Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering

Marcus Ciolkowski[1], Oliver Laitenberger[2], Sira Vegas[3], and Stefan Biffl[4]

[1] Dept. of Computer Science, Universität Kaiserslautern,
D-67655 Kaiserslautern, Germany
ciolkows@informatik.uni-kl.de
[2] Droege & Comp. GmbH, Internationale Unternehmer-Beratung,
Praterinsel 3-4, 80538 München, Germany
Oliver_Laitenberger@droege.de
[3] Facultad de Informática. Universidad Politécnica de Madrid
Campus de Montegancedo, 28660, Boadilla del Monte, Madrid, Spain
svegas@fi.upm.es
[4] TU Wien, Inst. f. Softwaretechnik und Interaktive Systeme
Favoritenstr. 9/188, A-1040 Wien, Austria
Stefan.Biffl@tuwien.ac.at

Abstract. A survey is an empirical research strategy for the collection of information from heterogeneous sources. In this way, survey results often exhibit a high degree of external validity. It is complementary to other empirical research strategies such as controlled experiments, which usually have their strengths in the high internal validity of the findings. While there is a growing number of (quasi-)controlled experiments reported in the software engineering literature, few results of large scale surveys have been reported there. Hence, there is still a lack of knowledge on how to use surveys in a systematic manner for software engineering empirical research.

This chapter introduces a process for preparing, conducting, and analyzing a software engineering survey. The focus of the work is on questionnaire-based surveys rather than literature surveys. The survey process is driven by practical experiences from two large-scale efforts in the review and inspection area. There are two main results from this work. First, the process itself allows researchers in empirical software engineering to follow a systematic, disciplined approach. Second, the experiences from applying the process help avoid common pitfalls that endanger both the research process and its results.

We report on two (descriptive) surveys on software reviews that applied the survey process, and we present our experiences, as well as models for survey effort and duration factors derived from these experiences.

1 Introduction

Although empirical methods have a long history in manufacturing and in traditional areas of science, technology, and medicine, up to now, they have had little impact on software development practices. Fortunately, the situation is constantly improving. An increasing number of researchers (and practitioners) take advantage of empirical research strategies to validate their findings and their work.

R. Conradi and A.I. Wang (Eds.): ESERNET 2001-2003, LNCS 2765, pp. 104–128, 2003.

The research strategies primarily employed involve the design and conduct of (quasi-)controlled experiments and cases studies. The results of these studies help researchers gain an understanding of how a technique works and why the technique is useful. Practitioners, on the other hand, benefit from those studies because the results help them assess the leverage they can expect from a particular technique. This may influence their decision as to whether and how to adopt it in their projects.

Any empirical study is, according to Hays [1], a problem in economics. Each choice that a researcher makes for a specific study design has its price. For example, the greater the number of treatments, subjects, and hypotheses considered, the more costly a study is likely to be. This explains why in many cases specific treatment combinations can only be investigated in a single experiment or case study in a specific environment. As a consequence, the possibility for general statements that characterizes the state of the practice in the software industry is limited.

Surveys are an empirical research strategy that helps address this problem. Although this research method allows the creation of a more general picture of technology usage in the software industry, few have been conducted. Hence, there is little experience-based information regarding the details of surveys in a software engineering context. Moreover, the available information focuses mostly on the empirical results, and usually not on the lessons learned while planning, conducting, analyzing and packaging the survey and its results.

This chapter presents practical experiences in the design and conduct of surveys in empirical software engineering to illustrate the usefulness of surveys as an empirical method. It provides some detailed insights into some of the issues. The purpose of this chapter is to describe how surveys can be applied in the software engineering field, not to compare suveys to alternative empirical strategies. The reported experiences were made in a large-scale review survey that was first performed in Germany and then extended worldwide. The goal of this questionnaire-based survey was to characterize the modern review practices as being used in the software industry.

We chose reviews as topic for this survey for several reasons. First, although the software review and inspection methodology has been around for almost 30 years, it is unclear how many organizations practice them on a regular basis. Second, the area is quite well researched. As the basic research questions are well known, reviews and inspections were good topics to gain experiences with how to conduct large-scale surveys.

The rest of this chapter has been organized as follows: Section 2 discusses the state of the practice regarding surveys in empirical software engineering. Section 3 shows the methods and procedures that have been followed to conduct the review survey. Section 4 presents two survey examples and experiences with the survey process, while Section 5 summarizes lessons learned during the two surveys. Finally, Section 6 concludes.

2 State of the Practice

This section details the state of the practice in surveys and reviews. Surveys are a broad investigation where information is collected in a standardized form from a group of people or projects [2]. In section 2.1, we briefly present the state of the

practice in lessons learned in software engineering surveys. Section 2.2 details reviews and inspections.

2.1 Lessons Learned in Software Engineering Surveys

There are several types of surveys; Wohlin et al. list in chapter 2 three types of surveys: Descriptive, explanatory, and exploratory surveys. *Descriptive surveys* can enable assertions about some population; for example, determining the distribution of certain characteristics or attributes without explanation for the distribution. *Explanatory surveys* aim at making explanatory claims about the population; for example, explain why developers choose one technique over another. *Explorative surveys* are used as a pre-study to find out opportunities and risks for a more thorough empirical investigation.

Surveys offer a number of advantages [3]: Explanatory surveys can confirm an effect and typically allow the usage of standard statistical techniques. Descriptive (and explanatory) surveys can generalize empirical findings to many projects / organizations and are applicable to real-world projects for research in the large. Exploratory surveys are suitable for early exploratory analysis, can use existing experience, and can help to identify best/worst practices.

Several authors have pointed out the advantages of using surveys in Software Engineering research. Although it is common to find literature surveys in the Software Engineering field, not many efforts are reported in conducting questionnaire-based ones [4]. The main reason for this is that, while the source of information for literature surveys are usually books, technical reports and so on, questionnaire-based surveys deal with people. This, along with the process of conducting the survey itself, makes the success of a questionnaire-based survey more difficult. Whether or not a survey is questionnaire-based, there are two important matters that have to be decided:

- Select the parameters of interest it has to examine.
- Identify the sources of information that are needed (books and/or research papers for literature surveys, and people to be asked for questionnaire based ones).

Regarding the first matter, the set of parameters of interest is usually developed incrementally, which means that several trials are needed in order to get a satisfactory set of the information the survey should request. This is closely related to the second matter. Literature has a greater availability than people. This is probably one of the key issues when asking why questionnaire-based surveys are not common in Software Engineering. Executing Software Engineering surveys require a specific type of person to be available (usually experts) who are usually not readily available, or are expensive to include.

Even though few questionnaire-based surveys have been reported in the Software Engineering area (e.g., [5]; see [3]), the focus of a survey is to get information from the identified sources of information. Thus, surveys and reports on surveys focus on the results obtained during the survey, and not on lessons learned from it.

One of the goals of this paper is to contribute to mitigate the lack of reports on experiences in conducting questionnaire-based surveys in Software Engineering.

2.2 Reviews and Inspections in Software Engineering

In the past two decades, reviews and inspections have emerged as one of most effective quality assurance techniques in software engineering. The primary goal of a review or an inspection is the detection and removal of defects before the testing phase begins. In this way, reviews and inspections strongly contribute to improve the overall quality of software with the corollary budget and time benefits.

In this article, reviews and inspections denote approaches following a process in which qualified personnel analyze a software product for the purpose of detecting defects. The process involves the following six phases in some way or the other: Planning, Overview, Defect Detection, Defect Collection, Defect Correction, and Follow-up. The objective of the planning phase is to organize a particular review or inspection when the materials pass entry criteria, such as when source code successfully compiles without syntax errors. The overview phase consists of a first meeting in which the author explains the product to other participants. The main goal of the defect detection phase is to scrutinize a software artifact to elicit defects. The defects detected by each participant must be collected and documented. Furthermore, a decision must be made whether a defect is really a defect. These are the main objectives of the defect collection phase. Throughout the defect correction phase, the author reworks and resolves defects found. Finally the objective of the follow-up phase is to check whether the author has resolved all defects.

This definition of review and inspection used in this article is broader in scope than the one originally provided by Fagan, which focuses only on inspection technologies. However, after Fagan's seminal introduction of the generic notion of inspection to the software domain at IBM in the early 1970s [6], a large body of contributions in the form of new methodologies and/or incremental improvements has been proposed promising to leverage and amplify the benefits of early quality enhancing activities within software development and even maintenance projects. Many of these contributions have been empirically investigated in the form of (quasi-)controlled experiments. Hence, there are some business cases that demonstrate the pro's and con's of the various approaches.

However, despite the academic work it is often unclear what factors drive the adoption of review and inspection techniques in industry. There is also little understanding of which factors drive their adaptation in practice. Answering these questions is challenging because of the large number of factors that impact a specific review or inspection implementation. For example, when looking at the process described above there are already a large number of decisions and, thus, factors involved in the process design. Hence, any effort to explain the usage of a specific process implementation must look at a large number of factors. Without a comprehensive effort few valid statements can be produced.

Fortunately, there are some literature surveys in the review and inspection area that allow for a collection of possible factors [7, 8, 9, 10, 11, 12]. The literature surveys usually present frameworks to structure the large amount of work in this area. Kim et al. [7] present a framework for software development technical reviews including software inspection [6], Freedman and Weinberg's technical review [13, 14], and Yourdon's structured walkthrough [15]. They segment the framework according to aims and benefits of reviews, human elements, review process, review outputs, and other matters. Macdonald et al. [16] describe the scope of support for the currently available inspection process and review tools. Porter et al. [9] focus their attention on

the organizational attributes of the software inspection process, such as the team size or the number of sessions, to understand how these attributes influence the costs and benefits of software inspection. Wheeler et al. [11] discuss the software inspection process as a particular type of peer review process and elaborate the differences between software inspection, walkthroughs, and other peer review processes. Tjahjono [10] presents a framework for formal technical reviews (FTR) including objective, collaboration, roles, synchronicity, technique, and entry/exit-criteria as dimensions. Tjahjono's framework aims at determining the similarities and differences between the review process of different FTR methods, as well as identifying potential review success factors. Laitenberger and Debaud [12] structure the work on inspection technologies around five core dimensions. In addition, they present causal models that allow the explanation of inspection success.

The existing literature surveys were the basis for the elicitation of possible success factors for reviews and inspections. In this way, the literature surveys were the basis for the surveys described later.

3 Method

This section details the most important activities of a survey. The process is based on the framework of the quality improvement paradigm [17], as well as on an empirical study process described in [3]. The section concludes with a model on effort/cost and duration factors for a survey.

3.1 The Survey Process

The process itself consists of six steps: (1) Study definition – determining the goal of the survey; (2) Design – operationalizing the survey goals into a set of questions; (3) Implementation – operationalizing the design to make the survey executable; (4) Execution – the actual data collection and data processing; (5) Analysis – interpretation of the data; and (6) Packaging – reporting the survey results.

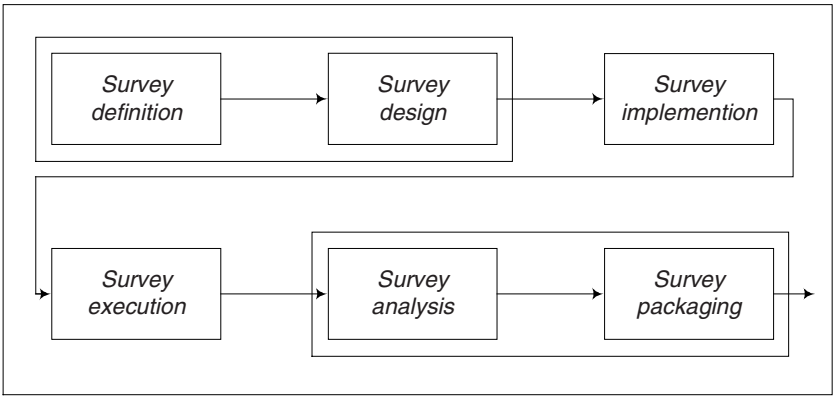

Fig. 1. Survey process steps and typical iterations in the process

These steps are often performed in an iterative fashion. Fig. 1 illustrates the process: the small boxes depict the process steps; larger boxes show typical iterations among activities. Iterations occur on three levels: (a) within a process step, (b) between the steps definition and design, analysis and packaging, (c) when learning from one survey to the next (e.g., pilot survey; replications; similar study designs).

3.2 Survey Definition

The primary objective of the survey definition step is to determine the goal of the survey to be performed; for example, stakeholders' perception of usage and effectiveness of processes before and after adopting a particular technology.

Goal definition is the most important step of a survey [3]: They often start with the awareness that we need further information about a specific topic that we might gain by asking people about it; e.g., that 'little is known about risk management' [18].

The next step is to review the literature to find available information about the specific topic. This helps determine the necessity for doing a survey. The review should result in an inventory of the research that has already been done on the topic; for example, it should produce an overview on risk management literature. Additionally, some exploratory 'in-depth' interviews with stakeholders can be done to get a clearer picture of the problems that should be addressed with the survey.

The review and the additional interviews should result in a statement that denotes the need for the survey and that clearly depicts the topic, population context, and scope. Sometimes research questions are formulated, too; for example, using GQM techniques [19]. Survey definition concerns also the feasibility and usefulness of the survey. For an explanatory survey, for example, it may be necessary to decide whether the necessary number of subjects can be contacted within the survey budget.

3.3 Survey Design

The design of a survey concerns the operationalization of the goal. It demonstrates how data can be collected and interpreted to give an answer to the research questions that are derived from the research goal [3]. The survey project plan records important decisions and activities. The following issues are the most important ones for survey design:

1. Definition of the target population and the survey sample.
2. Conceptual model of the objects and variables of the survey.
3. Approach for data collection.
4. Questionnaire design.
5. Approaches for data analysis.
6. Validity issues: Theoretic validity of the survey design and issues to be dealt with during survey implementation and execution.

Definition of the target population and the survey sample. From the survey goal the target population of the survey can be determined. A sample is a set of respondents selected from the population for the purpose to save time and money. Depending on the type of survey–exploratory, descriptive, or explanatory–the sample can be determined. This is an important step, as surveys are usually dealing with larger

populations than case studies or experiments; often, it is not possible to include the whole population into the sample.

Oppenheim [20] states that a representative sample should be drawn such that every member of the population has a statistically equal chance of being selected. The size of the sample depends on several factors: For example, the sampling error (the degree of precision of the sample taken) that can be tolerated, or the population size. Salant and Dillman [21] as well as Oppenheim [20] deal further on this topic.

The characterization of the target population is important when planning a representative sampling for this population. One approach that can be used is the snowballing technique: a few appropriate individuals are located and then asked for the names and addresses of others who fit the sampling requirements. A judgment sample can be taken to conduct a preliminary investigation [20]. Based on the characteristics of the target population, information on the sampling method, and information on the actual subjects of the survey, the bias of the sample can be determined; this bias in turn influences the level of validity of the survey.

The *conceptual model* describes the objects that are investigated, the variables and the expected relationships between them. A survey can focus on objects such as development organizations or projects; see, for example, [22, 23]. The variables that are defined during a survey should be strongly related to the research goal. The conceptual model follows from the definition of goals, questions and hypotheses, according to the GQM approach [17].

Approach for data collection. Depending on the type of data to be collected, the goal has to be expressed in a quantitative manner (including formal hypotheses on what to expect). Wohlin et al. state in chapter 2 that surveys have the ability to provide a large number of variables to evaluate, but that it is necessary to aim at obtaining the largest amount of understanding from the fewest number of variables, because this reduction also eases the analysis work. The two most common means for data collection are questionnaires (paper or electronic) and interviews (telephone or face-to-face) [24], sometimes complemented by literature surveys or project measurement, as appropriate. Questionnaires usually require much less time for the researcher than interviews, especially with mature questions. Interviews, on the other hand, help to reduce uncertainty with exploratory survey questions.

Question (or questionnaire) design is an important aspect of a survey as the clarity of the questions directly influences the quality of responses and thus of the survey data. The questions of the questionnaire should have a strong relation with the survey goal. Exploratory surveys often use open-ended questions that can capture unforeseen answer options, but are harder to analyze. Explanatory and descriptive surveys typically use pre-coded questions matured in pilot studies.

Approaches for data analysis depend on the type of survey and the collected data: For explanatory surveys statistical tests [2] have to be planned, while descriptive and exploratory surveys focus on basic descriptive statistics and aggregation of data for reporting. Typical analyses compare different populations of respondents; analyze associations and trends, or the consistency of scores. Furthermore, the validity of the collected data has to be checked with appropriate context and control questions.

Validity issues. As in every empirical study, surveys are also subject to validity threats. Validity considerations during survey design are a kind of forecast of possible problems and what the survey designer can do to avoid or at least detect them. In principle, similar threats as for experiments and case studies can occur. For an

extended description, the lists referenced in the respective chapters; for example, Wohlin et al. in chapter 2 and Robson et al. [2] can be consulted.

Internal validity in surveys concerns the level of controlling the variables, depending on the survey goal and the definition of variables in the study. Variables can be controlled by exclusion, by holding them constant, or by randomization. *External validity* in surveys focuses largely on the representativeness of the sample for the target population. Defining the population as well as taking the final sample determine the external validity. *Experimental validity* deals with the replication of the survey (i.e., 'do we get the same results when the survey is repeated?'). Therefore, it is important to restrict the subjectivity of answers to a minimum level by introducing context and control questions. *Construct validity* deals with the question 'do we measure what is intended to be measured'. A good start is to use similar designs from literature or research communities [3].

Typical survey problems in practice are that (a) they may rely on different projects/organizations keeping comparable data, (b) there is little control over variables, (c) they can at most confirm association but not causality, (d) they can be biased due to differences between respondents and non-respondents (i.e., non-response error), (e) questionnaire design may be tricky (ambiguity, validity, reliability).

3.4 Survey Implementation

The objective of the implementation step is to produce, collect, and prepare all the material that is required to conduct the survey according to the survey plan.

Material to be prepared includes means for data collection (e.g., data collection forms, data collection tools, on-line questionnaires, interview protocols). The effort to implement a complex questionnaire can be considerable, defects in the questionnaire can compromise the validity of the collected data, low usability may annoy prospective respondents and lower the response rate.

For explanatory and descriptive surveys a pilot survey is often performed in order to detect and correct any deficiencies in the prepared products or in the survey design. A goal is often to lower the effort and improve the ease of use for respondents, to improve the likely response rate, especially for large-scale surveys.

3.5 Survey Execution

The objective of the execution step is to run the survey according to the survey plan and collect the required data. The survey project manager has to check the actual execution with the plan and to conduct quality assurance activities that can be audited after the study for anomalies of execution that may impact the survey validity.

3.6 Survey Analysis

The objective of the analysis step is to analyze the collected data according to the data analysis methods selected during the survey definition in order to answer the questions derived from the survey goal. The analysis phase interprets the raw

measurement data. In principle several techniques can be used for this, ranging from common sense analysis (using standard descriptive techniques) to sophisticated statistical analysis techniques as appropriate based on the study design. Another issue is to check data validity based on information from design and execution.

3.7 Packaging

The objective of the packaging step is to report the survey and its result so that external parties are able to understand the results and their contexts. Typical target groups are management, quality management, researchers, and those who want to conduct a survey. Packaging survey results should be structured according to target group goals and interests. Examples include writing an executive summary for management, and detailed results for quality management and researchers in empirical software engineering. A standard format to present the survey results should pay attention to issues like: abstract or executive summary, problem statement, methods and procedures, acknowledgement on possible errors, findings, implications and (optional) appendices [21]; see also [25]. Further detail issues on research and analysis are particularly interesting for people who conduct surveys: "How to" steps, tips and tricks for survey conduct and improvement. Explicating the conceptual model as well as the refinement into questions should therefore by mandatory for each document where a survey is presented. However, this is not practice in survey studies, (see e.g. [22, 23]).

3.8 Effort/Cost and Duration Models

For someone planning to conduct a survey, the likely effort/cost and the minimal duration to conduct a survey variant are key information in the face of scare resources for research. Usually, the goal in planning is to optimize the collected information for a given budget or to answer a set of questions with the minimal budget. This subsection lists factors that are likely to have a strong influence on survey effort/cost and/or duration, and it can be used a checklist for the survey planner.

Effort/cost model: Estimate the staff hours needed for each step and role (at least survey planner and respondents) in the survey process. Use a likely cost per staff hour to derive the personnel cost for the survey. Add additional costs for material, tools, documents, travel, communication, and external services for the survey. There are some factors that influence the effort for survey steps (see also Fig. 1 to identify factors and relationships among survey goals, constraints, and solution approaches).

In Section 4.3.3, we detail factors that we found to be relevant in our surveys.

Duration model: Estimate the minimal duration for each step in the survey process based on the activities and size/complexity of results in each step. Further, forecast likely iterations of steps in the process and determine their influence on the overall process duration; for example, with a PERT model.

4 Two Survey Examples: ISERN and ViSEK Review Surveys

In this section, we describe two examples of surveys we conducted. Both studies had many things in common, although they had a slightly different focus. For both studies, the goal was to determine the state of the practice in software reviews and inspections. A sub-goal was to find significant context factors that influence how reviews are conducted.

In the following, we describe the process we followed for the surveys; that is, what the steps were we conducted, where we followed the theory (and where not), and what we learned during conduct.

Thereby, we first describe the ISERN survey [26], a survey we conducted within an international research network. The second survey was conducted within a German network of excellence (ViSEK, see http://*www.visek.de*). For the ViSEK survey [27], we point out what was different to the first survey. Following that, we present common results for both surveys. It is important to note that we cannot completely present all relevant issues in this section, as this would take too much space. Instead, we present some of the most important excerpts of our work.

4.1 ISERN Survey

The ISERN survey was planned and conducted by members of the *International Software Engineering Research Network* (ISERN; see http://www.iese.fhg.de/ISERN) as a common inter-cultural endeavor.

4.1.1 Survey Definition

We started the survey with the awareness that, although reviews and inspections are well known in practice and frequently examined in empirical research, we still do not know much about how they are applied in practice.

The survey goal was to describe the state of the practice for reviews and inspections. Doing that, we wanted to focus on the process of reviews; that is, how organizations that use reviews actually apply them. For this purpose, we defined "review" as a general term for all kinds of quality assurance activities that

- focus on finding defects,
- are conducted by developers (and not by an external group)
- are conducted as part of the development process, and
- are applied to all kinds of development products, such as requirements or code.

We based our survey, in particular the conceptual model, on existing literature surveys on inspections [7, 8, 9, 10, 11, 12].

A secondary goal of the survey was to find important context factors; for example, whether there is a difference between different types of organizations in the way they apply reviews.

4.1.2 Survey Design

Developing the design of a survey consists of several steps: defining the conceptual model, the population and sample, designing the questionnaire, and formulating validity issues. In the following, we detail each of these steps for the ISERN survey.

Conceptual Model
A conceptual model describes objects, variables and relations of interest for the survey. We identified three areas where we wanted to collect data: organizational context, high-level process, and detail-level process. In addition, we collected information about the person who filled in the questionnaire (the respondent), and about the testing process of the organization (see Table 1).

Table 1. Conceptual model of the ISERN survey

Context	Company Typical projects Typical products General attitude towards QA/maturity
High-level process	Reasons for/against reviews Entry/exit criteria for review process Metrics collection Experience/maturity of reviews Estimated effectiveness, effort Number of reviewers
Detail-level process	Review process steps and documents For each step: entry/exit, goals, effort Reading techniques used

The organizational context consists of several dimensions: questions to characterize the organization the respondent is working for, about typical projects and products, and about the general attitude towards quality assurance in the organization or business unit of the respondent. The purpose of the context questions is to classify organizations, to be able to identify factors that have an influence on how reviews are applied.

The high-level process construct describes the review process as a black box. It covers motivation for and against using reviews, entry/exit criteria for the review process, metric collection and estimation of review effectiveness. Further, it covers typical experience of reviewers as well as the typical number of reviewers in a review.

The detail-level construct describes the review process as a white box (or glass box). It covers process steps of reviews and documents reviewed, for each step: entry/exit criteria, goals for the step, and typical effort.

Definition of Target Population / Sampling
In our case, we defined our population as organizations (or business units) worldwide that develop software and that apply reviews.

As sample, we decided to use a convenience sampling. That is, we did not try to compile a representative list of organizations, as this would have required using a more or less complete global list of organizations and randomly selecting organizations from that list. To our knowledge, such a list does not exist. Instead, we decided to use a snowball system through ISERN and related research networks, such as ESERNET. The idea was that researchers in these communities should inform their industry contacts about the survey; these contacts should participate in the survey and, in turn, inform other interested parties. However, such a sample may not be

representative, because organizations that are in contact with research organizations may be significantly different from other organizations. Therefore, we decided to also use newsgroups and mailing lists to reach a larger audience, such as comp.software-engineering, or comp.software.testing.

However, the risks of such a convenience sample still exist. For example, it may be biased towards organizations that are interested in quality assurance. That is, we could not expect the results of our survey to be representative for the software industry in general. If at all, it could be representative for the part of the software industry that is interested in quality assurance. Fortunately, we were only interested in organizations that conduct reviews and thus are interested in quality assurance. That is, once the survey was conducted, we had to look at the context data to see how representative the organizations in our sample were for our population.

Questionnaire Design

When we started to develop our questionnaire, we used other questionnaires as a starting point. One was a questionnaire developed by Håkan Petersson, Thomas Thelin, and Katarina Kylvåg from Lund University in Sweden for characterizing inspections in Swedish companies. Further, we used a questionnaire that was developed and applied for improvement of technical reviews at Lucent, Germany [28].

Corresponding to the conceptual model, the questionnaire consisted of three main parts: The first part served to characterize the respondent and his or her organization. The second part asked questions on the high-level process; the third part on the detail-level process. In an additional fourth part, we asked questions on testing and feedback on the survey.

In total, we had about 130 questions in the final questionnaire. Some required just filling in a comment or text, and some were simple multiple-choice questions However, some questions were quite complex and required filling in a matrix, such as rate how relevant potential review goals were for the respondent's context. Table 2 presents an example for such a complex question, taken from the questionnaire design document (i.e., it is not an example of the final questionnaire layout).

In addition to the complexity of the questions, the questionnaire itself was complex, because we had many conditional questions. That is, some questions were only relevant when a previous answer had a specific value. For example, if someone does not collect metrics, it makes no sense to ask how the collected data are analyzed. That is, in some cases, questions can be skipped. In other cases, the question's layout itself changed depending on previous answers. For example, in one question we asked whether a document was produced during development, and in the next question we asked which of those documents were reviewed. Thus, in the second question, we only displayed documents that corresponded to a positive answer in the first question. Altogether, we had 32 conditional jumps in the questionnaire. Further, there were about 10 questions that changed their layout depending on previous answers.

Although the questionnaire was quite complex, the respondents did not have to deal with that complexity, as they just had to answer questions and press the "next question"-button; the tool did the rest. The effort of filling in the questionnaire was still quite high; respondents needed about one hour to answer it, no matter whether we used a paper based (during pilot studies) or web-based version.

Table 2. Example of a more complex matrix question in the final questionnaire

Please rate the importance of the following goals for reviews in your context:	Crucial ⇓		Desir -able		Irrelevant ⇓
Quality improvement (i.e., find and correct defects)	☐	☐	☐	☐	☐
Enforce the defined standards	☐	☐	☐	☐	☐
…	☐	☐	☐	☐	☐

Validity Considerations

Validity considerations during design are a kind of forecast of possible problems and what we do to avoid them. That is, we compile a plan of possible threats, some of which we can control with our design (e.g., by holding some variables constant), and state how we react to other threats we cannot control when they arise. Here, we present some important validity considerations for our survey.

Internal validity: Internal validity is about whether the design of the study allows us to draw valid conclusions. For example, if the level of control on the study is too low, it is impossible to draw conclusions. For a survey, the control is usually quite low; for example, it is usually impossible to know whether the respondents answer truthfully, or whether other effects bias the results (e.g., history effects—external events that influence someone's view and attitude towards a question). One measure we took to minimize possible threats to internal validity was to use a short time period during which the questionnaire could be answered. Therefore, the risk of external events influencing the result of the study was low. For the ISERN survey, this period was about 8 weeks. Other factors influencing internal validity are, for example:

- *Non-response error*: A significant number of people in the sample do not respond to the questionnaire and are different from those who do. This is relevant in our case, as we were not even able to find out who did not respond to the survey. Even then, due to the design and contents of the questionnaire, there is a tendency that the survey attracted more respondents who have already heard about or used reviews than those who have not, and they may be significantly different from the rest. Thus, it is impossible to draw any conclusions about the software industry in general: The true population of respondents that do not use reviews is probably much larger than the results of our survey indicate. However, this is not a problem, because we were not interested in drawing conclusions about the software industry in general.
- *Measurement error:* A respondent's answer is inaccurate, imprecise, or cannot be compared in any useful way to other respondent's answers. We tried to minimize this error by carefully designing the questionnaire and by making extensive use of pilot studies. In addition, we had the advantage that our conceptual model could be based on a "hard" process description (in contrast to "soft" models like in attitude measurement)

External validity: Another crucial consideration is *external validity*; that is, how representative the results are. For our survey, this is problematic, because we used convenience sampling to draw respondents from the population. That is, we had to carefully check the context data to see whether the participating organizations were representative for that part of the software industry that conducts reviews.

Experimental validity: Further, we have to consider *experimental validity*; that is, whether a repetition will achieve the same results. Experimental validity is highly tied to external validity: If the results are not representative for the population, a different sample might produce completely different results. In our case, from the data we collected, we saw that the external validity was quite high. However, the tricky question here is, whether the same respondents would today still give the same answers as when they originally filled out the questionnaire.

Construct validity: Last but not least, we had to consider *construct validity*; that is, whether we were measuring the right things. Construct validity is especially difficult when dealing with "soft" issues such as attitude measurement or ease of use/usability of products. In that case, it is necessary to very carefully check the construct validity with a series of studies (see, for example, [29]). In our case, we had the advantage that we could refer to a "hard" conceptual model, based upon published descriptions of the review process. The conceptual model could then directly be refined into questions and questionnaire items. Thus, only ambiguity of the questions is a problem, and we conducted several pilot studies to reduce that threat.

4.1.3 Survey Implementation

We had to consider two choices. One was to use questionnaires, the other to conduct interviews. As we wanted to do an descriptive study, interviews would have been an appropriate means, as they allow to clarify open-ended questions, allowing the researcher to gain a better understanding of, for example, how a process is conducted. However, we wanted to cover a large sample; therefore, we decided to go for a questionnaire (see section 4.1.2).

We believe that we avoided typical problems of such studies for several reasons. For one, the area where we wanted to conduct our study (review process) is quite well known, which means that we did not have to explore how the process itself looks like, only on how it is applied in practice. Therefore, we were able to ask closed questions. To avoid problems with the wording of questions and ambiguities, we conducted several pilot studies.

Pilot Studies

As already mentioned, the process for implementing a survey is usually not conducted sequentially. Usually, surveys are designed iteratively, doing pilots as kind of prototyping for the final questionnaire.

First, we conducted an internal pilot by testing and reviewing the questionnaire among the development team. We also conducted an external pilot at an ESERNET workshop on inspections, where we asked about 20 participants from industry and academia to fill in the questionnaire and give us feedback. Further, we conducted in-depth interviews at a large company, and we had an extensive review by experts within ISERN.

However, we still made mistakes during that process. The gravest of them was that, because of time problems, we did not fully analyze data from the ESERNET pilot. Usually, it is recommended to analyze any data from pilot studies, or to invent data if that is not possible. The reasoning behind this is that analyzing data can show flaws and vague formulations in the analysis plan and questions. It turned out that this would have helped us to avoid problems later on.

Questionnaire Tool

The basic decision one has to make is how the survey will be distributed to the respondents; that is, which tool to use for conducting the survey. For several reasons, we decided to use a web-based questionnaire. The main reason was that web-based questionnaires are easy to access, and that everyone in our population should have access to the internet as well as be familiar with web-based questionnaires.

However, the decision on how precisely to implement a web-based questionnaire remains. Several COTS tools for implementing web-based questionnaires exist. Usually, however, this choice is quite expensive, as it is necessary to pay a license fee.

There are several tools available that support the (web-based) execution of surveys, for example WebSurveyor, SurveySaid EE, JMP, GlobalPark, ConfirmIt and others (see also [30]). Instead of using a COTS product, it is also possible to implement an own, customized tool. The cost for that is associated with the effort of implementing it. For companies who have to pay developers, having a highly qualified developer working at implementing a questionnaire may be more expensive than paying for a COTS products. Additionally, the quality of commercial systems can be expected to be higher than that of an own implementation.

In our case, we decided at first to implement our own tool. The main reason was that we were not able to raise funding for using a COTS tool at the beginning. We had a student building an online questionnaire (using PHP), and working prototypes of our tool featured a graphical layout of the questionnaire in a sidebar so that respondents could easily jump back to earlier questions. In addition, we implemented a feature that allowed respondents to temporarily stop answering the questionnaire and continue later at the questions where they had stopped before. However, it turned out that the questionnaire was too complex to be implemented in that way within a reasonable time. Additionally, as we followed an incremental process, we mixed questionnaire design and implementation at that stage. Thus, there were several changes to the questionnaire itself while it was being implemented, in addition to errors that had to be removed from the implementation. These changes turned out to be unmanageable given the student's implementation.

In the end, we decided to give up the tool and instead to use a COTS product, by GlobalPark, for implementing the questionnaire. We had this option now, because we had been able to raise funding in the meantime through the ViSEK survey. In fact, after our own implementation failed, we first translated the questionnaire into German, adapted the design to ViSEK needs, and executed the ViSEK survey. Afterwards, we fed the questionnaire back into the ISERN survey, again slightly adapted the design, and thus extended the survey worldwide.

Implementing our complicated questionnaire with the COTS tool was a much simpler matter than before; the tool was able to handle even conditional questions well. Thus, implementing the questionnaire was a matter of a few days only. Because the tool relieved us of technical implementation details, we were able to focus on the questionnaire design, structure, and content.

While the COTS tool did not have all features that our own tool had had, it offered some support during conducting the survey that turned out to be useful. For example, it was able to give feedback on the survey progress, such as how many respondents had answered, at which questions respondents had stopped answering the questionnaire, or how much time they needed to answer the questionnaire. Additionally, it was possible to perform a simple analysis of the respondents'

answers. Further, the tool stored every answer the respondents gave immediately, so questionnaires were stored, even if they were only partially answered. Respondents were able to go back to earlier questions and change their answer; however, they had to use the browser's "back"-button for that. If a questionnaire has many questions, this approach is usually not useful.

Additional advantages in using a COTS tool are that, usually, the company that sells the tool also offers to give feedback on questionnaire design, which can help avoid problems.

4.1.4 Survey Execution

The execution phase was quite simple, at least for the researchers. We placed a call for participation into several mailing lists and newsgroups. All data collection and processing during that phase was done via the tool. All we had to do was to supervise the progress of the execution using GlobalPark's built-in features.

During this phase, the built-in analysis allowed us to react to a pattern, where people stopped immediately after answering the first question. When we looked more in detail, it turned out that the first question asked the respondent's name. Although we had promised anonymity before, we assumed that having to give their name was what stopped people from answering, so we removed that question.

For the ISERN survey, we received 105 responses from companies worldwide. We were able to observe that each new call for participation in a newsgroup was followed by a "peak" of respondents that wore quickly off after a few days. Thus, it seems reasonable for the future to use that pattern to place calls for participation.

We also made an interesting observation: Although answering the questionnaire took about one hour, as we had expected, only around 30% of the people who started responding completed the questionnaire. That is, 325 people started to answer the questionnaire, and only one third completed it. However, if we look closer at the drop-off pattern, we can see that most of those who did not complete the questionnaire did so after the first two pages, where we introduced the purpose and motivation of our study. That is, those people were not interested at participating. After that, only few people stopped answering; that is, if they started to answer questions, they would be likely to finish the questionnaire. That was surprising for us, as we had expected that respondents would gradually stop answering after some time, after getting annoyed or bored.

4.2 ViSEK Survey

The ViSEK survey had the same goals as the ISERN survey. Thus, both surveys were able to profit from each other. However, there were some slight differences between these surveys that we present in this section.

First, the *survey definition* was slightly different in ViSEK. ViSEK is a German project, funded by the German government, so the focus of this survey was mainly the software industry in Germany, and there was stronger additional interest in testing than in the ISERN survey. As a result, the questionnaire had to be translated into German. Usually, such a translation can cause many problems, for example in attitude measurement, as the construct validity is endangered [29].

The *survey design* used the same conceptual model as the ISERN survey, with a slightly higher interest in testing.

The *sampling procedure* was different for the ViSEK survey. Instead of using newsgroups, we used our own industrial contacts—that is, of the Fraunhofer Institute for Experimental Software Engineering—and of industrial contacts of the ViSEK project. Additionally, we used contacts from local quality improvement conferences. In total 865 people were identified to participate in the survey. These were invited to participate.

The *survey implementation* step was the same as in the ISERN survey, except that we immediately tried to use a COTS tool, as the own implementation had failed for the ISERN survey.

During the *survey execution* step, the only difference between the ISERN and ViSEK surveys was that we were able to use a feature of our tool, to send personalized e-mails to contact persons instead of anonymous calls in newsgroups and send up to two reminders to those who had not answered yet. It turned out that this approach resulted in a high response rate of ca. 14%. In our opinion, using personalized e-mails and sending reminders caused this high response rate.

Like in the ISERN survey, we were able to observe a "peak" of respondents each time the tool sent a reminder. Also like in the ISERN survey, about one third of the respondents did not complete the questionnaire; and again, most of them stopped after the introductory pages.

4.3 Considerations for Both Surveys

In this section, we combine our results for the survey steps analysis and packaging. Further, we present our findings concerning effort models from both studies.

4.3.1 Survey Analysis

The goal of this step is to analyze the collected data with respect to the goals of the survey. In the following, we first present some of the results from both surveys (i.e., for ISERN and ViSEK surveys together) and then briefly discuss some validity issues.

Results

Here, we briefly present some excerpts of results from the studies. As the focus of this chapter is to present our experiences, not results of the study, we will only present some results about the high-level and detail-level process constructs. Interested readers should refer to [27].

Regarding the high-level review process, we were most interested in what motivated someone to use reviews, or what stopped them from using reviews. Further, we wanted to know to what extent the review process was conducted; whether it would be applied to all development stages, and what the entry/exit criteria look like.

Considering motivation for reviews, we asked, among others, one question on goals that are associated with reviews. 73% rated quality improvement as a very important goal of reviews, 52% stated that the evaluation of the project status is most important, and 54% see reviews as a means to enforce standards. As reasons against reviews, most respondents mentioned that time pressure prevented the use of reviews (75%), that reviews are too expensive (56%), and that they do not know how to introduce reviews (50%).

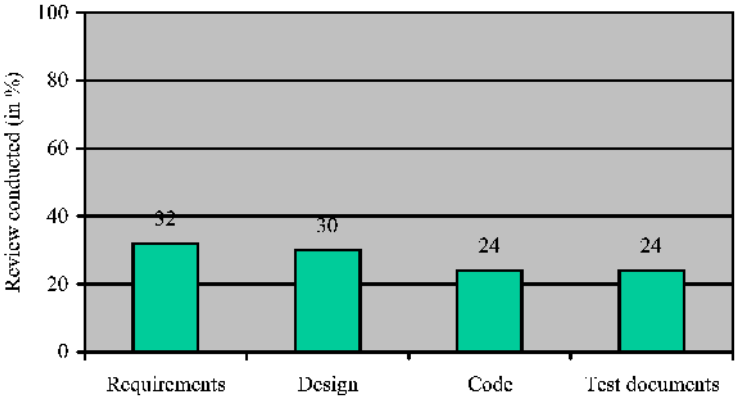

Fig. 2. Review usage: This figure shows the percentage of respondents who usually conduct a review in a specific development phase

Regarding entry/exit criteria of the review process, it seems that almost one third of the participants do not use formal criteria at all to determine when to start a review. For example, concerning how they identified whether a document should be reviewed, 32% stated that they used no criteria at all to select a document for review, and 44% review a document because it is required in the project plan.

Reviews can be conducted at all development stages, from requirements to code, or to check test documents. However, it seems that more than two thirds of the participants do not perform reviews on a regular basis for any development stage (see Fig. 2). Thus, together with other results, we have the impression that reviews are not conducted systematically in practice.

Looking at the detail-level review process, we were, among other things, interested in which review process steps were conducted on a regular basis, and what reading techniques are used. Our data show that many review steps are not conducted on a regular basis. For example, only 20% of the respondents stated that they usually conduct a planning and overview step, and only 40% stated that they usually have a defect detection or defect collection phase, and around 35% usually have a follow-up meeting. Again, this confirms our impression that reviews are not conducted in a systematical way.

Regarding reading techniques used during review, an area where there has been significant research over the last years, we were interested in what reading techniques are applied in practice. Around 55% state they use checklists, and 35% state they use no specific reading technique at all. Only about 10% use more advanced reading techniques, such as scenario-based techniques [31]. However, as many of the respondents have been contacted through our research network, where advanced reading techniques have been promoted for several years, the number of 10% may be a bit higher for our sample than in the rest of the population.

We found a tendency in our data that the use of reviews increased with having certification, with the size of the organization (i.e., large organizations were more likely to apply reviews), with the number of years in business, and with the project size. We conducted chi-square tests to confirm these findings; these tests were significant at the 5% level.

So far, the results seem to indicate that, although reviews are quite often applied in practice, they are not applied in a systematical manner. Moreover, our findings lead us to the assumption that reviews are often introduced because they are required for a certain certification, not because management believes in their usefulness.

Validity Discussion
During survey design, validity discussion is a "forecast" of possible threats and reaction to them. Later, the validity discussion refers to the concrete study situation; that is, we check which threats are really present in the data. In particular, we have to find out how representative our sample is. In the following, we present a small excerpt of available data.

It turned out that the respondents came from quite mixed environments. Interestingly, these demographic results were almost similar for both surveys, although the ISERN survey had participants from all over the world, while the ViSEK participants were from Germany only. Organizations of all sizes participated, from very small (1 – 5 employees) to very large (more than 10000 employees). All in all, about 47% of the respondents were from small and medium enterprises of less than 500 employees.

Further, they were from all kinds of industrial branches, from embedded systems (27%), telecommunication (10%), information systems (30%), web or mobile applications (14%), and others (19%). Concerning important quality aspects, 77% rated security as important, another 77% reliability. 56% of the respondents claim it is crucial that they meet real-time requirements. Concerning safety, the risk of financial loss is major concern for 66% of the respondents, and the risk of life is a concern for 33% of the respondents.

Based on these data, it is hard to conclude whether the sample is representative, as only few studies are available that describe the population (e.g., [32]). However, we believe that there may be a slight bias towards large enterprises and towards embedded/telecommunication organizations compared to the typical software industry. Our population was not the complete software industry, but only the part of it that conducts reviews and is thus more interested in quality assurance. Based on our experience, we believe that our sample may be representative for that part of the software industry. At least, our sample covers a wide range of different types of organizations.

4.3.2 Packaging

The objective of the packaging step is to report the survey and its result so that external parties are able to understand the results and their contexts. So far, we have packaged only parts of the survey results. We wrote several reports on the ViSEK survey ([27, 33]). The first is a technical report on the survey result, while the second one is a short newsletter, an executive summary for interested parties. The first report is in English, the second one was written in German to acknowledge the fact that the survey was conducted within a German project.

Further, this bookchapter is part of the packaging for our surveys, where we document the experiences we made during planning and execution of the surveys.

The ISERN survey has so far not been completely analyzed; first results have been presented at the ISERN workshop 2002 (see http://www.iese.fhg.de/ISERN/ for details).

In addition, we have planned several other publications, among them a German quality conference (SQM). We plan to use conferences to discuss our findings within the community of practitioners to see how well our conclusions meet the practice.

4.3.3 Effort/Cost and Duration Models

Here, we present the experiences we made concerning the models presented in Section 3.8. We provide a short listing and discussion of factors that had important influence on survey cost and duration. We distinguish between "linear" and "exponential" influence. We do not want to imply a mathematical relationship with these expressions; rather, we want to convey a feeling to what extent an increase in a factor influences the effort or time for that step. Please refer to Table 3 for an overview of costs for the ISERN survey. In the following, we present, for each step, the relevant factors and how they influenced the effort for that step.

Table 3. Effort per process step, and role in ISERN and ViSEK surveys

Process step / Role	Experimenter	Subject
Definition	80 to 100 staff hours	-
Design	500 to 600 staff hours	-
Implementation	Own tool: 150 to 200 staff hours COTS: 40 staff hours	-
Execution	25 staff hours	60 to 90 minutes
Analysis	300 to 400 staff hours	-
Packaging	300 to 400 staff hours	-
Sum	*Own tool: 1.350 to 1.725 staff hours* *COTS: 1.200 to 1.550 staff hours*	-

Definition: Time/effort increases if: (a) the research question(s) is/are vague or are numerous, or (b) many stakeholders are involved. In the beginning, we had to put much effort into a clear definition of the research question. The number of research questions also plays a major role; it is important to restrict them to few. Another important influence factor is the number of stakeholders involved. The more stakeholders are involved, the more potential interest conflicts about research questions will arise, which increase the amount of communication involved. Thus, the number of stakeholders is a factor that may contribute exponentially to the total effort.

Design: Time/effort increases basically with (a) the complexity of the conceptual model; (b) the number of contacts based on validity goals and anticipated response rate; and (c) the length of the questionnaire/interview (i.e., number and complexity of questions). The main influence factor for design is the complexity of the conceptual model, which is, in turn, influenced by the number and complexity of research questions. In our experience, the research questions contribute exponentially to the conceptual model; that is, even one additional question can increase the complexity of the conceptual model significantly. In our case, the conceptual model was quite complex, which resulted in a large and complex questionnaire (many conditional questions). This increased the design effort significantly. Therefore, the complexity of the conceptual model has an exponential influence on design effort.

Implementation: Time/effort increases with the complexity of the questionnaire; that is, with the number and complexity of the questions and dependencies (i.e., some questions change or are not relevant dependent on answers given to earlier questions). Further, the tool used plays a major role for implementing the questionnaire structure and phrasing the questions, and for usability testing. When building our own, customized tool, the complexity of the questionnaire had an exponential influence on implementation effort. A COTS tool can significantly lower the effort, and can reduce the influence of the questionnaire's complexity to a linear one.

Execution: Time/effort increases with (a) the number of respondents depending on (b) the data collection approach – questionnaire and/or interviews. During this step, the main influence factor was the number of respondents. As the cost for reaching the respondents was almost zero (in contrast to a traditional sending of paper questionnaires via mail), the only cost from our side was to track the survey status with tool help, and to place calls for participation. Thus, the cost for that step is the cost of having respondents fill in the questionnaire; this amounts to a linear influence.

Analysis: Time/effort increases with the complexity of the conceptual model. If (a) many questions have to be analyzed, and (b) many relations between different variables have to be considered, the analysis is much more complex and time consuming than for a simple model. Depending on the complexity of research questions, this influence may be linear.

Packaging: Time/effort increases with the (a) complexity of the conceptual model; (b) the audiences you want to address: The more diverse audiences you want to address, the more effort you have to put into writing reports. Time/effort can be decreased with (c) a good infrastructure that supports, for example, storing empirical data in a structured way, which makes easier to reuse previously packaged results. The conceptual model has a linear influence here; more important is the number and kind of reports you want to write.

5 Lessons Learned

The examples presented in the previous section demonstrated the usefulness of the survey process. Each of these steps was important for the success of the survey. Overall, the results were helpful to further investigate review and inspection technology. While following the process, there were some lessons learned that may be beneficial for practitioners.

First of all, it is important to follow the recommended survey process. It may be necessary to apply the steps in an iterative manner. The main element of the design phase is the construction of the questionnaire. Typically, after an introduction, which discloses the sponsorship of the survey, the questionnaire begins with non-threatening questions that stimulate interest. The first question should be clearly related to the announced purposes of the survey (not a background question, for instance). Some recommend the second question be open-ended, to allow the respondent to "get into" the subject. Non-threatening background information questions (e.g., demographic information) should be posed early so that these controls will be available if the respondent fatigues and does not answer the later questions. The survey then proceeds to attitude questions, often sequencing from general and less threatening items toward more specific and more sensitive items. Sensitive background questions are usually

put at the end. However, the more toward the end of the survey a question is, the lower its response rate is apt to be. For this reason, balancing the foregoing considerations, it is desirable to administer a survey with different question orders to lessen the order/response bias. Modern survey tools usually offer this functionality.

Modern Internet surveys are proliferating. Web surveys clearly work better with software engineering topics, since most respondents are connected to the Internet anyway and it is more attractive, or easier, for them to participate. Yet, this requires more effort regarding the graphical design of the survey. It may also require a higher budget. The question is whether to buy a professional tool. While this is not a yes/no question, the experience of the ViSEK/ISERN-Survey is that it can be time-consuming to build an own, customized, tool. This may be possible for a small number of questions; for example, for less than ten. However, if it is larger and there are conditional responses, one should think about support of a professional tool vendor. In addition, the tool vendor often has experience in questionnaire design and, thus, offers support on questionnaire design. Moreover, these tools often include additional functionality and analysis support; for example, for monitoring at which question participants stop answering the questionnaire. In the context of the ViSEK/ISERN-questionnaire, we used the tool Globalpark. The high response rate of 14% for the ViSEK-Survey indicates that the response rate for online questionnaires can be much higher than for paper-based questionnaires (usually between 1% and 4%). This may be due to the built-in functionality of the tool to remind those that did not complete the questionnaire, and the low effort to answer online questionnaires (you don't have to send mail).

Once the data are collected, they have to be analyzed and the results need to be packaged to develop an interpretation. This effort is often neglected in the design and conduct of the survey. The approach of choice is an iterative approach to analysis: Start with simple analysis questions (e.g., about the demographic information) before going into the details. The documentation adds to the overall packaging effort that needs to be considered in the models.

In both the ViSEK and the ISERN survey, the number of the questions, the clarity of the questions, especially sets of conditional questions, and the tool implementation were major drivers for effort and duration of the survey.

6 Conclusions

Surveys are empirical procedures in qualitative and quantitative research in which researchers administer a questionnaire to a sample or to the entire population in order to describe attitudes, opinions, behaviors, or characteristics of the population. From the results of this survey, the researcher makes claims about trends in the population. Although surveys belong to the set of standard procedures in other disciplines, little has been reported about using this approach to gather data about techniques, methods, and tools in software engineering. In this chapter, we presented practical experiences in the design and conduct of surveys in a software engineering context.

The challenge of a survey derives from the complexity of the investigated topic. In this situation a well-defined survey process is a beacon that can be followed. The process defines the roles, the activities, and the results of each phase. Hence, the complexity can be handled and confusion be avoided. The process is particularly

important when researchers from different sites are involved in the survey work. In this situation, the survey process synchronizes the work among participants. The ViSEK/ISERN-survey presented in this paper is an example that illustrates this experience. Finally, deviations from any process are a fact of life. However, to manage a survey effort one needs to know when and why deviations may occur to ensure the successful completion of the whole work.

While using the process in the context of the two inspection surveys, it became obvious that a scientific survey is not a trivial undertaking. It requires careful research and planning, is labor intensive, and can take weeks to implement and analyze. It is not just the development of a questionnaire and the subsequent collection of data. Depending on the complexity of the topic, planning can easily consume up to two months of work; sometimes even more, especially if the questionnaire cannot be taken "off the shelf" but needs to be developed from scratch. Even worse, since there is a clear lack of models in software engineering, one often needs to invest a considerable amount of time to develop the conceptual model. Only the availability of the model ensures that the survey results can be interpreted in an adequate manner.

The process described in this paper has demonstrated its usefulness in the context of two large empirical studies about inspection technologies. This effort itself can be regarded as a kind of empirical study to validate the process. Although some of the experiences presented may sound trivial they are sometimes challenging to fulfill. At least, the work presented in this chapter increases the awareness for practitioners and researchers of the crucial elements involved while planning, designing, conducting and analyzing a survey.

Acknowledgements. Parts of this work have been funded by the ViSEK project, which is in turn funded by the Geman Federal Ministry of Education and Research (BMBF).

The surveys described here have been initiated, planned and conducted by members of the International Software Engineering Research Network (ISERN). We would like to thank all members who have taken part in this endeavour.

Last but not least, we would like to thank the participants of the pilot studies and early interviews for their valuable input, and the participants of the survey for their time and interest.

References

[1] Hays, W. L., Statistics for the social sciences, London: Holt, Rinehart and Winston, 1977.
[2] Robson, C., Real World Research: A Resource for Social Scientists and Practitioners-Researchers, Blackwell, 1993.
[3] Freimut, B., Punter, T., Biffl, S. and Ciolkowski, M., "State-of-the-Art in Empirical Studies," ViSEK Technical Report 007/E, 2002.
[4] Ticehurst, G. and Veal, A., Business Research Methods: A Managerial Approach, Australia: Addison Wesley Longman, 1999.
[5] Dybå, T., "Improvisation in small software organizations," IEEE Software, 17(5), pp. 82–87, September 2000.
[6] Fagan, M. E., "Design and Code Inspections to Reduce Errors in Program Development," IBM Systems Journal, 15(3), pp. 182–211, 1976.

[7] Kim, L. P. W., Sauer, C. and Jeffery, R., "A Framework for Software Development Technical Reviews," Software Quality and Productivity: Theory, Practice, Education and Training, 1995.
[8] Macdonald, F., Miller, J., Brooks, A., Roper, M. and Wood, M., "Applying Inspection to Object-Oriented Software," Software Testing, 6, pp. 61–82, 1996.
[9] Porter, A. A., Votta, L. G. and Basili, V. R., "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment," IEEE Transactions on Software Engineering, 21(6), pp. 563–575, 1995.
[10] Tjahjono, D., Exploring the effectiveness of formal technical review factor with CSRS, a collaborative software review system, PhD thesis, Department of Information and Computer Science, 1996.
[11] Wheeler, D. A., Brykczynski, B. J. and Meeson, R. N., "Software Peer Reviews," in R. H. Thayer, ed., Software Engineering Project Management, , IEEE Computer Society, 1997.
[12] Laitenberger, O. and DeBaud, J., "An Encompassing Life-Cycle Centric Survey of Software Inspection," Journal of Systems and Software, 50 (1), 2000.
[13] Freedman, D. P. and Weinberg, G. M., Handbook of Walkthroughs, Inspections, and Technical Reviews, New York: Dorset House Publishing, 1990.
[14] Weinberg, G. M. and Freedman, D. P., "Reviews, Walkthroughs, and Inspections," IEEE Transactions on Software Engineering, 12 (1), pp. 68–72, 1984.
[15] Yourdon, E., Structured Walkthroughs, N.Y.: Prentice Hall, 4th edition, 1989.
[16] Macdonald, F. and Miller, J., Modelling Software Inspection Methods for the Application of Tool Support, Technical Report RR–95–196 [EFoCS–16–95], University of Strathclyde, UK, 1995.
[17] Basili, V. R., Caldiera, G. and Rombach, H. D., "Experience Factory," in J. J. Marciniak, ed., Encyclopedia of Software Engineering , John Wiley & Sons, pp. 469–476, 1994.
[18] Ropponen, J. and Lyytinen, K., "Components of software development risk: how to address them? A project manager survey",IEEE Transactions on Software Engineering", 26(6), 2000.
[19] Basili, V. R., Caldiera, G. and Rombach, H. D., "Measurement," in J. J. Marciniak, ed., Encyclopedia of Software Engineering, John Wiley & Sons, 1994.
[20] Oppenheim, A., Questionnaire design, interviewing and attitude measurement, London: Pinter, 1992.
[21] Salant, P. and Dillman, D. A., How to conduct your own survey?, New York: John Wiley and Sons, 1994.
[22] Paulk, M. C., Goldenson, D. and White, D. M., The 1999 Survey of High Maturity Organizations, Technical Report CMU/SEI–2000–SR–002, SEI, 2000.
[23] European Software Institute, "1995/1996 Software excellence study. Summary of results", 1996.
[24] Babbie, E., Survey Research Methods, Wadsworth, 1990.
[25] Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K. and Rosenberg, J., "Preliminary guidelines for empirical research in software engineering," IEEE Transactions on Software Engineering, 28(8), pp. 721–734, 2002.
[26] Ciolkowski, M., Shull, F. and Biffl, S., "A Family of Experiments to Investigate the Influence of Context on the Effect of Inspection Techniques," Empirical Assessment of Software Engineering (EASE), Keele, UK, 2002.
[27] Laitenberger, O., Vegas, S. and Ciolkowski, M., The State of the Practice of Review and Inspection Technologies in Germany, Technical Report ViSEK/010/E, ViSEK, 2002.
[28] Laitenberger, O., Leszak, M., Stoll, D. and Emam, K. E., "Evaluating a Model of Review Success Factors in an Industrial Setting," Proceedings of the International Symposium on Software Metrics, 1999.
[29] Davis, F., "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," MIS Quarterly, pp. 319–340, September 1989.
[30] Assmann, D. and Kempkens, R., Tools for Measurement Support, Technical Report No. 97.00/E, Fraunhofer Institute for Experimental Software Engineering, December 2000.

[31] Basili, V. R., Shull, F. and Lanubile, F., "Building Knowledge through Families of Experiments," Transactions on Software Engineering, 25 (4), pp. 456–473, 1999.
[32] GfK; Fraunhofer IESE; Fraunhofer ISI, Analyse und Evaluation der Softwareentwicklung in Deutschland, Eine Studie für das Bundesministerium für Bildung und Forschung (in German), 2000.
[33] Ciolkowski, M. and Kalmar, R., Software-Reviews sind als Instrument zur Qualitätssicherung in der Industrie anerkannt, ViSEK Newsletter, 2002.