to obtain missing historical factual data (Flynn et al. 1990). It is of course hard for the researcher to assess the quality of the data, although some information can be obtained by investigating the purpose of the original data collection, and by interviewing relevant people in the organization.

In study QA, archival data was a major source of information. Three different projects from one organization were studied. One of the projects was conducted prior to the study, which meant that the data from this project was analyzed in retrospect. We studied process models as well as project specifications and reports. In study XP, archival data in the form of process models were used as complementary sources of information.

4.5 Metrics

The above mentioned data collection techniques are mostly focused on qualitative data. However, quantitative data is also important in a case study. Software measurement is the process of representing software entities, like processes, products, and resources, in quantitative numbers (Fenton and Pfleeger 1996).

Collected data can either be defined and collected for the purpose of the case study, or already available data can be used in a case study. The first case gives, of course, most flexibility and the data that is most suitable for the research questions under investigation.

The definition of what data to collect should be based on a goal-oriented measurement technique, such as the Goal Question Metric method (GQM) (Basili and Weiss 1984; van Solingen and Berghout 1999). In GQM, goals are first formulated, and the questions are refined based on these goals, and after that metrics are derived based on the questions. This means that metrics are derived based on goals that are formulated for the measurement activity, and thus that relevant metrics are collected. It also implies that the researcher can control the quality of the collected data and that no unnecessary data is collected.

Examples of already available data are effort data from older projects, sales figures of products, metrics of product quality in terms of failures etc. This kind of data may, for example, be available in a metrics database in an organization. When this kind of data is used it should be noticed that all the problems are apparent that otherwise are solved with a goal oriented measurement approach. The researcher can neither control nor assess the quality of the data, since it was collected for another purpose, and as for other forms of archival analysis there is a risk of missing important data.

The archival data in study QA was mainly in the form of metrics collected from defect reporting and configuration management systems but also from project specifications. Examples of metrics that were collected are number of faults in modules, size of modules and duration for different test phases. In study XP, defect metrics were used as complementary data for triangulation purposes.

4.6 Checklists

The checklist items for preparation and conduct of data collection are shown in Tables 6 and 7, respectively.

Table 6 Preparation for data collection checklist items

11. Is a case study protocol for data collection and analysis derived (what, why, how, when)? Are procedures for its update defined?
12. Are multiple data sources and collection methods planned (triangulation)?
13. Are measurement instruments and procedures well defined (measurement definitions, interview questions)?
14. Are the planned methods and measurements sufficient to fulfill the objective of the study?
15. Is the study design approved by a review board, and has informed consent obtained from individuals and organizations?

Table 7 Collecting evidence checklist items

16. Is data collected according to the case study protocol?
17. Is the observed phenomenon correctly implemented (e.g. to what extent is a design method under study actually used)?
18. Is data recorded to enable further analysis?
19. Are sensitive results identified (for individuals, the organization or the project)?
20. Are the data collection procedures well traceable?
21. Does the collected data provide ability to address the research question?

5 Data Analysis

5.1 Quantitative Data Analysis

Data analysis is conducted differently for quantitative and qualitative data. For quantitative data, the analysis typically includes analysis of descriptive statistics, correlation analysis, development of predictive models, and hypothesis testing. All of these activities are relevant in case study research.

Descriptive statistics, such as mean values, standard deviations, histograms and scatter plots, are used to get an understanding of the data that has been collected. Correlation analysis and development of predictive models are conducted in order to describe how a measurement from a later process activity is related to an earlier process measurement. Hypothesis testing is conducted in order to determine if there is a significant effect of one or several variables (independent variables) on one or several other variables (dependent variables).

It should be noticed that methods for quantitative analysis assume a fixed research design. For example, if a question with a quantitative answer is changed halfway in a series of interviews, this makes it impossible to interpret the mean value of the answers. Further, quantitative data sets from single cases tend to be very small, due to the number of respondents or measurement points, which causes special concerns in the analysis.

Quantitative analysis is not covered any further in this paper, since it is extensively covered in other texts. The rest of this chapter covers qualitative analysis. For more information about quantitative analysis, refer for example to (Wohlin et al. 2000; Wohlin and Höst 2001; Kitchenham et al. 2002).

In study RE and study QC the main analyses were conducted with quantitative methods, mainly through analysis of correlation and descriptive statistics, such as scatter plots. In the QC case, the quantitative data acted as a trigger for deeper understanding. Patterns in the data, and lack thereof generated questions in the feedback session. The answers lead to changes in the data analysis, e.g. filtering out some data sources, and to identification of real patterns in the data.

In study XP, the main analysis was conducted with qualitative methods, but this was combined with a limited quantitative analysis of number of defects found during different years in one of the organizations. However, there would probably have been possibilities to conduct more complementary analyses in order to corroborate or develop the results from the qualitative analysis.

5.2 Qualitative Data Analysis

Since case study research is a flexible research method, qualitative data analysis methods (Seaman 1999) are commonly used. The basic objective of the analysis is to derive conclusions from the data, keeping a clear chain of evidence. The chain of evidence means that a reader should be able to follow the derivation of results and conclusions from the collected data (Yin 2003). This means that sufficient information from each step of the study and every decision taken by the researcher must be presented.

In addition to the need to keep a clear chain of evidence in mind, analysis of qualitative research is characterized by having analysis carried out in parallel with the data collection and the need for systematic analysis techniques. Analysis must be carried out in parallel with the data collection since the approach is flexible and that new insights are found during the analysis. In order to investigate these insights, new data must often be collected, and instrumentation such as interview questionnaires must be updated. The need to be systematic is a direct result of that the data collection techniques can be constantly updated, while the same time being required to maintain a chain of evidence.

In order to reduce bias by individual researchers, the analysis benefits from being conducted by multiple researchers. The preliminary results from each individual researcher is merged into a common analysis result in a second step. Keeping track and reporting the cooperation scheme helps increasing the validity of the study.

5.2.1 General Techniques for Analysis

There are two different parts of data analysis of qualitative data, hypothesis generating techniques and hypothesis confirmation techniques (Seaman 1999), which can be used for exploratory and explanatory case studies, respectively.

Hypothesis *generation* is intended to find hypotheses from the data. When using these kinds of techniques, there should not be too many hypotheses defined before the analysis is conducted. Instead the researcher should try to be unbiased and open for whatever hypotheses are to be found in the data. The results of these techniques are the hypotheses as such. Examples of hypotheses generating techniques are “constant comparisons” and “cross-case analysis” (Seaman 1999). Hypothesis *confirmation* techniques denote techniques that can be used to confirm that a hypothesis is really true, e.g. through analysis of more data. Triangulation and replication are examples of approaches for hypothesis confirmation (Seaman 1999). *Negative case analysis* tries to find alternative explanations that reject the hypotheses. These basic types of techniques are used iteratively and in combination. First hypotheses are generated and then they are confirmed. Hypothesis generation may take place within one cycle of a case study, or with data from one unit of analysis, and hypothesis confirmation may be done with data from another cycle or unit of analysis (Andersson and Runeson 2007b).

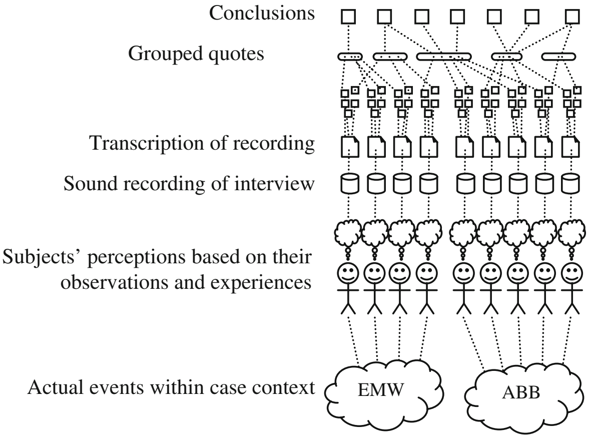
This means that analysis of qualitative data is conducted in a series of steps (based on (Robson 2002), p. 459). First the data is coded, which means that parts of the text can be given a code representing a certain theme, area, construct, etc. One code is usually assigned to many pieces of text, and one piece of text can be assigned more than one code. Codes can form a hierarchy of codes and sub-codes. The coded material can be combined with comments and reflections by the researcher (i.e. “memos”). When this has been done, the researcher can go through the material to identify a first set of hypotheses. This can, for example, be phrases that are similar in different parts of the material, patterns in the data, differences between sub-groups of subjects, etc. The identified hypotheses can then be used when further data collection is conducted in the field, i.e. resulting in an iterative approach where data collection and analysis is conducted in parallel as described above. During the iterative process a small set of generalizations can be formulated, eventually resulting in a formalized body of knowledge, which is the final result of the research attempt. This is, of course, not a simple sequence of steps. Instead, they are executed iteratively and they affect each other.

The activity where hypotheses are identified requires some more information. This is in no way a simple step that can be carried out by following a detailed, mechanical, approach. Instead it requires ability to generalize, innovative thinking, etc. from the researcher. This can be compared to quantitative analysis, where the majority of the innovative and analytical work of the researcher is in the planning phase (i.e. deciding design, statistical tests, etc). There is, of course, also a need for innovative work in the analysis of quantitative data, but it is not as clear as in the planning phase. In qualitative analysis there are major needs for innovative and analytical work in both phases.

One example of a useful technique for analysis is tabulation, where the coded data is arranged in tables, which makes it possible to get an overview of the data. The data can, for example be organized in a table where the rows represent codes of interest and the columns represent interview subjects. However, how to do this must be decided for every case study.

There are specialized software tools available to support qualitative data analysis, e.g. NVivo and Atlas. However, in some cases standard tools such as word processors and spreadsheet tools are useful when managing the textual data.

In study XP, the transcribed interviews were initially analyzed by one of the researchers. A preliminary set of codes were derived from the informal notes and applied to the transcripts. The preliminary set of codes was: project model, communication, planning, follow-up, quality, technical issues and attitudes. Each statement in the transcribed interviews was given a unique identification, and classified by two researchers. The transcribed data was then filled into tables, allowing for analysis of patterns in the data by sorting issues found by, for example, interviewee role or company. The chain of evidence is illustrated with the figure below (from Karlström and Runeson 2006)


Conclusions
Grouped quotes
Transcription of recording
Sound recording of interview
Subjects' perceptions based on their observations and experiences
Actual events within case context
EMW
ABB

In study RE and QA the main analysis was quantitative, although some qualitative analysis was conducted on the information that was gathered in feedback sessions. However, this analysis would probably benefit from being conducted in a more structured way, e.g. by recording, transcribing, and coding feedback data before analysis.

5.2.2 Level of Formalism

A structured approach is, as described above, important in qualitative analysis. This means, for example, in all cases that a pre-planned approach for analysis must be applied, all decisions taken by the researcher must be recorded, all versions of instrumentation must be kept, links between data, codes, and memos must be explicitly recorded in documentation, etc. However, the analysis can be conducted at different levels of formalism. In (Robson 2002) the following approaches are mentioned:

- Immersion approaches: These are the least structured approaches, with very low level of structure, more reliant on intuition and interpretive skills of the researcher. These approaches may be hard to combine with requirements on keeping and communicating a chain of evidence.
- Editing approaches: These approaches include few a priori codes, i.e. codes are defined based on findings of the researcher during the analysis.
- Template approaches: These approaches are more formal and include more a priori based on research questions.
- Quasi-statistical approaches: These approaches are much formalized and include, for example, calculation of frequencies of words and phrases.

To our experience editing approaches and template approaches are most suitable in software engineering case studies. It is hard to present and obtain a clear chain of evidence in informal immersion approaches. It is also hard to interpret the result of, for example, frequencies of words in documents and interviews.

Study XP used an editing approach. The analysis started with a set of codes (see Section 5.2.1*), which was extended and modified during the analysis. For example, the code "communication" was split into four codes: "horizontal communication", "vertical communication", "internal communication" and, "external communication".*

5.2.3 Validity

The validity of a study denotes the trustworthiness of the results, to what extent the results are true and not biased by the researchers' subjective point of view. It is, of course, too late to consider the validity during the analysis. The validity must be addressed during all previous phases of the case study. However, the validity is discussed in this section, since it cannot be finally evaluated until the analysis phase.

There are different ways to classify aspects of validity and threats to validity in the literature. Here we chose a classification scheme which is also used by Yin (2003) and similar to what is usually used in controlled experiments in software engineering (Wohlin et al. 2000). Some researchers have argued for having a different classification scheme for flexible design studies (credibility, transferability, dependability, confirmability), while we prefer to operationalize this scheme for flexible design studies, instead of changing the terms (Robson 2002). This scheme distinguishes between four aspects of the validity, which can be summarized as follows:

- Construct validity: This aspect of validity reflect to what extent the operational measures that are studied really represent what the researcher have in mind and what is investigated according to the research questions. If, for example, the constructs discussed in the interview questions are not interpreted in the same way by the researcher and the interviewed persons, there is a threat to the construct validity.

- Internal validity: This aspect of validity is of concern when causal relations are examined. When the researcher is investigating whether one factor affects an investigated factor there is a risk that the investigated factor is also affected by a third factor. If the researcher is not aware of the third factor and/or does not know to what extent it affects the investigated factor, there is a threat to the internal validity.
- External validity: This aspect of validity is concerned with to what extent it is possible to generalize the findings, and to what extent the findings are of interest to other people outside the investigated case. During analysis of external validity, the researcher tries to analyze to what extent the findings are of relevance for other cases. There is no population from which a statistically representative sample has been drawn. However, for case studies, the intention is to enable analytical generalization where the results are extended to cases which have common characteristics and hence for which the findings are relevant, i.e. defining a theory.
- Reliability: This aspect is concerned with to what extent the data and the analysis are dependent on the specific researchers. Hypothetically, if another researcher later on conducted the same study, the result should be the same. Threats to this aspect of validity is, for example, if it is not clear how to code collected data or if questionnaires or interview questions are unclear.

It is, as described above, important to consider the validity of the case study from the beginning. Examples of ways to improve validity are triangulation, developing and maintaining a detailed case study protocol, having designs, protocols, etc. reviewed by peer researchers, having collected data and obtained results reviewed by case subjects, spending sufficient time with the case, and giving sufficient concern to analysis of “negative cases”, i.e. looking for theories that contradict your findings.

In study XP, validity threats were analyzed based on a checklist by Robson (2002). *It would also have been possible to analyze threats according to construct validity, internal validity, external validity, and reliability. Countermeasures against threats to validity were then taken. For example, triangulation was achieved in different ways, results were reviewed by case representatives, and potential negative cases were identified by having two researchers working with the same material in parallel. It was also seen as important that sufficient time was spent with the organization in order to understand it. Even if the case study lasted for a limited time, this threat was lowered by the fact that the researchers had had a long-term cooperation with the organization before the presented case study.*

In study QA, e.g. data triangulation was used to check which phase the defect reports originated from. The alignment between the phase reported in the trouble report, and the person's tasks in the project organization was checked.

5.3 Checklist

The checklist items for analysis of collected data are shown in Table 8.

6 Reporting

An empirical study cannot be distinguished from its reporting. The report communicates the findings of the study, but is also the main source of information for judging the quality of the study. Reports may have different audiences, such as peer researchers, policy makers, research sponsors, and industry practitioners (Yin 2003). This may lead to the need of writing different reports for difference audiences. Here, we focus on reports with peer researchers as main audience, i.e. journal or conference articles and possibly accompanying

Table 8 Analysis of collected data checklist items

22. Is the analysis methodology defined, including roles and review procedures?
23. Is a chain of evidence shown with traceable inferences from data to research questions and existing theory?
24. Are alternative perspectives and explanations used in the analysis?
25. Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors in the analysis?
26. Are there clear conclusions from the analysis, including recommendations for practice/further research?
27. Are threats to the validity analyzed in a systematic way and countermeasures taken? (Construct, internal, external, reliability)

technical reports. Benbasat et al. propose that due to the extensive amount of data generated in case studies, "books or monographs might be better vehicles to publish case study research" (Benbasat et al. 1987).

Guidelines for reporting experiments have been proposed by Jedlitschka and Pfahl (2005) and evaluated by Kitchenham et al. (2008). Their work aims at defining a standardized reporting of experiments that enables cross-study comparisons through e.g. systematic reviews. For case studies, the same high-level structure may be used, but since they are more flexible and mostly based on qualitative data, the low-level detail is less standardized and more depending on the individual case. Below, we first discuss the characteristics of a case study report and then a proposed structure.

6.1 Characteristics

Robson defines a set of characteristics which a case study report should have (Robson 2002), which in summary implies that it should:

- tell what the study was about
- communicate a clear sense of the studied case
- provide a "history of the inquiry" so the reader can see what was done, by whom and how.
- provide basic data in focused form, so the reader can make sure that the conclusions are reasonable
- articulate the researcher's conclusions and set them into a context they affect.

In addition, this must take place under the balance between researcher's duty and goal to publish their results, and the companies' and individuals' integrity (Amschler Andrews and Pradhan 2001).

Reporting the case study objectives and research questions is quite straightforward. If they are changed substantially over the course of the study, this should be reported to help understanding the case.

Describing the case might be more sensitive, since this might enable identification of the case or its subjects. For example, "a large telecommunications company in Sweden" is most probably a branch of the Ericsson Corporation. However, the case may be better characterized by other means than application domain and country. Internal characteristics, like size of the studied unit, average age of the personnel, etc may be more interesting than external characteristics like domain and turnover. Either the case constitutes a small subunit of a large corporation, and then it can hardly be identified among the many subunits, or it is a small company and hence it is hard to identify it among many candidates. Still, care must be taken to find this balance.

Providing a “history of the inquiry” requires a level of substantially more detail than pure reporting of used methodologies, e.g. “we launched a case study using semi-structured interviews”. Since the validity of the study is highly related to what is done, by whom and how, it must be reported about the sequence of actions and roles acting in the study process. On the other hand, there is no room for every single detail of the case study conduct, and hence a balance must be found.

Data is collected in abundance in a qualitative study, and the analysis has as its main focus to reduce and organize data to provide a chain of evidence for the conclusions. However, to establish trust in the study, the reader needs relevant snapshots from the data that support the conclusions. These snapshots may be in the form of e.g. citations (typical or special statements), pictures, or narratives with anonymized subjects. Further, categories used in the data classification, leading to certain conclusions may help the reader follow the chain of evidence.

Finally, the conclusions must be reported and set into a context of implications, e.g. by forming theories. A case study can not be generalized in the meaning of being representative of a population, but this is not the only way of achieving and transferring knowledge. Conclusions can be drawn without statistics, and they may be interpreted and related to other cases. Communicating research results in terms of theories is an underdeveloped practice in software engineering (Hannay et al. 2007).

6.2 Structure

Yin proposes several alternative structures for reporting case studies in general (Yin 2003).

- Linear-analytic—the standard research report structure (problem, related work, methods, analysis, conclusions)
- Comparative—the same case is repeated twice or more to compare alternative descriptions, explanations or points of view.
- Chronological—a structure most suitable for longitudinal studies.
- Theory-building—presents the case according to some theory-building logic in order to constitute a chain of evidence for a theory.
- Suspense—reverts the linear-analytic structure and reports conclusions first and then backs them up with evidence.
- Unsequenced—with none of the above, e.g. when reporting general characteristics of a set of cases.

For the academic reporting of case studies which we focus on, the linear-analytic structure is the most accepted structure. The high level structure for reporting experiments in software engineering proposed by Jedlitschka and Pfahl (2005) therefore also fits the purpose of case study reporting. However, some changes are needed, based on specific characteristics of case studies and other issues based on an evaluation conducted by Kitchenham et al. (2008). The resulting structure is presented in Table 9. The differences and our considerations are presented below.

In a case study, the theory may constitute a framework for the analysis; hence, there are two kinds of related work: a) earlier studies on the topic and b) theories on which the current study is based.

The design section corresponds to the case study protocol, i.e. it reports the planning of the case study including the measures taken to ensure the validity of the study.

Table 9 Proposed reporting structure by Jedlitschka and Pfahl (2005) and modification proposed by Kitchenham et al. (2008) and adaptations to case study reporting, influenced by Robson (2002)

Experiments		Case studies	
Section heading	Subsections	Section headings	Subsections
Title		Title	
Authorship		Authorship	
Structured abstract		(Structured) abstract	
Motivation	Problem statement	Introduction	Problem statement
	Research objectives		Research objectives
	Context		Context
Related work		Related work	Earlier studies
			Theory
Experimental design	Goals, hypotheses and variables	Case study design	Research questions
	Design		Case and subjects selection
	Subjects		Data collection procedure(s)
	Objects		Analysis procedure(s)
	Instrumentation		Validity procedure(s)
	Data collection procedure		
	Analysis procedure		
	Evaluation of validity		
Execution	Sample	Results	Case and subjects description, covering execution, analysis and interpretation issues.
	Preparation		Subsections may be structured e.g. according to coding scheme, each linking observations to conclusions
	Data collection performed		Evaluation of validity
	Validity procedure		
Analysis	Descriptive statistics		
	Data set reduction		
	Hypothesis testing		
Interpretation	Evaluation of results and implications		
Conclusions and Future work	Relation to existing evidence	Conclusions and Future work	Summary of conclusions
	Impact		Relation to existing evidence
	Limitations		Impact/Implications
	Future work		Limitations
			Future work
Acknowledgements		Acknowledgements	
References		References	
Appendices		Appendices	

Since the case study is of flexible design, and data collection and analysis are more intertwined, these sections may be combined into one. Consequently, the contents at the lower level must be adjusted, as proposed in Table 9. Specifically for the combined data section, the coding scheme often constitutes a natural subsection structure. Alternatively, for a comparative case study, the data section may be structured according to the compared cases, and for a longitudinal study, the time scale may constitute the structure of the data section. This combined results section also includes an evaluation of the validity of the final results.

The case studies were presented in different formats. Study XP was, for example, presented to the involved companies in seminar format, and to the research community in journal format (Karlström and Runeson 2006*), to practitioners in a magazine format (Karlström and Runeson* 2005*), and in the form of a Ph.D. thesis (Karlström* 2004*). The journal format paper is structured similar to the proposed model above, although the outline hierarchy differs slightly.*

6.3 Checklist

The checklist items for reporting are shown in Table 10.

7 Reading and Reviewing Case Study Research

7.1 Reader's Perspective

The reader of a case study report—independently of whether the intention is to use the findings or to review it for inclusion in a journal—must judge the quality of the study based on the written material. Case study reports tend to be large, firstly since case studies often are based on qualitative data, and hence the data cannot be presented in condensed form, like quantitative data may be in tables, diagrams and statistics. Secondly, the conclusions in qualitative analyses are not based on statistical significance which can be interpreted in terms of a probability for erroneous conclusion, but on reasoning and linking of observations to conclusions.

Reviewing empirical research in general must be done with certain care (Tichy 2000). Reading case study reports requires judging the quality of the report, without having the power of strict criteria which govern experimental studies to a larger extent, e.g. statistical

Table 10 Reporting checklist items

28. Are the case and its units of analysis adequately presented?
29. Are the objective, the research questions and corresponding answers reported?
30. Are related theory and hypotheses clearly reported?
31. Are the data collection procedures presented, with relevant motivation?
32. Is sufficient raw data presented (e.g. real life examples, quotations)?
33. Are the analysis procedures clearly reported?
34. Are threats to validity analyses reported along with countermeasures taken to reduce threats?
35. Are ethical issues reported openly (personal intentions, integrity issues, confidentiality)
36. Does the report contain conclusions, implications for practice and future research?
37. Does the report give a realistic and credible impression?
38. Is the report suitable for its audience, easy to read and well structured?

confidence levels. This does however not say that any report can do as a case study report. The reader must have a decent chance of finding the information of relevance, both to judge the quality of the case study and to get the findings from the study and set them into practice or build further research on.

The criteria and guidance presented above for performing and reporting case studies are relevant for the reader as well. However, in our work with derivation of checklists for case study research (Höst and Runeson 2007), evaluation feedback identified a need for a more condensed checklist for readers and reviewers. This is presented in Table 11 with numbers referring to the items of the other checklists for more in depth criteria.

8 Summary

Case study research is conducted in order to investigate contemporary phenomena in their natural context. That is, no laboratory environment is set up by the researcher, where factors can be controlled. Instead the phenomena are studied in their normal context, allowing the researcher to understand how the phenomena interact with the context. Selection of subjects and objects is not based on statistically representative samples. Instead, research findings are obtained through the analysis in depth of typical or special cases.

Cases study research is conducted by iteration over a set of phases. In the design phase objectives are decided and the case is defined. Data collection is first planned with respect to data collection techniques and data sources, and then conducted in practice. Methods for data collection include, for example, interviews, observation, and usage of archival data. During the analysis phase, insights are both generated and analyzed, e.g. through coding of data and looking for patterns. During the analysis it is important to maintain a chain of evidence from the findings to the original data. The report should include sufficient data and examples to allow the reader to understand the chain of evidence.

Table 11 Reader's checklist items

39. Are the objective, research questions, and hypotheses (if applicable) clear and relevant? 1, 2, 5, 29, 30
40. Are the case and its units of analysis well defined? 1, 5, 28
41. Is the suitability of the case to address the research questions clearly motivated? 8, 9, 14
42. Is the case study based on theory or linked to existing literature? 3
43. Are the data collection procedures sufficient for the purpose of the case study (data sources, collection, validation)? 11, 13, 16, 18, 21, 31
44. Is sufficient raw data presented to provide understanding of the case and the analysis? 32
45. Are the analysis procedures sufficient for the purpose of the case study (repeatable, transparent)? 22, 33
46. Is a clear chain of evidence established from observations to conclusions? 6, 17, 20, 23, 25
47. Are threats to validity analyses conducted in a systematic way and are countermeasures taken to reduce threats? 27, 34, 37
48. Is triangulation applied (multiple collection and analysis methods, multiple authors, multiple theories)? 7, 12, 22, 24
49. Are ethical issues properly addressed (personal intentions, integrity, confidentiality, consent, review board approval)? 4, 10, 15, 19, 35
50. Are conclusions, implications for practice and future research, suitably reported for its audience? 26, 29, 36, 37, 38

The numbers after each item refer to corresponding items in the checklists for each process step

This paper aims to provide a frame of reference for researchers when conducting case study research in software engineering, which is based on an analysis of existing case study literature and the author's own experiences of conducting case studies. As with other guidelines, there is a need to evaluate them through practical usage.

Acknowledgement The authors are grateful to the feedback to the checklists from the ISERN members and IASESE attendants in September 2007. A special thank to Professor Claes Wohlin, Mr. Kim Weyns and Mr. Andreas Jedlitschka for their review of an earlier draft of the paper. Thanks also to the anonymous reviewers for proposals on substantial improvements. The work is partly funded by the Swedish Research Council under grant 622-2004-552 for a senior researcher position in software engineering.

Appendix A

Table 12 Researcher's checklist

Case study design

1. What is the case and its units of analysis?
2. Are clear objectives, preliminary research questions, hypotheses (if any) defined in advance?
3. Is the theoretical basis—relation to existing literature or other cases—defined?
4. Are the authors' intentions with the research made clear?
5. Is the case adequately defined (size, domain, process, subjects…)?
6. Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors using the proposed design?
7. Does the design involve data from multiple sources (data triangulation), using multiple methods (method triangulation)?
8. Is there a rationale behind the selection of subjects, roles, artifacts, viewpoints, etc.?
9. Is the specified case relevant to validly address the research questions (construct validity)?
10. Is the integrity of individuals/organizations taken into account?

Preparation for data collection

11. Is a case study protocol for data collection and analysis derived (what, why, how, when)? Are procedures for its update defined?
12. Are multiple data sources and collection methods planned (triangulation)?
13. Are measurement instruments and procedures well defined (measurement definitions, interview questions)?
14. Are the planned methods and measurements sufficient to fulfill the objective of the study?
15. Is the study design approved by a review board, and has informed consent obtained from individuals and organizations?

Collecting Evidence

16. Is data collected according to the case study protocol?
17. Is the observed phenomenon correctly implemented (e.g. to what extent is a design method under study actually used)?
18. Is data recorded to enable further analysis?
19. Are sensitive results identified (for individuals, the organization or the project)?
20. Are the data collection procedures well traceable?
21. Does the collected data provide ability to address the research question?

Analysis of collected data

22. Is the analysis methodology defined, including roles and review procedures?
23. Is a chain of evidence shown with traceable inferences from data to research questions and existing theory?

24. Are alternative perspectives and explanations used in the analysis?
25. Is a cause–effect relation under study? If yes, is it possible to distinguish the cause from other factors in the analysis?
26. Are there clear conclusions from the analysis, including recommendations for practice/further research?
27. Are threats to the validity analyzed in a systematic way and countermeasures taken? (Construct, internal, external, reliability)

Reporting

28. Are the case and its units of analysis adequately presented?
29. Are the objective, the research questions and corresponding answers reported?
30. Are related theory and hypotheses clearly reported?
31. Are the data collection procedures presented, with relevant motivation?
32. Is sufficient raw data presented (e.g. real life examples, quotations)?
33. Are the analysis procedures clearly reported?
34. Are threats to validity analyses reported along with countermeasures taken to reduce threats?
35. Are ethical issues reported openly (personal intentions, integrity issues, confidentiality)
36. Does the report contain conclusions, implications for practice and future research?
37. Does the report give a realistic and credible impression?
38. Is the report suitable for its audience, easy to read and well structured?

Appendix B

Table 13 Reader's checklist

39. Are the objective, research questions, and hypotheses (if applicable) clear and relevant? 1, 2, 5, 29, 30
40. Are the case and its units of analysis well defined? 1, 5, 28
41. Is the suitability of the case to address the research questions clearly motivated? 8, 9, 14
42. Is the case study based on theory or linked to existing literature? 3
43. Are the data collection procedures sufficient for the purpose of the case study (data sources, collection, validation)? 11, 13, 16, 18, 21, 31
44. Is sufficient raw data presented to provide understanding of the case and the analysis? 32
45. Are the analysis procedures sufficient for the purpose of the case study (repeatable, transparent)? 22, 33
46. Is a clear chain of evidence established from observations to conclusions? 6, 17, 20, 23, 25
47. Are threats to validity analyses conducted in a systematic way and are countermeasures taken to reduce threats? 27, 34, 37
48. Is triangulation applied (multiple collection and analysis methods, multiple authors, multiple theories)? 7, 12, 22, 24
49. Are ethical issues properly addressed (personal intentions, integrity, confidentiality, consent, review board approval)? 4, 10, 15, 19, 35
50. Are conclusions, implications for practice and future research, suitably reported for its audience? 26, 29, 36, 37, 38

The numbers after each item refer to corresponding items in the Researcher's checklist

References

Amschler Andrews A, Pradhan AS (2001) Ethical issues in empirical software engineering: the limits of policy. Empir Softw Eng 6(2):105–110 doi:10.1023/A:1011442319273

Anastas JW, MacDonald ML (1994) Research design for the social work and the human services. New York, Lexington

Andersson C, Runeson P (2007a) A replicated quantitative analysis of fault distribution in complex software systems. IEEE Trans Softw Eng 33(5):273–286 doi:10.1109/TSE.2007.1005

Andersson C, Runeson P (2007b) A spiral process model for case studies on software quality monitoring—method and metrics. Softw Process Improv Pract 12(2):125–140 doi:10.1002/spip.311

Avison D, Baskerville R, Myers M (2001) Controlling action research projects. Inf Technol People 14(1):28–45 doi:10.1108/09593840110384762

Basili VR, Weiss DM (1984) A methodology for collecting valid software engineering data. IEEE Trans Softw Eng SE10(6):728–739

Basili VR, Selby RW, Hutchens DH (1986) Experimentation in Software Engineering. IEEE Trans Softw Eng SE12(7):733–744

Baskerville RL, Wood-Harper AT (1996) A critical perspective on action research as method for information systems research. J Inf Technol 11:235–246 doi:10.1080/026839696345289

Benbasat I, Goldstein DK, Mead M (1987) The case research strategy in studies of information systems. MIS Q 11(3):369–386 doi:10.2307/248684

Corbin J, Strauss C (2008) Basics of qualitative research, 3rd edn. Sage

Dalkey N, Helmer O (1963) An experimental application of the delphi method to the use of experts. Manage Sci 9(3):458–467

Dittrich Y (ed) (2007) Special issue on qualitative software engineering research. Inf Softw Technol 49 (6):531–694. doi:10.1016/j.infsof.2007.02.009

Dittrich Y, Rönkkö K, Eriksson J, Hansson C, Lindeberg O (2008) Cooperative method development. combining qualitative empirical research with method, technique and process improvement. Empir Softw Eng 13(3):231–260 doi:10.1007/s10664-007-9057-1

Eisenhardt KM (1989) Building theories form case study research. Acad Manage Rev 14(4):532–550 doi:10.2307/258557

Easterbrook S, Singer J, Storey M-A, Damian D (2008) Selecting empirical methods for software engineering research, Chapter 11 in Shull et al. (2008)

Fenton N, Pfleeger SG (1996) *Software Metrics—A Rigorous and Practical Approach*, Thomson Computer

Flynn BB, Sakakibara S, Schroeder RG, Bates K, Flynn EJ (1990) Empirical research methods in operations management. Oper Manage 9(2):250–284 doi:10.1016/0272-6963(90)90098-X

Flyvbjerg B (2007) Five misunderstandings about case-study research. In *Qualitative Research Practice: Concise Paperback Edition*. Sage, pp 390–404

Gorschek T, Garre P, Larsson S, Wohlin C (2006) A model for technology transfer in practice. IEEE Softw 23(6):88–95 doi:10.1109/MS.2006.147

Hannay JE, Sjøberg DIK, Dybå TA (2007) Systematic review of theory use in software engineering experiments. IEEE Trans Softw Eng 33(2):87–107 doi:10.1109/TSE.2007.12

Hove SE, Anda BCD (2005) Experiences from conducting semi-structured interviews in empirical software engineering research. Proceedings 11th IEEE International Software Metrics Symposium (Metrics 2005) 23:1–10

Höst M, Runeson P (2007) Checklists for Software Engineering Case Study Research, In *Proceedings First International Symposium on Empirical Software Engineering and Measurement*, pp 479–481

Iversen JH, Mathiassen L, Nielsen PA (2004) Managing risk in software process improvement: an action research approach. MIS Q 28(3):395–433

Jedlitschka A, Pfahl D (2005) Reporting guidelines for controlled experiments in software engineering, In *Proceedings of ACM/IEEE International Symposium on Empirical Software Engineering*, pp 95–104, see also Chapter 8 in Shull et al. (2008)

Johnson P, Kou H, Paulding M, Zhang Q, Kagawa A, Yamashita T (2005) Improving software development management through software project telemetry. IEEE Softw 22(4):76–85 doi:10.1109/MS.2005.95

Karahasanović A, Anda B, Arisholm E, Hove SE, Jørgensen M, Sjøberg DIK, Welland R (2005) Collecting feedback during software engineering experiments. Empir Softw Eng 10:113–147 doi:10.1007/s10664-004-6189-4

Karlström D (2004) *Integrating Management and Engineering Processes in Software Product Development*, PhD Thesis ISRN LUTEDX/TETS—1069-SE+230p, Lund University.

Karlström D, Runeson P (2005) Combining agile methods with stage-gate project management. IEEE Softw 22(3):43–49 doi:10.1109/MS.2005.59

Karlström D, Runeson P (2006) Integrating agile software development into stage-gate product development. Empir Softw Eng 11:203–225 doi:10.1007/s10664-006-6402-8

Klein HK, Myers MD (1999) A set of principles for conducting and evaluating interpretative field studies in information systems. MIS Q 23(1):67–88 doi:10.2307/249410

Kitchenham B, Pickard L, Pfleeger SL (1995) Case studies for method and tool evaluation. IEEE Softw 4 (12):52–62 doi:10.1109/52.391832

Kitchenham B, Pfleeger SM, Pickard LM, Jones PW, Hoaglin DC, El Eman K, Rosenberg J (2002) Preliminary guidelines for empirical research in software engineering. IEEE Trans Softw Eng 28(8):721–734 doi:10.1109/TSE.2002.1027796

Kitchenham B (2007) *Guidelines for performing Systematic Literature Reviews in Software Engineering*, Version 2.3, EBSE Technical Report EBSE-2007-01, Keele University and University of Durham

Kitchenham B, Al-Khilidar H, Ali Babar M, Berry M, Cox K, Keung J, Kurniawati F, Staples M, Zhang H, Zhu L (2008) Evaluating guidelines for reporting empirical software engineering studies. Empir Softw Eng 13(1):97–121 doi:10.1007/s10664-007-9053-5

Lee AS (1989) A scientific methodology for MIS case studies. MIS Q 13(1):33–54 doi:10.2307/248698

Lethbridge TC, Sim SE, Singer J (2005) Studying software engineers: data collection techniques for software field studies. Empir Softw Eng 10(3):311–341 see also Chapter 1 in Shull et al. (2008)

Moher T, Schneider GM (1981) Methods for improving controlled experimentation in software engineering, *Proceedings of the 5th International Conference on Software Engineering* pp 224–233

Owen S, Budgen D, Brereton P (2006) Protocol analysis: a neglected practice. Commun ACM 49(2):117–122 doi:10.1145/1113034.1113039

Perry DE, Sim SE, Easterbrook S (2005) Case studies for software engineers, *29th Annual IEEE/NASA Software Engineering Workshop—Tutorial Notes* pp 96–159

Pervan G, Maimbo H (2005) Designing a case study protocol for application in IS research, *The Ninth Pacific Conference on Information Systems* pp 1281–1292

Ramesh V, Glass RL, Vessey I (2004) Research in computer science: an empirical study. J Syst Softw 70(1–2):165–176 doi:10.1016/S0164-1212(03)00015-3

Regnell B, Höst M, Natt och Dag J, Beremark P, Hjelm T (2001) An industrial case study on distributed prioritisation in market-driven requirements engineering for packaged software. Requirements Eng 6:51–62 doi:10.1007/s007660170015

Robinson H, Segal J, Sharp H (2007) Ethnographically-informed empirical studies of software practice. Inf Softw Technol 49:540–551 doi:10.1016/j.infsof.2007.02.007

Robson C (2002) *Real World Research*. Blackwell, (2nd edition)

Seaman C (1999) Qualitative methods in empirical studies of software engineering. IEEE Trans Softw Eng 25(4):557–572 see also Chapter 2 in Shull et al. (2008)

Sharp H, Robinson H (2004) An ethnographic study of XP practice. Empir Softw Eng 9(4):353–375 doi:10.1023/B:EMSE.0000039884.79385.54

Shull F, Feldman RL (2008) Building theories from multiple evidence sources. In: Shull F et al (ed) Guide to advanced empirical software engineering. Springer-Verlag, London

Shull F, Basili V, Carver J, Maldonado JC, Travassos GH, Mendonca M, Fabbri S (2002) Replicating software engineering experiments: addressing the tacit knowledge problem, *Proceedings on International Symposium Empirical Software Engineering* pp 7–16

Shull F, Singer J, Sjøberg D (eds) (2008) Guide to Advanced Empirical Software Engineering. Springer-Verlag: London

Sim SE, Singer J, Storey M-A (2001) Beg, borrow, or steal: using multidisciplinary approaches in empirical software engineering research, an ICSE 2000 workshop report. Empir Softw Eng 6(1):85–93 doi:10.1023/A:1009809824225

Singer J, Vinson NG (2002) Ethical issues in empirical studies of software engineering. IEEE Trans Softw Eng 28(12):1171–1180 doi:10.1109/TSE.2002.1158289

Sjøberg DIK, Dybå T, Anda BCD, Hannay J (2008) Building theories in software engineering. In: Shull F et al (ed) Guide to advanced empirical software engineering. Springer-Verlag, London

Sjøberg DIK, Hannay JE, Hansen O, Kampenes VB (2005) A survey of controlled experiments in software engineering. IEEE Trans Softw Eng 31(9):733–753 doi:10.1109/TSE.2005.97

Stake RE (1995) *The art of case study research*. Sage

Tichy WF (1998) Should computer scientists experiment more? Computer 31(5):32–40 doi:10.1109/2.675631

Tichy WF (2000) Hints for reviewing empirical work in software engineering. Empir Softw Eng 5(4):309–312 doi:10.1023/A:1009844119158

van Solingen R, Berghout E (1999) *The goal/question/metric method. A practical guide for quality improvement of software development*. McGraw-Hill

von Mayrhauser A, Vans AM (1996) Identification of dynamic comprehension processes during large scale maintenance. IEEE Trans Softw Eng 22(6):424–438 doi:10.1109/32.508315

Wallace C, Cook C, Summet J, Burnett M (2002) Human centric computing languages and environments. *Proceeding Symposia on Human Centric Computing Languages and Environments* pp 63–65

Wohlin C, Höst M (2001) Special section: controlled experiments in software engineering, guest editorial. Inf Softw Technol 43(15):921–924 doi:10.1016/S0950-5849(01)00200-2

Wohlin C, Höst M, Ohlsson MC, Regnell B, Runeson P, Wesslén A (2000) *Experimentation in software engineering—an introduction*. Kluwer

Wohlin C, Höst M, Henningsson K (2003) Empirical research methods in software engineering. In: Conradi R, Wang AI (eds) *Empirical Methods and Studies in Software Engineering—Experiences from ESERNET*, Springer

Yin RK (2003) Case study research. Design and methods, 3rd edn. London, Sage

Zelkowitz MV, Wallace RW (1998) Experimental models for validating technology. IEEE Comput 31(5): 23–31

Per Runeson is a Professor in Software Engineering at Lund University, Sweden, and is the academic leader of the Software Engineering Research Group. He received a Ph.D. from Lund University in 1998 and has five years of industrial experience as a consulting expert in software engineering. His research interests include methods to facilitate, measure and manage various aspects of software quality, especially testing and inspection methods as well as agile methods. The research has a strong empirical focus including cooperation with major companies, resulting in more than 90 papers published in international journals and conference proceedings. He is a member of the editorial board of the Journal of Empirical Software Engineering and several program committees.

Martin Höst is an Associate Professor in Software Engineering in the Software Engineering Research Group at the Department of Computer Science, Lund University, Sweden. He received an M.Sc. degree from Lund University in 1992 and a Ph.D. degree in Software Engineering from the same university in 1999. His main research interests include software process improvement, software quality, empirical software engineering, and computer simulation of software development processes. The research is mainly conducted through empirical methods such as case studies, controlled experiments, and surveys. He has published more than 40 papers in international journals and conference proceedings.