Quality of Design, Analysis and Reporting of Software Engineering Experiments: A Systematic Review

Vigdis By Kampenes

Thesis submitted for the degree of Ph.D.

Department of Informatics
Faculty of Mathematics and Natural Sciences
University of Oslo

September 2007

Series of dissertations submitted to the
Faculty of Mathematics and Natural Sciences, University of Oslo.
No. 671

ISSN 1501-7710

Cover: Inger Sandved Anfinsen.
Printed in Norway: AiT e-dit AS, Oslo, 2007.

Produced in co-operation with Unipub AS.
The thesis is produced by Unipub AS merely in connection with the
thesis defence. Kindly direct all inquiries regarding the thesis to the copyright
holder or the unit which grants the doctorate.

Unipub AS is owned by
The University Foundation for Student Life (SiO)

Abstract

Background: Like any research discipline, software engineering research must be of a certain quality to be valuable. High quality research in software engineering ensures that knowledge is accumulated and helpful advice is given to the industry. One way of assessing research quality is to conduct systematic reviews of the published research literature.

Objective: The purpose of this work was to assess the quality of published experiments in software engineering with respect to the validity of inference and the quality of reporting. More specifically, the aim was to investigate the level of statistical power, the analysis of effect size, the handling of selection bias in quasi-experiments, and the completeness and consistency of the reporting of information regarding subjects, experimental settings, design, analysis, and validity. Furthermore, the work aimed at providing suggestions for improvements, using the potential deficiencies detected as a basis.

Method: The quality was assessed by conducting a systematic review of the 113 experiments published in nine major software engineering journals and three conference proceedings in the decade 1993-2002.

Results: The review revealed that software engineering experiments were generally designed with unacceptably low power and that inadequate attention was paid to issues of statistical power. Effect sizes were sparsely reported and not interpreted with respect to their practical importance for the particular context. There seemed to be little awareness of the importance of controlling for selection bias in quasi-experiments. Moreover, the review revealed a need for more complete and standardized reporting of information, which is crucial for understanding software engineering experiments and judging their results.

Implications: The consequence of low power is that the actual effects of software engineering technologies will not be detected to an acceptable extent. The lack of reporting of effect sizes and the improper interpretation of effect sizes result in ignorance of the practical importance, and thereby the relevance to industry, of experimental results. The lack of control for selection bias in quasi-experiments may make these experiments less credible than randomized experiments. This is an unsatisfactory situation, because quasi-experiments serve an important role in investigating cause-effect relationships in software engineering, for example, in industrial settings. Finally, the incomplete and unstandardized reporting makes it difficult for the reader to understand an experiment and judge its results.

Conclusions: Insufficient quality was revealed in the reviewed experiments. This has implications for inferences drawn from the experiments and might in turn lead to the accumulation of erroneous information and the offering of misleading advice to the industry. Ways to improve this situation are suggested.

Acknowledgement

This thesis work is the tangible result of work in which I have depended on the help, support, and inspiration of many people. First, I wish to thank my supervisors, Dag Sjøberg and Tore Dybå, for including me in their work, for their constructive and scientific guidance, and for their continuous belief in my work. In addition, I wish to thank all my colleges in the Department of Software Engineering for their discussions, inspirations, and recreational support during the four and a half years of my PhD work. A special thanks goes to Jo Hannay for his help and for excellent cooperation. I thank Magne Jørgensen for valuable reviews of parts of the work and for inspiring discussions. I also acknowledge Bente Anda, Erik Arisholm, and Lionel Briand for their help and advice. In addition, I thank Gunnar Bergersen for his infectious enthusiasm and encouragement.

I want to thank the other members of the Context project for their cooperation: Ove Hansen, Amela Karahasanovic, Nils-Kristian Liborg, and Anette Rekdal.

I also acknowledge Reidar Conradi and Jingyue Li for including me in the survey work on COTS-based development. Even if this work does not constitute a direct part of this thesis, it served as interesting and informative variation to the PhD work. I also thank Reidar Conradi for advice on the thesis work.

The Simula Research Laboratory is a unique institution, in which everything is designed to facilitate research of the highest quality. I am grateful for the opportunity to work in such an excellent environment and professional atmosphere, with such helpful staff and proficient researchers. I also thank the Research Council of Norway, the University of Oslo, and Simula Research Laboratory for funding this work. I thank Chris Wright for proofreading this thesis.

I am grateful to my friends for listening to my concerns and for their support. Finally, I offer a heartfelt thanks to my family for their wonderful care, support, and encouragement. My closest family has put up with a great deal due to my working overtime and being mentally absent, so a special debt of gratitude goes to Camilla, Anders, and Inge for allowing me to complete this work.

List of Papers

The following papers are included in this thesis:

1. **A survey of controlled experiments in software engineering**
   Dag I.K. Sjøberg, Jo E. Hannay, Ove Hansen, Vigdis By Kampenes,
   Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal
   In *IEEE Transactions on Software Engineering* Vol. 31, No. 9, pp. 733-753, 2005.

2. **A systematic review of statistical power in software engineering experiments**
   Tore Dybå, Vigdis By Kampenes, and Dag I.K. Sjøberg
   In *Information and Software Technology* Vol. 48, No. 8, pp. 745-755, 2006

3. **A systematic review of effect size in software engineering experiments**
   Vigdis By Kampenes, Tore Dybå, Jo E. Hannay, and Dag I.K. Sjøberg
   In *Information and Software Technology* Vol. 4, No. 11-12, pp.1073-1086, 2007.

4. **A systematic review of quasi-experiments in software engineering**
   Vigdis By Kampenes, Tore Dybå, Jo E. Hannay, and Dag I.K. Sjøberg
   Submitted to *Information and Software Technology*, 2007.

A workshop article on the reporting of effect sizes also constitutes part of the PhD work. However, its content was incorporated in Article 3, so it is not regarded as a separate part of the thesis:

Effect size in empirical software engineering experiments
Vigdis By Kampenes
Presented at the 3rd International Workshop, WSESE2005 in Oulu, Finland, June 13-16
Published in *Guidelines for Empirical Work in Software Engineering*, edited by Andreas Jedlitschka and Marcus Ciolkowski. A publication by Fraunhofer IESE, pp. 14-21, 2005.

While I was working on my PhD, I also contributed to the research methodological aspects of a survey of COTS based development in the IT industry. This work is not included in my thesis:

An empirical study of variations in COTS-based software development processes in norwegian IT industry

Jingyue Li, Finn Olav Bjørnson, Reidar Conradi, and Vigdis By Kampenes

In *Empirical Software Engineering*, Vol. 11, No. 3, pp. 433-461, 2006.

Reflections on conducting an international survey of CBSE in ICT industry

Reidar Conradi, Jingyue Li, Odd Petter Slyngstad, Vigdis By Kampenes, Christian Bunse, Maurizio Morisio, and Marco Torchiano

In *Proceedings of the Fourth International Symposium on Empirical Software Engineering (ISESE'05)*, Noosa Heads, Australia, November 17-18, IEEE Computer Society, pp. 214-223, 2005.

An empirical study on COTS component selection process in norwegian IT companies

Jingyue Li, Finn Olav Bjørnson, Reidar Conradi, and Vigdis By Kampenes

In *Proceedings of the International workshop on models and processes for the evaluation of COTS component (MPEC'04)*, Edinburgh, Scotland, May 25, IEE Press, pp. 27-30, 2004.

Contents

Summary

1 Introduction

An indication of the maturity of a research discipline is the quality of the methods used. One broad category of research methods is the experiment, which is the classical scientific way of identifying cause-effect relationships. This thesis investigates the quality of published software engineering experiments. In this respect, the thesis work differs from traditional PhD work within software engineering, which usually investigates software engineering topics. This introductory chapter further motivates this research perspective.

1.1 Empirical research in software engineering

Software engineering deals with the systematic development, evaluation, and maintenance of software. It is multidisciplinary, in that it embraces technology, human behaviour, and issues of economics (in terms of cost and effectiveness), and language (in terms of syntax and semantics). Given this complexity, it is far from trivial to determine what works and what does not. For example, which software engineering methods, techniques, languages, or tools are most effective for whom in which situation? Or, which software engineering skills are most helpful for performing different types of software engineering tasks?

If such questions are phrased as research questions and evaluated in a research study or in a family of research studies, they can be answered scientifically. If research does not investigate such problems, decisions might be based on who, among the software engineering methods' proponents, shouts the loudest.

People tend to interpret the term research differently. Hence, many activities that claim to be research are, in fact, not. For example, building a system is development, not research, if no research questions are investigated in the process. In 1992, Basili [6] presented four research paradigms that help to distinguish research activities from development activities. The paradigm applied in this thesis is that of *empirical methods*, according to which research questions are those that can be answered by “objective observations” [11] and that are investigated by such methods as experiments, surveys, case-studies, and action research [113]. Central to the use of empirical methods is the importance of experience for the formation of concepts and the acquisition of knowledge [115].

It is important to apply empirical methods in software engineering research for two main reasons: (1) software engineering deals with human performance, and (2) software engineering is an applied discipline. Regarding the human aspect, empirical methods have traditionally been used in social science and psychology, where the concern is human behaviour. Also, it is argued by Wohlin *et al.* [126] that software engineering is very much governed by human behaviour in that people develop, evaluate and maintain software and it is conjectured by Endres and Rombach [38] p. 269 that “Human-based methods can only be studied empirically.” Regarding the applied aspect, if they are to investigate the practical challenges that the IT industry faces, research methods should be based on observations and not on mathematical or theoretical proofs. Hence, software engineering work is best studied by empirical methods.

It is not just single empirical studies that are valuable. In turn, published empirical research can be used in secondary analyses for the purpose of research synthesis, which summarizes or combines the findings of different studies on a topic or a research question [34]. Such research synthesis is one important element in evidence-based research, which aims at making scientifically gathered empirical evidence available to practitioners. Evidence-based software engineering is presented by Dybå, Jørgensen, and Kitchenham, in [37, 59, 64].

The extent of published empirical studies in software engineering has been assessed by Tichy *et al.* [121], Zelkowitz and Wallace [128], and Glass *et al.* [43]. Even though these assessments had different perspectives and collected different types of data, their conclusions were fairly similar: in sum, there is very little use of empirical methods to assess the validity of claims. Whereas Tichy *et al.*, and Zelkowitch and Wallace, claim that the practice should be improved, Glass *et al.* did not criticise current practice, but wonder whether the research community might not benefit from a greater extent of empirical work.

However, the worth of empirical methods in software engineering is emphasized by many researchers [6-8, 39, 73, 113, 120] and *empirical software engineering* (ESE) has become a working concept. In addition, as noted by Sjøberg *et al*. [113], the focus on ESE is reflected in such forums as the Journal of Empirical Software Engineering (EMSE, from 1996), the IEEE International Symposium on Software Metrics (METRICS, from 1993), Empirical Assessment & Evaluation in Software Engineering (EASE, from 1997), and the IEEE International Symposium on Empirical Software Engineering (ISESE, from 2002). From 2007, ISESE and METRICS will be merged into one conference called the International Symposium on Empirical Software Engineering and Measurement (ESEM). Furthermore, in 2000, Perry *et al.* [88] published a roadmap for empirical studies, in 2002, Kitchenham *et al.* [66] provided guidelines for empirical research, in 2003, Endres and Rombach summarized empirical evidence [38], and the future of empirical methods in software engineering research is discussed in a recent article by Sjøberg *et al.* [113]. Furthermore, contributions from the workshop on critical assessments and future directions for ESE issues in 2006 are edited by Basili *et al.*[5] and published and a book on advanced empirical software engineering issues, edited by Shull *et. al* [109] is forthcoming.

1.2 The role of the software engineering experiment

The role of the experiment in software engineering research is to compare different software engineering technologies, methods, etc. with respect to, for example, effectiveness, usefulness, or costs by letting software engineers conduct one or more software engineering tasks. Whereas other empirical methods aim at observing and explaining, the experiment tests hypotheses and can be used as a decision tool. Hence, it plays an important role in answering key questions for practitioners in the IT industry, for example, what works best for a specific development task, method A or Method B? However, the experiment must not be viewed in isolation. As Endres and Romback write: "Learning is best accelerated by a combination of controlled experiments and case-studies", [38] p. 270.

The first experiment in software engineering was reported by Grant in 1967 [44] and up to 1993, only 17 experiments in software engineering were published according to Zendler [129]. The review described in this thesis found 114 published software engineering experiments from 1993-2002. Hence, there was a formidable increase in experimentation in the period 1993-2002 compared with the first two and a half decades in the history of software engineering experimentation. Furthermore, an assessment by Segal *et al.* [103] of publications in the Journal of Empirical Software Engineering from 1997-2003 revealed a dominance of experiments over other empirical methods. In addition, in recent years, guidelines and text books on experimentation suited for software engineering have been published by Kitchenham *et al.* [66], Juristo and Moreno [57], and Wohlin *et al.* [126], as well as additional literature on methods listed in Section 2. Thus, the experiment is receiving increasing attention in software engineering research.

1.3 Assessment of experimental quality

The analysis of experimental results consists of making interpretations of, and drawing conclusions from, quantitative data, often by using statistical methods. Experimental quality can be formally expressed in terms of the *validity* of such inferences. In this thesis, quality is measured in terms of three factors: the validity of inference, and the completeness and consistency of the reporting of experimental information.

Four main types of validity are described by Shadish *et al.* [106]: *Statistical conclusion validity*, *internal validity*, *construct validity* and *external validity*; see Table 1.

Table 1. Validity types

Statistical conclusion validity	The validity of inferences about the correlation (covariation) between treatment and outcome.
Internal validity	The validity of inferences about whether an observed covariation between A (the presumed treatment) and B (the presumed outcome) indicates a causal relationship from A to B as those variables were manipulated or measured.
Construct validity	The degree to which inferences are warranted from the observed persons, settings, and cause and effect operations included in a study to the constructs that these instances might represent.
External validity	The validity of inference about whether the cause-effect relationship holds over variation in persons, settings, treatment variables, and measurement variables.

These types of validity seek to cover decisions that the researcher must face when making inferences from the data:

- Is there a relationship between the variables? (statistical conclusion validity)
- Does the relationship indicate a causal relationship? (internal validity)

- How good is the relationship between the abstract constructs and the sampling particulars? (construct validity)
- How can we generalize from the results? (external validity)

Validity cannot be measured directly, but the experiment can be checked against possible *threats to validity* [106]. In order to enable valid inferences, and thereby conclusions that can be relied upon, the experiment must therefore be designed and analyzed to avoid or control such threats to validity. Only then can the experiments help to provide a foundation for theory building in software engineering and provide practical guidance to the industry, which is the ultimate goal of all research in software engineering.

The importance of quality of reporting is emphasized by Endres and Rombach [38], p. 272: "Empirical results are transferable only if abstracted and packaged with context". It is important to report (1) information that enables the experiment to be replicated, and (2) information that enables the reader to understand and judge the experiment and inferences made.

Conducting experiments is a complex task, which might explain why reports from other research areas show a lack of validity in experimentation and sparse reporting of important experimental information, for example, information systems [4, 95], medicine [3, 20, 32, 47], and social science [22, 25, 60, 61, 84, 102, 106, 118]. Because ESE is a younger research discipline than these other research areas, it probably suffers from similar problems regarding quality. However, we cannot assume that the same problems are present in ESE without verifying their existence. Moreover, the feature of quality challenges might be domain-specific and discussions of directions for improvements must be suited to the specific research problems present within the area in question. Hence, there is a need to assess the quality of experimentation in ESE, to understand the cause of possible insufficiencies, and to provide guidelines to improve the quality of experiments. This is the rationale for the research work described in this thesis, which is a systematic review of software engineering experiments published in the decade 1993-2002.

1.4 Thesis structure

The thesis is organized as follows:

Summary. This part introduces the thesis papers. Section 2 describes the background to the research problem and gives an overview of related literature. Section 3 presents the research questions. Section 4 describes the research method applied. Section 5 summarizes the result of the research. Section 6 summarizes the answers to the research questions, discusses implications of the results, provides recommendations for improvements, presents limitations of the thesis work, and offers directions for future research. Section 7 concludes. Appendix A presents the underlying data-material for this review. Appendix B presents a preliminary review of experiments published in 2007. Then, references for the summary are listed.

Papers. This part includes the four papers of this thesis. The papers assess distinct aspects of the quality of the reviewed controlled experiments and provide recommendations for improvements.

Paper 1: A survey of controlled experiments in software engineering

Dag I.K. Sjøberg, Jo E. Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanovic, Nils-Kristian Liborg, and Anette C. Rekdal
IEEE Transactions on Software Engineering, Vol. 31, No. 9, pp. 733-753, 2005.

Paper 1 summarizes the characteristics of the experiments surveyed, such as topics investigated, tasks performed, the nature of the participants, the type of application systems used, and the experimental environment. Dag Sjoberg provided the idea for this work and initiated it. My contribution was to participate in defining inclusion and exclusion criteria for the selection of articles, hereunder the definition of controlled experiments in software engineering, and to participate in reading and judging articles in the later selection phase. I also participated in the collection of the entire dataset and was responsible for collecting the data on tasks, and internal and external validity. Dag Sjøberg took the lead in the analysis and the writing of the overall article, but I was responsible for several parts of the work.

Abstract: The classical method for identifying cause-effect relationships is to conduct controlled experiments. This paper reports on how controlled experiments in software engineering are conducted at present and the extent to which relevant information is reported. Among the 5,453 scientific articles published in 12 leading software engineering journals and conferences in the decade from 1993 to 2002, 103 articles (1.9 percent) reported controlled experiments in which individuals or teams performed one or more software engineering tasks. This survey characterizes quantitatively the topics of the experiments and their subjects (number of subjects, students versus professionals, recruitment, and rewards for participation), tasks (type of task, duration, and type and size of application), and environments (location, development tools). Furthermore, the survey reports on how internal and external validity is addressed and the extent to which experiments are replicated. The gathered data reflects the relevance of software engineering experiments to industrial practice and the scientific maturity of software engineering research.

Paper 2: A systematic review of statistical power in software engineering experiments

Tore Dybå, Vigdis By Kampenes, and Dag I.K. Sjøberg

Information and Software Technology, Vol. 48, No. 8, pp. 745-755, 2006.

Paper 2 assesses the statistical power level in the experiments and gives recommendations for improvements. Tore Dybå provided the idea for this work and initiated it. All three authors participated in planning the review. I performed an independent review of all the articles identifying primary tests for each experiment. Tore Dybå did the same work and all three authors met to discuss the differences in our findings and agreed on a final set of primary tests. Tore Dybå took the lead in the analysis and writing of the article, with the two authors contributing.

Abstract. Statistical power is an inherent part of empirical studies that employ significance testing and is essential for the planning of studies, for the interpretation of study results, and for the validity of study conclusions. This paper reports a quantitative assessment of the statistical power of empirical software engineering research, using as a basis the 103 papers on controlled experiments (of a total of 5453 papers) published in nine major software engineering journals and three conference proceedings in the decade 1993-2002. The results show that the statistical power of software engineering experiments falls substantially below accepted norms as well as the levels found in the related discipline of information systems research. Given this study's findings, additional attention must be directed to the adequacy of sample sizes and research designs to ensure acceptable levels of statistical power. Furthermore, the current reporting of significance tests should be improved by reporting effect sizes and confidence intervals.

Paper 3: A systematic review of effect size in software engineering experiments

Vigdis By Kampenes, Tore Dybå, Jo E. Hannay and Dag I.K. Sjøberg

To appear in *Information and Software Technology*, 2007.

Paper 3 describes the extent to which effect sizes are reported, how effect sizes have been interpreted, and the values detected in the experiments. I provided the idea for this work and initiated it. I also did the review of the experiments regarding the information about effect sizes and performed the computation of effect sizes, when these were not reported. I took the lead in the analysis and writing of the article, with the three authors contributing.

Abstract. An effect size quantifies the effects of an experimental treatment. Conclusions drawn from the results of tests of hypotheses might be erroneous if effect sizes are not judged in addition to statistical significance. This paper reports a systematic review of 92 controlled experiments published in 12 major software engineering journals and conference proceedings in the decade 1993-2002. The review investigates the practice of effect size reporting, summarizes standardized effect sizes detected in the experiments, discusses the results, and provides recommendations for improvements. Standardized and/or unstandardized effect sizes were reported in 29% of the experiments. Interpretations of the effect sizes in terms of practical importance were not discussed beyond references to standard conventions. The standardized effect sizes computed from the reviewed experiments were equal to observations in psychology studies and slightly larger than standard conventions in behavioural science.

Paper 4: A systematic review of quasi-experiments in software engineering

Vigdis By Kampenes, Tore Dybå, Jo E. Hannay and Dag I.K. Sjøberg

Submitted to *Information and Software Technology*, 2007.

Paper 4 reports on the types of quasi-experiment performed, the extent to which they are performed, and the extent to which they are designed and analysed to handle threats to selection bias. I provided the idea for the work and initiated it. I also did the review of the experiments. In addition, Jo Hannay reviewed parts of the material. I took the lead in the analysis and writing of the article, with the three authors contributing.

Abstract. Experiments in which study units are assigned to experimental groups nonrandomly are called quasi-experiments. They allow investigations of cause-effect relations in settings in which randomization is inappropriate, impractical, or too costly. The procedure by which the nonrandom assignments are made might result in selection bias, that is, pre-experimental differences between the groups that could influence the results. By detecting the cause of the selection bias, and designing and analyzing the experiments accordingly, the effect of the bias may be reduced or eliminated. To investigate how quasi-experiments are performed in software engineering (SE), we conducted a systematic review of the experiments published in nine major SE journals and three conference proceedings in the decade 1993-2002. Among the 114 experiments detected, 35% were quasi-experiments. In addition to field experiments, we found several applications for quasi-experiments in SE. However, there seems to be little awareness of the precise nature of quasi-experiments and the potential for selection bias in them. The term "quasi-experiment" was used in only 10% of the articles reporting quasi-experiments; only half of the quasi-experiments measured a pretest score to control for selection bias, and only 8% reported a threat of selection bias. On average, larger effect sizes were seen in randomized than in quasi-experiments, which might be due to selection bias in the quasi-experiments. We conclude that quasi-experimentation is useful in many settings in SE, but their design and analysis must be improved (in ways described in this paper), to ensure that inferences made from this kind of experiment are valid.

2 Background

This chapter categorizes the literature on the methodology for experimentation in ESE and places the thesis in context. Then, the topics for the assessment of the quality of experiments are described and the challenges that motivated this work are highlighted.

2.1 Types of existing guidelines on experimentation in ESE

Currently, there are 34 scientific articles and three books dedicated to experimental methodology in ESE; see Table 2. The literature includes textbooks, guidelines, assessments, and position papers, all of which have the common feature of offering guidance regarding experimentation, either explicitly or in terms of recommendations based on assessments or experiences. Excluded from this overview is literature that focuses on methods of investigating specific software engineering topics, such as estimation, programming, or defect detection.

In Table 2, this literature is categorized according to (1) whether the guidance is based on a review of the literature or uses empirical data to provide examples only and (2) whether the text focuses on experiments or concerns empirical research in general.

For the majority of the literature, the text is not based on a systematic review. These are guidelines, text books, and position papers that either discuss future directions of experimentation and/or empirical research methods, or address experimental methodology, for example, replications, meta-analysis, or the assessment of statistical power. Twenty-two percent of the texts categorized are literature reviews of published experiments. The majority of these reviews assess the extent to which various empirical research methods are used. Only two articles describe an assessment of experiments: Hannay *et al.* [46], which assesses the use of theory in experiments and Zendler [129], which builds a theory for software engineering practice on the basis of published experiments.

So, the table reveals that few assessments of experiments are performed, even if there are many experimental method issues addressed in the literature. In this respect, this thesis work fills a gap in the ESE literature on the methodology of experimentation.

Note that argumentation can be based on reviews made by others. The overview shown in Table 2 has not taken this aspect into consideration, because it was difficult to categorize the literature accordingly. There were several ways in which studies based their arguments on evidence drawn from reviews made by others: either directly through references to software engineering reviews or reviews in other research fields, or indirectly through references to related guidelines that in turn referred to reviews. In addition, there were various degrees to which studies based their arguments on results from other reviews. Nevertheless, Table 2 illustrates that there is a need for more quantitative assessments on which the literature can be based, either directly or indirectly.

2.2 Quality of design and analysis of experiments

The basics of the design and analysis of experiments are well established and documented; see, for example, [23, 85]. The general fundamentals of statistics are described in text books, such as [10] and separate books are often dedicated to specific statistical methods; see, for example, [24]. However, the appropriate use of the theoretical basis for experimentation is limited by constraints that often occur in practice and that create threats to validity.

The reviewed experiments are investigated according to the following threats to validity, which are due to deficiencies in the design and analysis of the experiment: insufficient statistical power, lack of analysis of effect size, and potential systematic bias in quasi-experiments.

2.2.1 Statistical power

Statistical power is defined as the probability that a statistical test will correctly reject the null hypothesis [29]. A test without sufficient statistical power will not be able to provide the researcher with enough information to draw conclusions regarding the acceptance or rejection of the null hypothesis. Hence, a lack of statistical power is a threat to the validity of conclusions drawn from statistical data.

Knowledge of statistical power can influence each of the planning, execution, and results of empirical research. If the power of statistical tests is weak, the probability of finding significant effects is small, and it is then likely that the outcomes of the study will be insignificant. Furthermore, if the study fails to provide information about the statistical power of its tests, it is not possible to determine whether the insignificant results were due to insufficient power or the phenomenon under investigation actually did not exist. This will inevitably lead to misinterpretation of the outcomes of the study.

Thus, failure to provide an adequate level of statistical power has implications for both the execution and outcome of research: “If resources are limited and preclude attaining a

Table 2 **Research method literature in ESE on experimentation ***

Extent of use of empirical studies	Experimental methodology (Details on specific experimental issues)	Empirical research including experiments (High-level overviews and discussions)
Literature reviews/surveys	Hannay *et al.*2006 [46] – Software engineering theory use in experimentation Zendler 2001 [129] – Theory building from experiments	Zannier *et al.* 2006 [127] – Quantity and quality of empirical evaluations Segal *et al.* 2005 [103] – Nature of evidence from empirical research Shaw 2003 [108] – Reporting advice for software engineering research Glass 2002 *et al.* [43] – Several issues in software engineering research Zelkowitz & Wallace 1997 [128] - Extent of experimental validation Tichy *et al.* 1995 [121] - Extent of exp. evaluation in computer science
Empirical studies used as examples/illustrations	Miller 2005 [79] – Replications Jedlitschka *et al.* 2005 [55] – Reporting guidelines Kitchenham *et al.* 2004 [65] – Human factors Miller 2004 [78] – Statistical significance testing Shull *et al.* 2004 [111] – Knowledge sharing Jørgensen & Sjøberg 2004 [58] – Generalization and theory building Laitenberger & Rombach 2003 [69] – Quasi-experiments Houdek 2003 [54] – External experiments Juristo and Moreno 2003 [57] – Text book on experimentation Shull *et al.* 2002 [110] – Replications Sjøberg *et al.* 2002 [114] – Realism Miller 2000 [77] – Meta-analysis Basili *et al.* 1999 [9] – Families of experiments Wohlin 1999 [126] – Text book on experimentation Singer 1999 [112] – Reporting experimental results Miller *et al.* 1997 [80] – Statistical power Pfleeger 1994/95 [89-93] - Design and analysis Fenton & Pfleeger 1994 [40] – Design and analysis Basili *et al.* 1986 [8] – A framework for experimentation Moher & Schneider 1982 [83] – Methodology and exp. research Moher & Schneider 1981 [82] – Human factors Curtis 1980 [36] – Measurement and experimentation	Basili *et al.*(Eds.) 2007 [5]– ESE issues† Shull *et al.* (Eds.) 2008 [109]– Advanced topics in ESE Sjøberg *et al.* 2007 [113] – The future of empirical methods Endres & Rombach 2003 [38] – Text book chapter on empirical research Kitchenham *et al.* 2002 [66] – Guidelines for empirical research Tichy 1998 [120] – Extent of experimentation Basili 1996 [7] – The role of experimentation Potts 1993 [94] – Realism in software engineering research Basili 1993 [6] – Experimental paradigm

* Authors, year of publication, reference, and keyword for the contents of the literature.
† Two of the contributions were literature reviews of empirical research.

satisfactory level of statistical power, the research is probably not worth the time, effort, and cost of inferential statistics." [4] (p. 96).

The fundamental approach to statistical power analysis was established by Jacob Cohen, who first addressed the issue in 1962 in a description of a review of a volume of the *Journal of Abnormal and Social Psychology* [27]. The result from the review demonstrated the neglect of power issues and motivated Cohen to write his book on statistical power in 1969 [28]. He writes:

> *What behavioral scientist would view with equanimity the question of the probability that his investigation would lead to statistically significant result, i.e., its power? And it was clear to me that most behavioral scientists not only could not answer this and related questions, but were even unaware that such questions were answerable.*
>
> Cohen 1969 [28] (preface)

His book has become a standard reference on statistical power, in large part because of his definitions of small, medium, and large effect sizes, which make power calculations possible when little or no knowledge about the effect size is available. His book was later supplemented by other books [68, 71] and guidelines [3, 124] on statistical power.

Cohen's work has prompted researchers in other disciplines to assess the statistical power of their literature. This is seen in social and abnormal psychology [25, 102], applied psychology [22, 84], education [15], communication [21], behavioural accounting [12], marketing [100], management [19, 41, 74, 84], international business [16], and information systems [4, 95]. All these assessments reported overall insufficient power in the experiments, even if some of the assessments found sufficient power for the detection of large effect sizes.

In ESE, in 1981, Moher *et al.* [82] were the first to describe how to perform power analysis. Moher *et al.* [83] also mention power indirectly through discussions about sample size in 1982. Then power does not seem to be addressed until Miller *et al.* [80] published an article in 1997 about the little used and misunderstood concepts of statistical power. Following this publication, power has been addressed frequently. In their textbook on experimentation published in 1999, Wohlin *et al.* [126] describe the concept of power and list lack of power as a threat to statistical conclusion validity. In 2000, Miller [77] emphasised the importance of reporting the power of the experiment when including non-significant results in meta-analysis. Kitchenham *et al.* [66] published guidelines in 2002 that recommend calculating the minimum sample size required to achieved the expected power. In 2003, Juristo and Moreno [57] described the concept of power and how to determine sample size in their text book on experimentation. Miller mentions power analysis in relation to statistical significance testing in 2004 [78] as well as in relation to the replication of experiments in 2005 [79]. Increased statistical power is part of the vision for future empirical research presented by Sjøberg *et al.* in 2007 [113].

The only assessment of statistical power analysis in software engineering experiments was made by Miller *et al*. [80]. The message was that there is inadequate reporting of, and attention paid to, statistical power in the ESE literature, which leads to potentially flawed research designs and questionable validity of results:

> *Any researcher not undertaking a power analysis of their experiment has no idea of the role that luck or fate is playing with their work and consequently neither does the Software Engineering community.*
>
> *Miller [80] p. 286.*

Although Miller *et al*. [80] made an important contribution in directing attention to the concept of statistical power in ESE research and how it can be incorporated within the experimental design process, they based their arguments on an informal review of the literature. In order to verify whether this result was representative for software engineering experiments in general, it would be necessary to conduct more formal investigations, similar to that of other disciplines, of the state-of-the-practice in ESE research with respect to statistical power. This was the rationale for the thesis work on the assessment of statistical power in software engineering experiments as described in Paper 2.

2.2.2 Effect size

An effect size tells us the degree to which the phenomenon under investigation is present in the population. It is the magnitude of the relationship between treatment variables and outcome variables. There are several types of effect size measures, for example, correlations, odds ratios, and differences between means.

If effect size is not judged as part of the experimental results, incorrect or imprecise conclusions might be drawn. Whereas *p*-values reveal whether a finding is *statistically* significant, effect size indicates *practical* significance, importance, or meaningfulness.

Interpreting effect sizes is thus critical, because it is possible for a finding to be statistically significant but not meaningful, and *vice versa* [31, 71].

Shadish *et al.* [106] describe the inaccurate estimation of effect size as a threat to statistical conclusion validity. They also recommend reporting effect size as part of the results from statistical tests; hence, a lack of reporting of effect size can also be regarded as a threat to statistical conclusion validity.

In addition to being meaningful for the analysis and reporting of experimental results, previously published effect sizes can be used in meta-analyses [50], in statistical power analyses [29, 71], and for purposes of comparison. Such use requires the reporting of either effect sizes, or sufficient data to enable effect sizes to be estimated.

The first approach to determining the magnitude of the effect was published seven decades ago for a study of agricultural treatments [26], but effect size as a concept was first introduced by Cohen in 1969 [28] in his work on power analysis. His definitions of effect size values have become standard, not only for power analysis, but also as reference values when reporting effect sizes as part of experimental results. In 1976, Glass [42] introduced the concept of meta-analysis, as a method of combining the results of studies that used different scales of measurement by applying effect size measures. He proposed two types of measure, which have become de facto standards: the standardized mean difference effect size and the product-moment correlation coefficient.

So, initially, there were two main applications for effect size measures: power analysis and meta-analysis. Then authors started recommending effect size analysis to substitute or supplement the null hypothesis testing procedure [30, 35, 53, 61, 119]. Now, there exist text books on effect size estimation for reporting experimental results [45, 67, 96] and a number of papers that suggest new or adjusted measures of effect size [13, 86, 87, 97, 98].

In psychology research, assessments have revealed an unacceptable low reporting of effect size in published articles [60, 118]. Several journals in social science now require that effect sizes be reported [122], and recommendations for the reporting of effect sizes are included in publishing guidelines for research in medicine [3] and psychology [124], from which the following quotation is found:

> *We must stress again that reporting and interpreting effect sizes in the context of previously reported effects is essential to good research. It enables readers to evaluate the stability of results across samples, designs, and analyses. Reporting effect sizes also informs power analyses and meta-analyses needed in future research.*
>
> Wilkinson and the task Force on Statistical Inference [124], p.599.

There is one major limitation of the effect size measure: there is no unambiguous mapping from an effect size to a value of practical importance. Even small effects might have practical importance. For example, the optimization of a method for detecting defects that yields only a 1% increase in error detection would be of little practical importance for most types of software, but might be of great practical importance for safety-critical software, particularly if the added 1% belongs to the most critical type of errors. Hence, observed effect sizes must be judged in context [13, 35, 53, 61, 99, 101, 117, 122, 124]. This means that a contextual judgment of observed effect sizes must be made and a standardized interpretation avoided. Therefore, in addition to the reporting of effect sizes, a nuanced interpretation and discussion of them is important. Sechrest and Yeaton [101] offer approaches to deciding whether a given difference between groups is large/small, important/unimportant:

- A judgmental approach that combines intuitive judgments with the judgment of experts in the field.
- A normative approach, where the size of effect is compared with empirically based norms.
- A cost-benefit analysis that seeks to establish that the benefits outweigh the costs. Even a small effect may be worthwhile if the costs of producing it are relatively trivial. In software engineering, effort tends to be the major cost drivers, hence a cost-benefit analysis equals a cost-effectiveness analysis, where effect sizes are weighted by the efforts required to produce them.

As an alternative to assessing the standardized effect size for practical importance, Wilkilson *et al.* [124] suggest that the unstandardized effect size should be reported when the unit of measurements are meaningful on a practical level, for example, the mean difference instead of the standardized mean difference. Unstandardized measures of effect size are not given much attention in the literature, but are included in the overview of effect size measures in [72].

In ESE, the magnitude of effect is first mention in relation to power considerations by Moher *et al.* in 1981 [82]. Then it is not addressed until 1995 by Pfleeger [90]. In the planning of the experiment, she recommends asking such questions as “How large a difference will be considered important?” Then, in 1997, Miller *et al.* [80] described the concept of measure of effect size and its role in power analyses. The earliest recommendation that effect size be reported was made by Miller in the context of meta-analyses in 2000:

> *Although the significance test is obviously an important result from the experimental procedure, it is by no means the full story. The effect size is equally important, without it other researchers are in a poor position to estimate the importance of the results, even if they are significant. Unfortunately few, if any, software engineering experiments report effect size estimates, their incorporation into the results of empirical studies would greatly aid other researchers.*
>
> Miller [77], p.37

The reporting of effect size is also recommended by Kitchenham *et al.* in 2002 [66]. The authors also recommend distinguishing between statistical significance and practical importance:

> *...first see whether the result is real (statistical significant); then see whether it matters (practical significance). For example, with a large enough dataset, it is possible to confirm that a correlation as low as 0.1 is significantly different from 0. However, such a low correlation is unlikely to be of any practical importance. In some cases, even if the results are not statistical significant, they may have some practical importance.*
>
> Kitchenham *et al.* [66], p. 731

The reporting of effect size is also recommended by Miller in 2004 [78] as a supplement to significance testing and in 2005 [79] to compare studies and replications. The most recent article that recommends the reporting of effect sizes is the article on the future of empirical methods by Sjøberg *et al.* [113] in 2007.

So, the importance of effect size reporting and the role that effect size has in power analyses and meta-analyses have been addressed earlier in ESE. However, there has been no formal assessment of the extent to which effect sizes are used and, if reported, how they are interpreted. Furthermore, unstandardized effect sizes are not mentioned in the ESE literature and there exists no overview in our field of the standardized effect size values observed. Further discussions of the use of effect size in software engineering experiments will gain from knowledge of the state of practice. Hence, the aim of the systematic review of effect size, as described in Paper 3, was to provide empirical evidence about the use of effect sizes and, on the basis of the findings, to suggest directions for improvement.

2.2.3 Quasi-experimentation

Randomization is the procedure of randomly assigning participants to experimental groups. Experiments in which study units are assigned to experimental groups nonrandomly are called *quasi-experiments* [33]. They allow the investigation of cause-effect relations in settings in which randomization is inappropriate, impractical, or too costly. For example, in software engineering, the costs of teaching the experimental subjects all the technologies (the different treatment conditions) so that they can apply them in a meaningful way may be prohibitive. Moreover, when the levels of participants' skill constitute treatment conditions, or if different departments of companies constitute experimental groups, randomization cannot be used.

The nonrandom assignment procedure might result in *selection bias*, that is, a systematic difference between the experimental groups that could influence the results. For example, when projects are compared within a company, there is a chance that participants within projects are more alike than between projects, e.g., in terms of some types of skill that influence the performance in the experiment. Moreover, if the participants select experimental groups themselves, people with similar backgrounds might select the same group. Such differences between experimental groups might generate other differences of importance for the experimental outcome as well. Hence, selection bias is a threat to internal validity. By detecting the cause of the selection bias, and designing and analyzing the experiments accordingly, the effect of the bias may be reduced or eliminated.

The concept of *randomization* was introduced by Fisher in 1925 [18]. Its use is widespread, because it is the cornerstone that underlies the use of statistical methods. Statistical methods require that the observations are realizations of independently distributed random variables and randomization usually makes this assumption valid [85]. Randomization also prevents any systematic differences between the experimental groups before the experimental tasks are performed. Simple randomization does not guarantee equal experimental groups in a single experiment, but because differences are created only by chance, the various participant characteristics will be divided equally among the treatment conditions in the long run, over several experiments.

However, experimental practices revealed that it is not always possible to achieve ideal methodological circumstances. Moreover, there are experimental settings for which randomization is possible, but not optimal for the purpose of the study. The need for valid inferences from such experiments motivated the work on the theory of *quasi-experimentation*. This work was first presented by Campbell [17] in 1957 and by Campbell and Stanley [18] in 1963 and later developed by Cook and Campbell [33] and Shadish *et al.* [106]. The theory provides the following: (1) alternative experimental designs for studying outcomes when a randomized experiment is not possible, (2) practical advice for implementing quasi-experimental designs, and (3) a conceptual framework for evaluating such research through validity assessments [104]. The theory claims that when properly designed and analysed, quasi-experiments can be good approximations to randomized experiments. Central to the theory is the use of various design elements to control for the potential selection bias that might be present due to the non-random assignment procedure.

Researchers have attempted to assess how elements from the quasi-experimental theory work in practice. This is not trivial because selection bias cannot be measured directly from experimental results. Findings in psychology suggest that by avoiding the self-selection of experimental groups as the assignment method and/or adjusting for pre-experimental differences by using pretest scores, selection bias can be eliminated completely [2], or at least to some extent [51, 52, 75, 105].

However, the quasi-experimental theory seems not to be implemented in practice to any large extent. Shadish *et al.* [106] claim that the most frequently used quasi-experimental designs typically lead to causal conclusions that are ambiguous. Further, empirical results from research in medical science, psychology, and criminology show that randomized experiments and quasi-experiments have provided different results [20, 32, 51, 81, 105, 107, 116, 123, 125].

To improve the performance of nonrandomized experiments, publication guidelines in psychology recommend that researchers determine sources of bias in quasi-experiments, adjust for their effects, and describe how this has been done [124]. Moreover, the importance of conducting quasi-experiments properly has been recognized in fields of research other than psychology, such as environmental science [70], economics [76], and, recently, medical science [47-49].

In ESE, the handling of non-randomized experiments is first mentioned by Pfleeger in 1994 [90]; she recommends documenting the areas where lack of randomization may affect the validity of results in cases where complete randomization is not possible. The term *quasi-experiment* was first used in the ESE literature by Wohlin *et al.* in 1999 [126]. In the context of meta-analyses, Miller [77] recommends using randomization because of the published observed differences in effect sizes reported in epidemiological trials. In their guidelines in 2002, Kitchenham *et al.* [66] recommend identifying and controlling for bias in non-randomized experiments. They also recommend using well-documented experimental designs and consulting a statistician if it is not possible to implement such designs. Then, in 2003, Laitenberger and Rombach [69] described the concept and conduct of quasi-experiments and claimed that quasi-experiments represent a promising approach to increasing the amount of empirical studies in the software engineering industry. In 2007, Sjøberg *et al*. [113] recognised that quasi-experiments will play an important role in future experimental research in ESE, because they offer opportunities to improve the rigour of large-scale industrial studies.

So, the quasi-experiment is recognized as an important part of cause-effect investigations by several researchers in different areas, including ESE. Assessments in other areas of research show that quasi-experiments are poorly performed and that randomized experiments and quasi-experiments sometimes provide different results. Such assessments have not yet been conducted in ESE. In order to determine how the situation can be improved, it is necessary to provide and overview of the state of practice. Furthermore, a discussion of how to handle selection bias in software engineering quasi-experiments requires an overview of the types of quasi-experiments being conducted. The lack of any such overview inspired the work on quasi-experimentation that is described in Paper 4.

2.3 Quality of reporting of experiments

When reporting experiments, it is important to prioritize what information to include. Many reviews have documented deficiencies in reports of clinical trials in medical research, which have resulted in detailed guidelines on what to report [3]. Research in psychology has experienced similar problems and publication guidelines have been developed [1, 124].

In ESE, the method literature presented in Table 2 gives implicit guidelines on what to report through recommendations regarding what issues are important in experimentation. Explicit guidelines on reporting are provided by the following works. In 1987, Basili *et al.* [8] suggested a framework for experimentation that provides a structure for presenting experiments. In 1999, Singer [112] provided an introduction to the American Psychological Association (APA) style guidelines. In 1999, Wohlin *et al.* [126] described the presentation and packaging of experiments and in 2002, Kitchenham provided guidelines for reporting [66]. In 2003, Juristo and Moreno [57] provided a guide to documenting experimentation. Simultaneously, Shaw [108] published advice on how to write good software engineering research papers. With respect to the replication of experiments, knowledge sharing through packages with raw data and text documentations was addressed by Shull and co-authors in two articles from 2002 and 2004 [110, 111]. These articles describe a solution to the problem of space when reporting experiments in journal articles. In 2005 Jedlitschka and Pfahl [55] reported a survey of the most prominent published proposals for reporting guidelines and suggest a unified standard for reporting of controlled experiments. These guidelines have been subject to an evaluation study [63] and an improved version will be provided [56].

Existing guidelines tend to be based on empirical data from other research areas or only on anecdotal evidence. In order to determine more specifically what kinds of guideline are need the most, a systematic assessment of the reporting practices in ESE was required. Such an assessment is provided in this thesis for some experimental issues.

3 Research Questions

The quality of experiments in ESE has not been previously assessed systematically. Hence, a systematic review of published experiments in software engineering and recommendations for improvements based on the findings may be a helpful contribution to, the ideally, continuous process of increasing quality of ESE experiments. More specifically, this research had two main aims:

1. *To provide a quality assessment*. To that end, the extent to which software engineering experiments are designed, analysed, and reported to help enable valid inference from the results must be determined.
2. *To provide recommendations for improvements.* Appropriate ways to address the potential deficiencies found in the quality assessment must be determined.

The assessment of quality is limited to the following issues of design and analysis: statistical power level, effect size analysis, and quasi-experimentation. Statistical power analysis is performed in the design phase, but affects the analysis because the results must be viewed in relation to the planned power. Low power is a threat to statistical conclusion validity. Effect size analysis is performed in the analysis of results. However, it must be considered in the design phase in order to include the magnitude of effect in research questions or the formulation of hypotheses and procedures for gathering data. If effect sizes are not reported, statistical conclusion validity is threatened. Quasi-experimentation requires extra effort in the design and analysis phase in order to eliminate or reduce potential selection bias. Selection bias is a threat to internal validity.

Thus, the experiments are assessed according to aspects of statistical conclusion validity and internal validity. Concept validity and external validity are assessed only in terms of how they are reported in the articles.

The quality of reporting influences the reader’s ability to understand the experiment and validate the results.

The aim of assessing quality is refined into subgoals, captured by the following research questions:

RQ1 What is the statistical power level for the detection of small, medium, and large effect size values?

RQ2a) To what extent is effect size reported as part of the experimental results?

RQ2b) If effect size is reported, how is it interpreted?

RQ3a) To what extent is randomization used in the assignment procedure?

RQ3b) To what extent are quasi-experiments designed and analysed to control for selection bias?

RQ4 To what extent is information regarding the following attributes reported: subjects, experimental setting, experimental design, analysis, and validity?

RQ1 is answered in Paper 2, RQs 2a-b are answered in Paper 3, and RQs 3a-b are answered in Paper 4. RQ4 is addressed in all four papers, but especially emphasized in Paper 1.

4 Research Method

This section describes the execution of the systematic review. A systematic review is a rigorous and auditable method for evaluating and interpreting all available research relevant to a particular research question, topic area, or phenomenon of interest [62]. Using existing guidelines for medical researchers as a basis, Kitchenham [62] described the following procedures for performing systematic reviews:

1. Identification of the need for a review
2. Development of a review protocol
3. Identification of research
4. Selection of primary studies
5. Study quality assessment
6. Data extraction & monitoring
7. Data synthesis
8. Reporting the review

This review work started two years before these guidelines were available. Hence, these procedures have not been followed strictly, but have been used as guidance in the later phases of the work. Still, the research method of the thesis can be described in terms of the main steps described in the guidelines, as shown below.

4.1 Identification of the need for a review

The aim of this investigation was to make an empirical assessment of software engineering experiments and, on the basis of the findings, provide recommendations for improvements. The necessity of making valid inferences from the results provides the motivation for this work.

The chosen research method was a systematic review of published experiments over a decade, because published articles are the main source of information about experiments conducted world wide. By making the assessment a quantitative review of the literature, the state of practice of software engineering experimentation would be revealed. In addition, a thorough empirical foundation would be established, upon which further qualitative investigations of experimentation could be based, for example, elaborations of the reasons for the quantitative findings.

An investigation of related work on assessments of experimentation in software engineering revealed that the major difference between those assessments and this review work is that they describe the extent and characteristics of various types of empirical study, while this review provide an in-depth study of controlled experiments only; see Paper 1 for details.

4.2 Development of a review protocol

The first part of this review involved several people and was organised as a research project. This part comprised the selection of experiments, as well as the data gathering, analysis, and reporting of the experimental issues described in Paper 1. For this part, decisions regarding the planning and conducting the review were made in weekly meetings and substantiated in a document that took the form of a comprehensive version of the upcoming journal article. In addition, decisions were documented in meeting reports and separate database documentation. Elements in the planning process were

- research questions,
- procedures for selection of studies,
- operational definition of a controlled experiment,
- inclusion and exclusion criteria,
- data to be extracted,
- reporting strategies, and
- time schedule and distribution of tasks.

The second part of the systematic review comprised the investigation of statistical power, effect size, and quasi-experimentation, which are described in Papers 2-4. As the database of articles was already established, this part only comprised data extraction, analysis, and reporting, as well as the planning of these activities. No formal protocol documents were made for this part, because few people were involved. The researcher responsible documented definitions and organised the data collection.

4.3 Identification of research

This review included 113 experiments in software engineering that were found in 103 articles published in nine major journals and three conference proceedings in the decade from 1993 to 2002; see Table 3. We consider these included journals to be leaders in software engineering research. Furthermore, ICSE is the principal conference in software engineering, and ISESE, Metrics, and EASE are major venues in empirical software engineering that report a relatively high proportion of controlled software engineering experiments. The conference Empirical Assessment & Evaluation in Software Engineering (EASE) is partially included, in that 10 selected articles from EASE appear in special issues of JSS, EMSE, and IST.

Table 3. Distribution of ESE studies employing controlled experiments: Jan. 1993 – Dec. 2002.

Journal/Conference Proceeding	Number	Percent
Journal of Systems and Software (JSS)	24	23.3
Empirical Software Engineering (EMSE)	22	21.4
IEEE Transactions on Software Engineering (TSE)	17	16.5
International Conference on Software Engineering (ICSE)	12	11.7
IEEE International Symposium on Software Metrics (METRICS)	10	9.7
Information and Software Technology (IST)	8	7.8
IEEE Software	4	3.9
IEEE International Symposium on Empirical Software Engineering (ISESE)	3	2.9
Software Maintenance and Evolution (SME)	2	1.9
ACM Transactions on Software Engineering Methodology (TOSEM)	1	1.0
Software: Practice and Experience (SP&E)	–	–
IEEE Computer	–	–
TOTAL:	103	100%

4.4 Selection of primary studies

In order to identify and extract article that described controlled experiments, one researcher systematically read the titles and abstracts of the 5,453 scientific articles published in the selected journals and conference proceedings for the period 1993-2002. Excluded from the search were editorials, prefaces, article summaries, interviews, news, reviews, correspondence, discussions, comments, reader's letters, and summaries of tutorials, workshops, panels, and poster sessions. If it was unclear from the title or abstract whether a controlled experiment was described, the entire article was read by both the same researcher and another person in the project team. Note that identifying the relevant articles is not straightforward because the terminology in this area is confusing. For example, several authors claim that they describe experiments even though no treatment is applied in the study. The following operational definition of a software engineering experiment was used in the review:

> *Software engineering experiment: A randomized experiment or a quasi-experiment in which individuals or teams (the experimental units) conduct one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages, or tools (the treatments).*

Inclusion criteria were as follows: the use of at least two treatment conditions, subjects, or teams as experimental units, and the performance of a software engineering task. In addition, the study had to be a cause-effect investigation, i.e., the use of a treatment had to precede the measure of an outcome.

Excluded from the review were several types of study that share certain characteristics with experiments, but do not apply the deliberate intervention essential to experiments. So, correlation studies, studies that are based solely on calculations using existing data (e.g., from data mining), and evaluations of simulated teams based on data for individuals were excluded. The last category falls outside the operational definition because the units are constructed after the run of the experiment. Studies that use projects or companies as treatment groups, in which data is collected at several levels (treatment defined, but no experimental unit defined) were also excluded. These were considered to be multiple case studies (even though the authors might refer to them as experiments). Also excluded were articles that, at the outset, would not provide sufficient data for our analyses (e.g., summaries of research programs). Moreover, usability experiments were not included because those are part of another discipline (human computer interaction). The list of included articles is provided in Appendix A.

4.5 Study quality assessment

Because the review aimed at assessing the quality of experiments, no experiment was excluded from the dataset on the grounds of a lack of quality. However, for investigations of statistical power and effect size, which were done on the level of statistical tests, seven experiments were excluded because we were unable to track which tests answered which hypothesis or research question.

4.6 Data extraction & monitoring

For the first part of the review (Paper 1), six researchers gathered data so that each aspect was covered by at least two persons. After the initial analysis, the results were compared and possible conflicts resolved by reviewing the articles collectively a third time or handing the article over to a third person.

For the investigation of statistical power (Paper 2), two researchers identified the primary statistical tests independently. A third researcher was then involved in reaching a consensus on which experiments and tests to include, using these two datasets as a basis.

Data for the effect size investigation (Paper3) was extracted by one researcher, whereas a dual review was done for parts of the data extraction in the investigation of quasi-experimentation (Paper 4).

The data from the first part of the review was stored in a relational database (MS SQL Server 2000). Data extracted for the investigation of power, effect size, and assignment methods were stored in separate excel sheets.

The total data model is shown in Figure 1. Some data was specific to an article, some was specific to an experiment, and some information concerned the combination of article and experiment. For example, an article might describe several experiments and a single experiment might be described in several articles, typically with a different focus in each article. Moreover, some data was specific to a statistical test or a task and some experiments were not analysed by statistical testing. Four experiments were reported in more than one article. In these cases, for some parts of the review, the data from the most recently published article was used for reporting, as recommended in [62]. Which articles that are included in each part of the review is described in Appendix A, as well as article-categorizations for some assessments.

4.7 Data synthesis and reporting the review

The data synthesis was a descriptive, quantitative analysis. All results relevant to the investigation were tabulated and figures were used when appropriate. The reviews were reported in the four journal articles, which constitute the main part of this thesis.

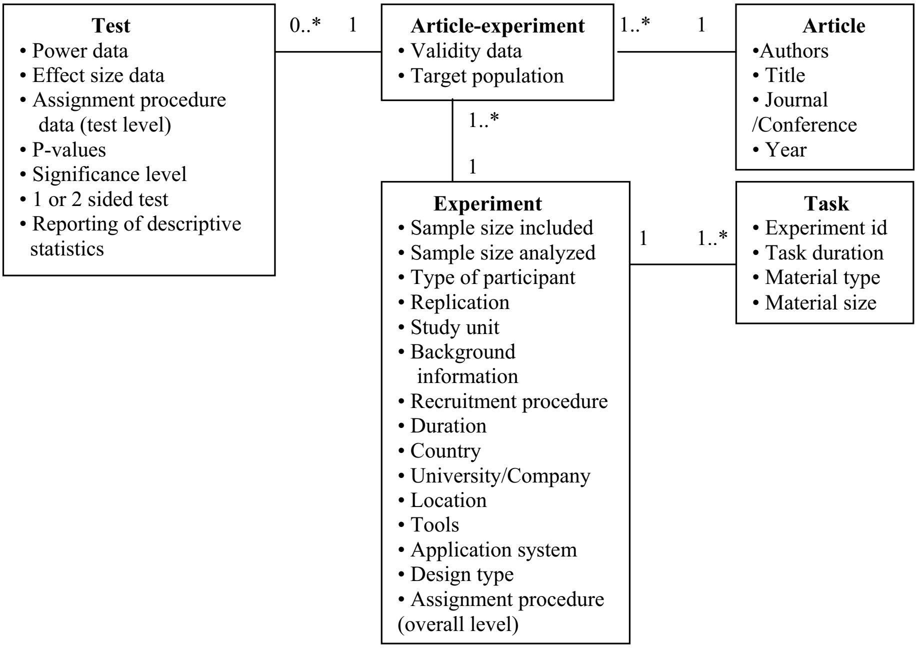

Figure 1. The data model for the review

5 Results

This section describes the results of the review: the assessments of statistical power, effect size analyses, quasi-experimentation, and quality of reporting.

5.1 Assessment of statistical power

The assessment of statistical power answered research question 1:

> *RQ 1: What is the statistical power level for the detection of small, medium and large effect size values?*

The investigation of statistical power is described in detail in Paper 2. This part of the review included the 92 experiments for which statistical testing was performed and the tests clearly described. For each primary statistical test in the experiment, the power was calculated on the basis of the type of statistical test and sample size. A significance level of 0.05 was used for all the tests and the power was calculated for small, medium, and large effect sizes as defined by Cohen [29]. By using this information, which is available in the planning phase of the experiment, the power calculated represents the pre-experimental power and hence shows how the experiment was designed with regard to power.

The result revealed an average power for detecting medium effect sizes in the software engineering experiments of 0.36, i.e., there was, on average, a probability of 0.36 that a null hypothesis would be rejected correctly; see Table 4. This power is far below the commonly accepted level of 0.8, which is also assumed to be the target level by most IS researchers [95].

Power increases with increasing effect size, provided that all other factors are kept constant. However, the average power for detecting large effect sizes, according to Cohen’s definition, was 0.63, which is also below the commonly acceptable level.

The power level of the tests would still have been acceptable if the effect sizes in ESE overall had been large. Unfortunately, this does not seem to be the case, judging from the results of the effect size computation (Paper 3). The median effect size value estimated from the experimental tests was 0.60 and even though 29% of the effect sizes were very large (above 1.10), 53% were of small or medium size (Table 4).

Table 4. Statistical power and observed effect sizes

	Small effect size	Medium effect size	Large effect size
Average power Based on 459 tests (Paper 2)	0.11	0.36	0.63
Proportion of effect sizes * Based on 284 tests for which effect size was possible to estimate (Paper 3)	30%	23%	47%

* Standardized mean difference effect size was estimated for all tests. In this table, values in (0-0.35) are categorized as "small", (0.26-0.65) as "medium" and (0.66, ->) as "large".

An additional indication that little attention is paid to considerations of power is that only 15% of the articles referred to the power of their significance test, and for only one experiment was it reported that an *a priori* power analysis had been performed.

The consequence of this low level of statistical power is that it is likely that many software engineering experiments fail to detect the actual effects of the technology being investigated. This review revealed that significance at the 0.05 level was achieved for half the tests (Table 5). Hence, combining this result with the low power observed suggests that increased power in software engineering experiments will lead to more tests being significant.

Table 5. Extent of statistical significance

	Tests	
Results	Number	Percentage
p-value < 0.05	119	51.3
p-value > 0.05	113	48.7
Total	232	100.0

5.2 Assessment of effect size analysis

The review of effect size reporting used all 113 experiments and answered research questions 2a) and 2b):

RQ 2a: To what extent is effect size reported as part of the experimental results?

RQ 2b: If effect size is reported, how is it interpreted?

The assessment of the 92 experiments that performed significance testing and described the tests clearly is presented in detail in Paper 3.

Overall, only 27 of the 113 descriptions of experiments (24%) reported at least one effect size (Table 6). All these experiments reported effect size as a supplement to information about statistical significance, whereas none of the experiments that did not use statistical testing reported any effect size. Only two of the experiments reported both standardized and unstandardized effect sizes.

Table 6. Extent of effect size reporting

Analysis method	Number of experiments	Experiments reporting effect size	
		Number	Percentage
Significance testing	99	27	27%
Descriptive statistics only	14	0	0
Total	113	27	24%

*In Paper 3, only 92 experiments were included in the investigation of effect size. Included here are (1) the additional seven experiments that used significance testing, but for which it was difficult to identify primary tests or main aims and (2) the 14 experiments for which statistical testing was not performed.

The reporting of unstandardized effect size was done more frequently for significant, than for non-significant, results. Another factor that seemed to influence the extent of effect size reporting is the number of treatment conditions tested in the experiment. None of the 51 primary tests that compared more than two treatment conditions reported the standardized effect size for the pairwise comparisons of treatments. Only four of these 51 tests reported the unstandardized effect size.

An important aspect of effect size reporting is the interpretation of its value. Even if the unstandardized effect size lends itself better to discussions of practical importance than does the standardized one, the only references to practical importance were made with respect to standardized effect sizes. In these cases, reference was made to Cohen's definitions of small, medium, and large values. Hence, the practical importance of the values was not discussed directly in relation to contextual factors, which is the recommended (but difficult) practice. This result is not unexpected, because few guidelines exist on how to discuss the practical importance of the results on the basis of effect size measures in general, and no guidelines directed to software engineering experiments in particular. Still, the result revealed insufficiencies that need to be addressed and discussed in the ESE community.

The unstandardized effect sizes appeared to be very suitable for discussions of the practical importance, for example, “Procedural roles reduced the loss of only singular defects by about 30%.” However, no such discussion was added to these measures.

5.3 Assessment of quasi-experimentation

This part of the review was based on all the 113 experiments and answered research questions 3a and 3b.

RQ 3a: *To what extent is randomization used in the assignment procedure?*

RQ 3b: *To what extent are quasi-experiments designed and analysed to control for selection bias?*

The results are described in detail in Paper 4. Among the 113 experiments, 66 were randomized experiments (58%) and 40 were quasi-experiments (35%), while the assignment procedure could not be obtained for 7 experiments (6%).

There seemed to be little knowledge about quasi-experimentation, because only four reports used the term *quasi-experiment*, only three of the quasi-experiments addressed threats to validity regarding selection bias, and relatively few used design elements to control for selection bias in the analysis. Regarding design elements, fewer than half of the experiments applied a pretest score to control for a potential selection bias and, apart from crossover design seen in eight quasi-experiments, no other ways of controlling for selection bias was observed.

The results suggest a need for better control regarding selection bias in software engineering experiments, in order to ensure valid inferences. Moreover, increased reporting of possible threats to selection bias that might influence the result is required, so that readers will understand the challenges in the experiments and can judge the results on this basis.

A comparison of the results from quasi-experiments with randomized experiments revealed lower average effect sizes in the quasi-experiments than in the randomized ones. There were few data points in this comparison of effect sizes; hence, this result should be investigated further in follow-up studies. Still, we should take note of the results, because the hypothesis that selection bias might influence the results from quasi-experiments has a theoretical foundation [106] and has empirical support in other research fields.

In order to discuss the use of quasi-experiments in software engineering, we must know the types of non-random assignment procedures that are used. This review detected four types; see Table 7.

(1) The non-equivalent experimental group design is the typical quasi-experimental design, which is described thoroughly in the literature [33, 106]. It was used in 38% of the quasi-experiments. Examples are field experiments in which professionals were included into the experimental groups on the basis of their availability and student experiments in which two sections of a class constituted the experimental groups on the basis of convenience. A third example is the investigation of how software engineering skills influenced performance for different technologies. For such comparisons, the most appropriate inclusion of participants to skill groups is to select subjects who already have skills, which is a non-random assignment procedure.

(2) Haphazard assignment is a non-random assignment procedure with no known bias, for example, when participants are assigned to experimental groups on an alternating basis from a sorted list. Haphazard assignment was used in 30% of the quasi-experiments.

Table 7. Types of quasi-experiments in software engineering

Type of quasi-experimental design	Number	Percent
Non-equivalent experimental group design	15	37.5
Haphazard assignment	12	30.0
Some randomization	7	17.5
Intra-subject experiments in which all participants applied the treatment conditions in the same order	6	15.0
Total	40	100.0

(3) Seven of the experiments were a combination of quasi-experiments and randomized experiments; hence, some of the comparisons in the experiments were exposed to a non-random assignment procedure.

(4) For six of the experiments, all the participants applied all treatments in the same order, only once. The reasons for choosing such designs are an expected larger learning effect from one of the technologies (which prevents a crossover design) combined with few available participants (which prevent an inter-subject design). However, this is a weak quasi-experimental design because it does not allow proper control of how learning effects may influence the second technology.

Only 45% of the quasi-experiments measured a pretest score of the participants's performance ability and none of the experiments attempted to measure such a score for teams of participants beyond averaging individual skills. Hence, how to measure software engineering skill appear to be a challenge for the ESE community.

5.4 Assessment of quality of reporting

The assessment of the quality of reporting answered research question 4:

RQ 4: To what extent is information regarding the following attributes reported: subjects, experimental setting, experimental design, analysis, and validity?

The quality of reporting was assessed in all parts of this review and is described in all the four papers included in this thesis, but is particularly emphasised in Paper 1. The major findings are now summarized.

Large variations in the quality of reporting are seen both across types of information assessed and across experiments. Insufficiencies include incomplete reporting, information reported at different places in the articles, and lack of consistent terminology. An example is the reporting of validity considerations that were made for ¾ of the experiments, at different places in the articles, and under different headings. For 54 experiments (48%), there was a special section entitled "Threats to (internal/external) validity" or other combinations that included the terms "threats" or "validity." Nine other experiments (eight%) had special sections on threats to validity but with other names (e.g., "Limitations to the results"). The reporting of threats to validity in yet another eight experiments were found in other sections.

An overview of the extent of the reporting of information regarding subjects, experimental setting, experimental design and analysis, and validity assessments is presented in Table 8.

Information regarding subjects was reported by most of the experiments in terms of sample size, types of subjects, and background information. However, only 21% reported the amount of drop-outs. Moreover, the type of background information and level of detail varied substantially. An example of detailed information on programming experience is: "On average, subjects' previous programming experience was 7.5 years, using 4.6 different programming languages with a largest program of 3510 LOC. Before the course, 69 percent of the subjects had some previous experience with object-oriented programming, 58 percent with programming GUIs." An example of a high-level description without figures is: "Some of the students had industrial programming experience." How the participants were recruited was described for only 36% of the experiments.

A description of the task performed was provided for all the experiments, but the duration of the performance was reported for only 61%. In addition, descriptions of the size of the materials and the use of tools were reported for slightly more than half the experiments.

Regarding experimental design and analysis, some experiments applied standard design names and referred to textbooks, while others just described the design in their own words. Moreover, whether a between-subject or a within-subject design was used for the particular tests was not always stated explicitly and was sometimes difficult to identify. Overall, issues of design and analysis were sparsely addressed. Only one experiment defined the population of subjects to which the results could be generalized. Moreover, as described in the previous sections, the assessments of power, effect size, and assignment procedures revealed incomplete reporting of these issues.

Even if internal and external validity were discussed in 2/3 of the experiments, most of these discussions took the form of a defence for the design and conducting of the experiment. Hence, threats to validity seemed underreported. Reports of only 5% and 11% of the experiments contained a discussion of statistical conclusion validity and construct validity, respectively.

Table 8. Extent of reporting for various experimental variables

Information attributes	Variables	Extent of reporting. Number of experiments		
		N	Total	%
Subjects	Sample size	113	113	100
	Mortality rate	24	113	21.2
	Type (student/professionals)	112	113	99.1
	Recruitment (Voluntarily/mandatory)	41	113	36.3
	Some kind of background information	99	113	87.6
	- Programming experience	37	113	32.7
	- Work experience	24	113	21.2
	- Task related experience	80	113	70.8
	- Grades	6	113	5.3
Experimental setting	Task	113	113	100.0
	Duration	69	113	61.1
	Application system	101	113	89.4
	Size of materials	67	113	59.3
	Location	40	113	35.4
	The use of tools	62	113	54.9
Design and analysis	Well-defined population	1	113	0.9
	Statistical power	1	92	1.1
	Effect size *	27	92	29.3
	Information available for estimation of at least one effect size	64	92	69.6
	Assignment procedure (randomized or quasi)	86	113	76.1
	Randomization method	3	66	4.5
Validity/limitations	Discussion of internal validity	71	113	62.8
	Threats to internal validity	26	113	23.0
	Discussion of external validity	78	113	69.0
	Discussing of statistical conclusion validity†	5	99	5.1
	Discussion of construct validity†	12	113	10.6

Note: Which experiments and articles that are included in these assessments is described in Appendix A.
* Extent of reporting refers to the number of experiments with at least one effect size reported.
† The number of experiments that discuss statistical conclusion validity and/or construct validity is based on the explicit use of these terms. The reporting of these types of validity needs to be investigated more thoroughly in future work.

6 Discussion

This section summarizes the answers to the research questions, discusses implications of the results, provides recommendations for improvements, presents limitations of the thesis work, and offers directions for future research.

6.1 Answers to the research questions

Below are the answers to each research question.

- *RQ1: What is the statistical power level for the detection of small, medium and large effect size values?*

  The average statistical power levels for detection of small, medium, and large effect size values were, 0.11, 0.36, and 0.63, respectively, which is below acceptable norms as well as below the levels found in the related discipline of IS research. In addition, and perhaps as an explanation for the low power level, the review revealed that inadequate attention was paid to power issues in the articles, with respect to the discussion, use, and reporting of statistical power analysis. This indicates that considerations of statistical power are underemphasized in experimental software engineering research.
- *RQ2a: To what extent is effect size reported as part of the experimental results?*

  Effect size was reported for only 24% of the experiments. Only two of the experiments reported both standardized and unstandardized effect sizes. Unstandardized effect sizes were reported more frequently for significant results than for non-significant result. None of the 51 primary tests that compared more than two treatment conditions reported the standardized effect size for the pairwise comparisons of treatments. Only four of these 51 tests reported the unstandardized effect size.
- *RQ2b: If effect size is reported, how is it interpreted?*

  Interpretations of the standardized effect sizes were made mostly in terms of references to Cohen’s definitions of values for small, medium, and large effect sizes. The practical implications of the results were not discussed in relation to contextual factors. Unstandardized effect sizes appeared to be very useful as a basis for discussions regarding the practical importance of the results. However, no interpretations or thorough discussions of these values were made.

- *RQ3a: To what extent is randomization used in the assignment procedure?*

  Randomization was performed in the majority of the experiments (58%), which suggests that many researchers in software engineering are aware that randomization is the most effective way of handling threats to internal validity. However, randomization is not always desirable or possible in SE, to which the percentage of quasi-experiments (35%) bears witness.
- *RQ3b: To what extent are quasi-experiments designed and analysed to control for selection bias?*

  Approximately half of the quasi-experiments applied design elements to control for selection bias; only three reported a threat to selection bias, and only four called the experiment a quasi-experiment. Hence, the impression is that there is little awareness of quasi-experimentation among researchers in software engineering.
- *RQ4: To what extent is information regarding the following attributes reported: subjects, experimental setting, experimental design, analysis, and validity?*

  Large variations in reporting quality are seen both across types of information assessed and across experiments. Insufficiencies include incomplete reporting, information reported at different places in the articles, and a lack of consistent terminology. Information about subjects and experimental settings varied substantially. For example, sample size and a description of tasks were reported for all the experiments, whereas information regarding recruitment and location were reported for less than 40 %. Furthermore, the subject's background information and the level of detail of this information varied to a large extent across experiments. For the most part, information regarding design, analysis, and validity was reported sparsely.

6.2 Implications

Low statistical power, sparse reporting of effect size, and insufficient handling of selection bias in quasi-experiments present threats to valid inference. In turn, this might lead to deficiencies in the accumulation of knowledge and the presentation of advice to industry.

More specifically, the implication of low statistical power is that the actual effects of new technologies or other types of treatment that are tested in the experiments will not be detected to an acceptable extent. Only half of the primary tests were significant at the 0.05 level, which supports this claim. In turn, low powered experiments might not be replicated, due to non-significant findings. Moreover, in addition to influencing single studies, low power may also result in invalid inferences being made from meta-analyses that include low-powered studies. In sum, low-powered experiments will tend to produce an inconsistent body of literature, thus hindering the advancement of knowledge.

Sparse reporting of effect sizes means that the inference from the hypothesis testing result is based on the p-values for most experiments. Because p-values provide no information about the practical importance of the results, the inferences made might be erroneous, or at least too little nuanced. More specifically, if an experiment includes a sufficient number of subjects, it is always possible to identify statistically significant differences, while if the experiment includes too few subjects (i.e. if it has insufficient power), p-values may be misleading.

A consequence of not interpreting the practical importance of effect size in relation to contextual factors is that the practical importance of the results will not be judged, because there is no unambiguous mapping from effect size measures to a measure of practical importance. For example, a medium effect size might be important for detecting an inspection technique in one domain, whereas a large effect size is required for a specific testing technique to be cost-effective. This means that applying Cohen's conventions mechanically has the same unwanted consequences as using the p-value mechanically.

When applying a non-random assignment procedure, the researcher must control for potential selection bias. The consequence of not controlling for potential selection bias in quasi-experiments, by using appropriate design elements, is that selection bias might influence the results. Hence, the observed effect might be caused by factors other than the treatment.

Incomplete and unstandardized reporting of experimental information and results means that readers will have difficulty in understanding the experiment and judging the result. Furthermore, little and arbitrary reporting on context variables, such as the experimental setting and the participants's skills hinders the accumulation of knowledge regarding which context factors influence which kinds of performance.

6.3 Recommendations for improvements

One main impression from the quality assessment is that the design and analysis of experiments needs to be better suited to the experimental situation at hand. A tendency seems to be to analyse all experiments as if they were randomized experiments with sufficient power even if this is not the case, with the aim of making a yes/no decision about the hypotheses tested on the basis of the results. Hence, the overall recommendation that issues from the assessment of experimental quality is a more deliberate use of design elements and an analysis that better adheres to the limitations of the experiment. Moreover, there is a need for more complete and standardized reporting of information that is crucial for understanding the experiment and judging the result.

Based on the findings, the following three major recommendations regarding software engineering experimentation are given: *include effect size considerations and power considerations in the planning of the experiment; be aware of the extra effort required for quasi-experimentation;* and *improve completeness and the standardization of reporting.* These recommendations are elaborated below.

6.3.1 Include effect size considerations and power considerations in the planning of the experiment

The low statistical power and the sparse reporting of both considerations of power and effect sizes suggest that a major challenge in software engineering experimentation is to specify which size of effect to detect in the experiment and to report and interpret effect size values.

There are three reasons for including considerations of effect size in the planning stages of the experiment. (1) Statements about which effect sizes are interesting to detect enable hypotheses to be formulated concretely and informatively, in comparison to the standard: “null difference” versus “not null difference”. (2) Considering effect size early forces the researcher to evaluate the outcome measure with regard to its usefulness in the inference process. If the measure is difficult to transform into effect size measures, other measures should be considered. (3) Considering effect size allows power to be considered, i.e., the sample size required to obtain a certain power is computed for a given effect size, test, and significance level. If this computation shows that an unrealistically large sample size is required, the researcher must change elements of the design and repeat the sample size computation in order to achieve acceptable power for the main test. Alternatively, if it is impossible to achieve acceptable power, the experiment will still have value as an exploratory study as long as this is made explicit.

For determining the effect size to be detected in the experiment, the researcher can both assess similar empirical research in the area and use the effect sizes found in these studies as a guide, and look at their own studies and pilot studies for guidance. Due to the limited number of empirical studies in software engineering, this approach may be difficult to apply at present [80]. However, increased reporting of effect size and discussions of their values will improve the current availability of effect size values. As a guide for the probability of achieving certain standardized effect sizes in software engineering experiments, the range of the two types of standardized effect size values detected in software engineering experiments can be used (Paper 3). Moreover, Cohen's definitions of small, medium, and large standardized effect size values available for several statistical tests are a useful aid when no other information is available. In addition to considerations regarding standardized effect sizes, the corresponding unstandardized effect sizes should be assessed. This is because the researcher needs to reflcet upon the practical importance of the various possible effect size values when the experiment is being planned and because the unstandardized effect size is better suited for such judgements than are the standardized ones.

6.3.2 Be aware of the extra effort required for quasi-experimentation.

This investigation revealed a need for improved design and analysis of quasi-experiments in ESE. More specifically, in order to control for selection bias, design elements such as pretest scores, crossover design, and several comparison groups should be used to a greater extent than is the case at present. If selection bias cannot be controlled for, quasi-experimental designs should be avoided, because it will be difficult to determine whether the result is due to the treatment or other factors.

Thirty percent of the quasi-experiments used haphazard assignment. In all of these experiments, the groups were formed so as to be balanced regarding one type of participant skill. This shows that, for many researchers, a non-random assignment procedure is viewed as being more appropriate than randomization for balancing the experimental groups. However, even if haphazard assignment might be a good approximation to randomization, little is known about its consequences, whereas the statistical consequences of randomization procedures have been well researched [106]. Therefore, whenever feasible, the researcher should use randomization, for example, blocked randomization based on one type of skill, in order to utilize the advantages of randomization.

Some experiments use randomization for some primary tests and a non-random assignment procedure for other primary tests. The author must make it explicit in the text that they are using such a mix and control threats to selection bias in the quasi-experimental part of the experiment.

Since there has been an increased focus on quasi-experiments in the method literature in recent years and since the importance of such experiments has been emphasized [69, 113], we might see an increase in experiments that use a quasi-experimental design. Such an increase will make it even more important to consider how to improve the conducting of quasi-experiments in software engineering.

6.3.3 Improve completeness and the standardization of reporting.

Authors of scientific articles have limited space available and must prioritize what information to report. The impression from the review is that the reporting of many tests is prioritized in the service of the complete reporting of a few tests. This is not a recommended practice. The quality of reporting will benefit from complete and thorough reporting of the major results only.

The findings from the assessment of the quality of reporting revealed that some information that is crucial for understanding and judging the experiment was reported for less than half the experiments. There is great room for improvement in the reporting of such information, as listed below.

- *Recruitment.* Recruiting subjects to experiments is not a trivial task, from either a methodological or a practical point of view. For example, volunteers may bias the results because they are often more motivated, skilled, etc., than are subjects who take part because it is mandatory in some way.
- *Location.* There is a trade-off between realism and control regarding the location of an experiment. Running an experiment in the usual office environment of subjects that are professionals allows a certain amount of realism, yet increases the threat to internal validity due to breaks, phone calls, and other interruptions. Controlling and monitoring the experiment is easier in a laboratory set up, but in such a setting, realism suffers.
- *Well-defined population.* If one tests hypotheses using statistics, it is necessary to have a well-defined population from which the sample is drawn [66].

- *Mortality rate.* All the experiments reported the sample size, which means that there is general agreement on the importance of this variable. However, there are two types of sample size: the number of subjects initially included in the experiment and the number of subjects included in the data analysis. Both these numbers must be reported, as well as the reasons for drop-outs or exclusions.
- *Statistical power.* Information from significance testing is incomplete if the statistical power is not included. In particular, if no significance is found, the result should be judged against the level of statistical power. The reporting of power compensates to some degree for the lack of validity due to low power or extremely high power, because the reader will be informed about how the power influences the result and can draw inferences accordingly.
- *Effect size.* The recommendation is to always report both a standardized and an unstandardized effect size measure, because they serve different, supplementary purposes. The standardized effect size aids other researchers in using the results. Moreover, it embraces both the location and spread of all the observations. The unstandardized effect size is easier to interpret than the standardized one and is therefore better suited as a basis for discussions of the practical importance of the results.
- *Randomization method.* If the method of randomization is not reported, the reader will be in no position to judge whether the procedure is in accordance with recommendations for randomization procedures.
- *Threats to validity.* Validity assessments should be reported for all experiments. It is difficult to report threats objectively, but the attempt must be made. All the potential types of threats to validity described by Shadish *et al.* [106] must be assessed, but not necessarily discussed due to space limitations in the article. The focus should be on reporting actual threats only. Threats that are handled or that are not a problem in the particular experiment can be omitted, because a thorough description of experimental design will include such information.

In the current section, special emphasis is given to the variables that are reported most infrequently. Nevertheless, all the variables listed in Table 5 should be reported. Hence, Table 5 can be used as a checklist to help to improve the completeness of the reporting of software engineering experiments. However, this is not a complete list, and researchers in software engineering should consult additional guidelines, such as those offered by Kitchenham *et al.* [66] and Jedlitschka *et al.* [55, 56].

The second issue in reporting quality is the location within the paper of the reporting of various issues. Experimental issues were described in various places in the articles, which often made information difficult to find. The experience with the review work suggested the following recommendation for reporting elements:

- structure abstracts appropriately,
- place all information about experimental design and conduct in one section,
- describe the methods of analysis used in one section,
- present the results in a single section,
- present threats to validity in one section, and
- conclude the paper in one section.

6.4 Limitations to this investigation

The main limitations to this research are publication selection bias and inaccuracy in data extraction, which are described in the individual papers. These limitations are summarized below.

- The review included published articles in what are regarded as the major journals and conference proceedings in software engineering in general and empirical software engineering in particular. Still, some experiments may have been overlooked, some of which might have provided useful insight to this review finding.
- An additional threat regarding the set of selected articles is that there is a risk that the findings are obsolete; the articles selected are from 5-14 years old. Therefore, a preliminary systematic review of experiments published in 2007 has been performed, see Appendix B. The results indicate that the recommendations given in this thesis are still relevant today.
- There exist no keyword standards for extracting controlled experiments from journals in a consistent manner. The operational definition of a controlled experiment with corresponding inclusion and exclusion criteria were used for the inclusion of articles. Still, the process was difficult and some experiments might have been overlooked.
- The lack of completeness and consistency in reporting made it difficult to gather the data. For example, it was not always clear from the reporting of the studies which hypothesis were actually tested, which significance tests corresponded to which

hypothesis, or how many observations were included for each test; hence, the extraction process may have resulted in inaccuracy in the data.

- Not all the variables were gathered by several researchers. Even if these variables were double checked by the same researcher, this represents a limitation of the process by which data was gathered.

Moreover, the review process did not follow all the steps for a systematic review that are suggested in [62]. In particular, for the investigation of effect size and quasi-experimentation, the research questions were changed during the review, which turned into an iterative process. Moreover, the process by which data was gathered became iterative because the gathered data triggered the collection of additional data. *Pre-review mapping* and *piloting the review protocol*, as suggested in [14], might have helped to reduce the number of iterations. In addition, the authors of the selected papers were not contacted for validity of the classification of their respective paper, although the procedure was partly applied in Paper 4. If the authors were contacted, issues might have been cleared.

6.5 Future work

Among the areas for future work identified through this research are the following:

- *Reasons for lack of quality.* The quantitative assessments performed in this thesis described current practice, but did not reveal the reasons for the practices. Hence, it would be interesting to follow up the findings by conducting a qualitative investigation, for example, a survey or interviews aimed at extracting reasons for the lack of reporting of power and effect size.

- *Similar reviews of other experimental topics.* This review shows that quantitative assessments of methodological aspects of software engineering research are valuable. The findings reveal insufficiencies and act as a basis for discussions of future practices. Hence, similar assessments of other experimental topics will contribute to the improvements of experimental quality in ESE. Examples of such topics are: a more detailed analysis of how experimental design is described in the articles; an investigation of what types of design are performed; whether or not the methods analysis used are appropriate for the design of the experiment; the extent to which the hypotheses and research questions are supported by similar research; the extent to which the results are discussed in the context of related research; An investigation of what types of measures (constructs) are used; and whether or not, and if so to what extent, do researchers tend to adapt to already used measures or develop their own measures suited for their experiment.

Systematic reviews of methodological topics are not constrained to experiments. Future work includes similar reviews of, for example, case studies and surveys.

- *The impact of context variables.* This review revealed a relatively low and arbitrary reporting of context variables, which might influence the results. Future work should investigate the extent to which the variation in the performance of subjects can be explained by their background, such as education and work experience, and to increase our knowledge of the impact of using students versus professionals as subjects in software engineering experiments.

- *Effect size of practical importance.* The investigation of effect size reveals that effect size is seldom reported and that practical importance is seldom discussed on the basis of the effect sizes. The recommendations provided in this thesis assume that the reporting of effect sizes influences the quality of inferences made from the results and that the lack of reporting of effect sizes is due to a lack of knowledge about its importance. However, an alternative explanation is that the interpretation of effect sizes is too difficult for effect sizes to have any value for the making of inferences. Future work should include further discussions and research on how to report and interpret effect size in software engineering experiments.

- *Selection bias in quasi-experiments.* This review found different results from quasi-experiments and randomized experiments. This finding should be investigated further, to reveal the effect of bias from different types of non-random assignment procedures in software engineering experiments. It is also of major interest to explore the extent to which the different types of design element eliminate or reduce the effect of bias. This can be investigated in experiments and in simulation studies.

- *Statistical conclusion and construct validity.* Only 5% of the experiments explicitly mentioned statistical conclusion validity and only 11% explicitly mentioned construct validity. However, these types of validity may have been addressed under different names and this possibility should be investigated further. Moreover, interesting future work would include assessments of which types of threat are reported.

- *Replication of this review.* This review revealed a need for increased statistical power, effect size reporting, control for selection bias in quasi-experiments, and completeness of reporting. It is hoped that this review and the corresponding recommendations for improvements, as well as other recently published guidelines, will inspire researchers in software engineering to improve current practice. In order to evaluate whether this has been the case, a replication of this review should be performed by assessing software engineering experiments published in the decade 2003-2012.

- *Further development and evaluation of the guidelines.* This thesis work consists of review results and guidelines. In combination, these two elements are ment to informe and inspire researchers to improve their experimental quality. How successful this approach is should be evaluated by (1) inspections as suggested by Kitchenham *et al.* [63] and (2) an investigation of the amount of papers making citation to the guidelines and assess whether the papers apply the recommendations. In addition, the guidelines must be consider to be further developed, for example, by providing a more detailed guidance on how to report effect size for different types of tests.

7 Conclusion

Software engineering research must be of a certain quality to be valuable. The quality of research can be investigated by conducting systematic reviews of the published literature, as was the case in this thesis.

Insufficient experimental quality was revealed with respect to the validity of inference and the completeness and consistency of the reporting of the experiments and their results. More specifically, this review revealed a need for an increased level of statistical power, increased use of effect size analysis, increased control for selection bias in quasi-experiments, and more complete and standardized reporting of these issues and the information regarding experimental subjects and settings. However, implementing these improvements face certain difficulties. Challenges and suggested approaches for meeting them are:

- *Estimation and interpretation of effect size values.* The challenge of estimating or guessing an effect size during the planning of the experiment is probably a major reason why statistical power is not considered. In addition, the interpretation of observed effect sizes is not straightforward and might explain why effect sizes are not reported well enough.

  Increased attention should be paid to effect sizes in the reporting of experiments. Researchers should report both standardized and unstandardized effect sizes and discuss these measures and the obtained values.

- *Difficulty in including a sufficient number of subjects to achieve acceptable power.* Particularly for experiments with professionals, it may be difficult to obtain large sample sizes in software engineering experiments. Even if attempts must be made to increase power, low-power experiments can still be valuable. However, such experiments are more exploratory than a well-designed experiment and this must be stated explicitly in the report. Statistical power must be reported and discussed as part of the results if significance testing is performed. An alternative is to omit significance testing and analyse the results by effect sizes and confidence intervals only.

- *Little knowledge of which skill factors that influence different types of performance for different types of technologies*. In order to allow pretest-based control with selection bias in quasi-experiments, we need more knowledge about the effect that different types of subject skill have on the performance of software engineering tasks. If researchers increase their reporting of how subjects' skills are distributed in their experimental groups, meta-studies can investigate how different types of skill influence performance in various experimental settings.

Appendix A. The underlying data-material for this review

This Appendix lists the reviewed articles, describes which articles that are used in each part of the review and provides information about article-categorization in parts of the analysis.

A.1 Experiments and articles used in each part of the review

There are 103 articles included in this systematic review [1, 103], which reports 113 unique controlled experiments. A total of 12 articles reports more than one experiment [2, 20, 39, 42, 43, 48, 56, 66, 75, 95, 96, 103]. Four of the experiments are reported in more than one article:

- one experiment was reported in [37, 38, 66]
- one experiment was reported in [69, 70]
- one experiment was reported in [8, 9, 11, 28]
- one experiment was reported in [72, 73]

Those articles that report the same experiments describe different research focus and different analyses of the data from the particular experiment. Hence, these articles are not "duplicates". There were 120 article-experiments in the study database. For the parts of this review that assessed analysis issues, only one article per experiment (the most recently published one) is included, because we wanted the unit of assessment to be unique experiments.

A.1.1. Experiments and articles included in the review of statistical power (Paper 2)

In the review of statistical power, 92 experiments are included. The exclusion of articles is described below:

- For fourteen experiments, no statistical testing was performed. These experiments are excluded from the review. The following articles each report one of these experiments: [14, 18, 22-24, 30, 45, 47, 51, 61, 100]. In addition, two experiments without statistical testing is reported in [96]. These **twelve articles** are excluded from the review of statistical power. One of the three experiments described in [95] did not perform statistical testing. Hence the experiment, but not the article, is excluded from the review.

- For seven experiments, we were not able to track which tests answered which hypothesis or research question. These are reported in the following **eight articles**, which are excluded from the review of statistical power [10, 41, 69, 70, 76, 85, 94, 97].
- Only one article per experiment is included in the review of statistical power. Hence, the following **five articles** are excluded [8, 9, 11, 37, 72]. One description of one experiment is excluded from [66], but the article also reports another experiment and is therefore not excluded.

There are 78 articles (103-12-8-5) included in the review of statistical power.

A.1.2. Experiments and articles included in the review of effect size (Paper 3)

The same 92 experiments and 78 articles included in the review of statistical power are included in the review of effect size, as described in Paper 3. In addition, a review of *the reporting of effect size* was performed for the 21 remaining experiments (reported in 20 articles) that were originally excluded from the statistical power and effect size investigation, i.e., the experiments for which no statistical testing was performed and for which we were not able to track which tests answered which hypothesis or research question [10, 14, 18, 22-24, 30, 41, 45, 47, 51, 61, 70, 76, 85, 94-97, 100]. The result from this additional review was presented in the summary of the thesis.

A.1.3. Experiments and articles included in the review of quasi-experiments (Paper 4)

All the 113 experiments were included in the review of quasi-experiments. Only one article per experiment was included and, hence, the following six articles were excluded: [8, 9, 11, 37, 69, 72]. These articles were used as additional source for information, but the data gathering was based on the most recently published article of the particular experiment.

A.1.4. Experiments and articles included in the assessment of reporting quality (all papers)

All the 103 articles describing the 113 experiments are included in the review that is described in Paper 1. Those articles that describe the same experiment were assessed in combination, in order to provide as complete information as possible about the particular experiment regarding *topic, subjects, tasks and experimental setting*.

A summary of the assessment of reporting quality is provided in Table 8 in the summary of the thesis. Information regarding *design and analysis* and *validity/limitations* were gathered from one of the following sets of experiments/articles:

- unique experiments reported in the most recently published article (113 experiments, 97 articles), six articles were excluded: [8, 9, 11, 37, 69, 72].
    - randomized experiments (66 experiments).
- unique experiments with statistical tests performed (99 experiments, 91 articles), see the description above of included experiments/articles in the review of statistical power.
- unique experiments with clearly described tests-hypotheses connection (92 experiments, 78 articles), see descriptions above.

A.2. Information about article-categorization in parts of the analysis

Reporting of power. Of the 78 papers in the review of statistical power, 12 articles discuss statistical power associated with the testing of null hypotheses [12, 13, 20, 25, 48, 49, 55, 58, 62, 64, 101, 103], while only one of the papers performed an a priori power analysis and used it to guide the choice of sample size [101].

Reporting of effect size. The following articles report at least one effect size for at least one of the reported experiments:

- Both standardized and unstandardized effect size are reported in two articles and two experiments [4, 49]
- Standardized effect size only is reported in five articles and eight experiments [12, 13, 39, 48, 64]
- Unstandardized effect size only is reported in 15 articles and 17 experiments [3, 17, 20, 27, 32, 33, 50, 54, 56, 75, 80, 82, 86, 92, 93]

Assignment procedure. In the mail-correspondence with the authors of unknown assignment procedures, I stated that the articles would be kept anonymous. Therefore, lists of articles categorized as quasi-experiments and randomized experiments are not provided.

References for the reviewed articles from 1993-2002

[1] T.K. Abdel-Hamid, K. Sengupta, and D. Ronan, Software project control: an experimental investigation of judgment with fallible information, *IEEE Transactions on Software Engineering* 19 (6) (1993) 603-612.

[2] R. Agarwal, P. De, and A.P. Sinha, Comprehending object and process models: an empirical study, *IEEE Transactions on Software Engineering* 25 (4) (1999) 541-556.

[3] E. Arisholm, D.I.K. Sjøberg, and M. Jørgensen, Assessing the changeability of two object-oriented design alternatives? A controlled experiment, *Empirical Software Engineering* 6 (3) (2001) 231-237.

[4] V.R. Basili, S. Green, O. Laitenberger, F. Lanubile, F. Shull, S. Sorumgard, and M.V. Zelkowitz, The empirical investigation of perspective-based reading, *Empirical Software Engineering* 1 (2) (1996) 133-164.

[5] A.C. Benander, B. Benander, and H. Pu, Recursion vs. iteration: an empirical study of comprehension, *The Journal of Systems and Software* 32 (1) (1996) 73-82.

[6] A.C. Benander, B.A. Benander, and J. Sang, An empirical analysis of debugging performance? Differences between iterative and recursive constructs, *The Journal of Systems and Software* 54 (1) (2000) 17-28.

[7] A. Bianchi, F. Lanubile, and G. Visaggio, A controlled experiment to assess the effectiveness of inspection meetings, *Proceedings of the Seventh International Software Metrics Symposium (METRICS'01)*, London, England, April 4-6 IEEE Computer Society (2001) 42-50.

[8] S. Biffl, Using inspection data for defect estimation, *IEEE Software* 17 (6) (2000) 36-43.

[9] S. Biffl and W. Grossmann, Evaluating the accuracy of defect estimation models based on inspection data from two inspection cycles, *Proceedings of the 23rd international conference on Software engineering (ICSE)*, Toronto, Canada, May 12-19 IEEE Computer Society (2001) 145-154.

[10] S. Biffl and M. Halling, Investigating the influence of inspector capability factors with four inspection techniques on inspection performance, *Proceedings of the 8th International Software Metrics Symposium (METRICS'02)* Ottawa, Canada, June 4-7 IEEE Computer Society (2002) 107-117.

[11] S. Biffl, B. Freimut, and O. Laitenberger, Investigating the cost-effectiveness of reinspections in software development, *Proceedings of the 23rd international conference on Software engineering (ICSE)*, Toronto, Canada, Mai 12-19 IEEE Computer Society (2001) 155-164.

[12] L.C. Briand, C. Bunse, and J.W. Daly, A controlled experiment for evaluating quality guidelines on the maintainability of object-oriented designs, *IEEE Transactions on Software Engineering* 27 (6) (2001) 513-530.

[13] L.C. Briand, C. Bunse, J.W. Daly, and C. Differding, Technical communication: an experimental comparison of the maintainability of object-oriented and structured design documents, *Empirical Software Engineering* 2 (3) (1997) 291-312.

[14] A. Brooks, F. Utbult, C. Mulligan, and R. Jeffery, Early lifecycle work: influence of individual characteristics, methodological constraints, and interface constraints, *Empirical Software Engineering* 5 (3) (2000) 269-285.

[15] J.M. Burkhardt, F. Detienne, and S. Wiedenbeck, Object-oriented program comprehension: effect of expertise, task and phase, *Empirical Software Engineering* 7 (2) (2002) 115-156.

[16] C. Calero, M. Piattini, and M. Genero, Empirical validation of referential integrity metrics, *Information and Software Technology* 43 (15) (2001) 949-957.

[17] M. Cartwright, An empirical view of inheritance, *Information and Software Technology* 40 (14) (1998) 795-799.

[18] D.Y. Chen and P.J. Lee, On the study of software reuse using reusable C++ components, *The Journal of Systems and Software* 20 (1) (1993) 19-36.

[19] K. Cox and K. Phalp, Replicating the CREWS use case authoring guidelines experiment, *Empirical Software Engineering* 5 (3) (2000) 245-267.

[20] J. Daly, A. Brooks, J. Miller, M. Roper, and M. Wood, Evaluating inheritance depth on the maintainability of object-oriented software, *Empirical Software Engineering* 1 (2) (1996) 109-132.

[21] D.E.H. Damian, A. Eberlein, M.L.G. Shaw, and B. Gaines, Using different communication media in requirements negotiation, *IEEE Software* 17 (3) (2000) 28-36.

[22] A. Drappa and J. Ludewig, Simulation in software engineering training, *Proceedings of the 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, June 4-11 ACM (2000) 199-208.

[23] A. Dunsmore, M. Roper, and M. Wood, Object-oriented inspection in the face of delocalisation, *ICSE. Proceedings of the 22nd international conference on Software engineering*, (2000) 467-476.

[24] A. Dunsmore, M. Roper, and M. Wood, Systematic object-oriented inspection an empirical study, *ICSE. Proceedings of the 23rd international conference on Software engineering*, (2001) 135-144.

[25] A. Dunsmore, M. Roper, and M. Wood, Further investigations into the development and evaluation of reading techniques for object-oriented code inspection, *ICSE. Proceedings of the 24th international conference on Software engineering*, (2002) 47-57.

[26] K. Finney, K. Rennolls, and A. Fedorec, Measuring the comprehensibility of Z specifications, *The Journal of Systems and Software* 42 (1) (1998) 3-15.

[27] W.B. Frakes and T.P. Pole, An empirical study of representation methods for reusable software components, *IEEE Transactions on Software Engineering* 20 (8) (1994) 617-630.

[28] B. Freimut, O. Laitenberger, and S. Biffl, Investigating the impact of reading techniques on the accuracy of different defect content estimation techniques, *Proceedings of the Seventh International Software Metrics Symposium (METRICS'01)*
London, England, April 4-6 IEEE Computer Society (2001) 51-62.

[29] P. Fusaro, F. Lanubile, and G. Visaggio, A relicated experiment to assess Requirements inspection techniques, *Empirical Software Engineering* 2 (1) (1997) 39-57.

[30] L.D. Gowen and J.S. Collofello, Assessing traditional verification's effectiveness on safety-critical software systems, *The Journal of Systems and Software* 26 (2) (1994) 103-115.

[31] R. Harrison, S. Counsell, and R. Nithi, Experimental assessment of the effect of inheritance on the maintainability of object-oriented systems, *The Journal of Systems and Software* 52 (2-3) (2000) 173-179.

[32] S.M. Henry and K. Todd Stevens, Using Belbin's leadership role to improve team effectiveness: an empirical investigation, *The Journal of Systems and Software* 44 (3) (1999) 241-250.

[33] G.S. Howard, T. Bodnovich, T. Janicki, J. Liegle, S. Klein, P. Albert, and D. Cannon, The efficacy of matching information systems development methodologies with application characteristics - an empirical study, *The Journal of Systems and Software* 45 (3) (1999) 177-195.

[34] M. Höst and C. Wohlin, An experimental study of individual subjective effort estimation and combinations of the estimates, *Proceedings of the 20th international conference on Software engineering (ICSE)*, Kyoto, Japan, April 19-25 IEEE Computer Society (1998) 332-339.

[35] M. Höst and C. Johansson, Evaluation of code review methods through interviews and experimentation, *The Journal of Systems and Software* 52 (2-3) (2000) 113-120.

[36] M. Höst, B. Regnell, and C. Wohlin, Using students as subjects - a comparative study of students and professionals in lead-time impact assessment, *Empirical Software Engineering* 5 (3) (2000) 201-214.

[37] P.M. Johnson and D. Tjahjono, Assessing software review meetings: a controlled experimental study using CSRS, *Proceedings of the 19th international conference on Software engineering (ICSE)* Boston, Massachusetts, USA, May 17-23 ACM Press (1997) 118-127.

[38] P.M. Johnson and D. Tjahjono, Does Every Inspection Really Need a Meeting?, *Empirical Software Engineering* 3 (1) (1998) 9-35.

[39] M. Jørgensen and D.I.K. Sjøberg, Impact of effort estimates on software project work, *Information and Software Technology* 43 (15) (2001) 939-948.

[40] M. Keil, L. Wallace, D. Turk, G. Dixon-Randall, and U. Nulden, An investigation of risk perception and risk propensity on the decision to continue a software development project, *The Journal of Systems and Software* 53 (2) (2000) 145-157.

[41] R.B. Kieburtz, L. Walton, L. McKinney, J.M. Bell, J. Hook, A. Kotov, J. Lewis, D.P. Oliva, T. Sheard, and I. Smith, A software engineering experiment in software component generation, *Proceedings of the 18th international conference on Software engineering (ICSE)* Berlin, Germany, March 25-29 IEEE Computer Society (1996) 542-552.

[42] J.D. Kiper, B. Auernheimer, and C.K. Ames, Visual depiction of decision statements: what is best for programmers and non-programmers?, *Empirical Software Engineering* 2 (4) (1997) 361-379.

[43] J. Koskinen, Experimental evaluation of hypertext access structures, *Software Maintenance and Evolution* 14 (2) (2002) 83-108.

[44] R. Krovi and A. Chandra, User cognitive representations: the case for an object-oriented model, *The Journal of Systems and Software* 43 (3) (1998) 165-176.

[45] S. Kusumoto, A. Chimura, T. Kikuno, K. Matsumoto, and Y. Mohri, A promising approach to two-person software review in educational environment, *The Journal of Systems and Software* 40 (3) (1998) 115-123.

[46] O. Laitenberger and J.M. DeBaud, Perspective-based reading of code documents at Robert Bosch GmbH, *Information and Software Technology* 39 (11) (1997) 781-791.

[47] O. Laitenberger and H.M. Dreyer, Evaluating the usefulness and the ease of use of a web-based inspection data collection tool, *Proceedings of the 5th International Software Metrics Symposium (METRICS)*, Maryland, USA, March 20-21 IEEE Computer Society (1998) 122-132.

[48] O. Laitenberger, K. El Emam, and T.G. Harbich, An internally replicated quasi-experimental comparison of checklist and perspective based reading of code documents, *IEEE Transactions on Software Engineering* 27 (5) (2001) 387-421.

[49] O. Laitenberger, C. Atkinson, M. Schlich, and K. El Emam, An experimental comparison of reading techniques for defect detection in UML design documents, *The Journal of Systems and Software* 53 (2) (2000) 183-204.

[50] L.P.W. Land, C. Sauer, and R. Jeffery, The use of procedural roles in code inspections: an experimental study, *Empirical Software Engineering* 5 (1) (2000) 11-34.

[51] F. Lanubile, F. Shull, and V.R. Basili, Experimenting with error abstraction in requirements documents, *5th IEEE International Software Metrics Symposium (METRICS)*, Maryland, USA, March 20-21 IEEE Computer Society (1998) 114-121.

[52] M. Lattanzi and S. Henry, Software reuse using C++ classes: the question of inheritance, *The Journal of Systems and Software* 41 (2) (1998) 127-132.

[53] K.B. Lloyd and D.J. Jankowski, A cognitive information processing and information theory approach to diagram clarity: a synthesis and experimental investigation, *The Journal of Systems and Software* 45 (3) (1999) 203-214.

[54] C.M. Lott, Technical communication: a controlled experiment to evaluate on-line process guidance, *Empirical Software Engineering* 2 (3) (1997) 269-289.

[55] F. MacDonald and J. Miller, A comparison of tool-based and paper-based software inspection, *Empirical Software Engineering* 3 (3) (1998) 233-253.

[56] R.A. Maxion and R.T. Olszewski, Eliminating exception handling errors with dependability cases: a comparative, empirical study, *IEEE Transactions on Software Engineering* 26 (9) (2000) 888-906.

[57] P. McCarthy, A.A. Porter, H. Siy, and L.G. Votta Jr, An experiment to assess cost-benefits of inspection meetings and their alternatives: a pilot study, *3rd IEEE International Software Metrics Symposium (METRICS)*, March 25-26 (1996) 100-111.

[58] J. Miller, M. Wood, and M. Roper, Further experiences with scenarios and checklists, *Empirical Software Engineering* 3 (1) (1998) 37-64.

[59] K.L. Mills, An experimental evaluation of specification techniques for improving functional testing, *The Journal of Systems and Software* 32 (1) (1996) 83-95.

[60] T. Moynihan, An experimental comparison of object-orientation and functional-decomposition as paradigms for communicating system functionality to users, *The Journal of Systems and Software* 33 (2) (1996) 163-169.

[61] M.C. Ohlsson, C. Wohlin, and B. Regnell, A project effort estimation study, *Information and Software Technology* 40 (11-12) (1998) 831-839.

[62] M.C. Otero and J.J. Dolado, An initial experimental assessment of the dynamic modeling in UML, *Empirical Software Engineering* 7 (1) (2002) 27-47.

[63] M. Peleg and D. Dori, The model multiplicity problem: experimenting with real-time specification methods, *IEEE Transactions on Software Engineering* 26 (8) (2000) 742-759.

[64] D. Pfahl, N. Koval, and G. Ruhe, An experiment for evaluating the effectiveness of using a system dynamics simulation model in software project management education, *7th IEEE International Software Metrics Symposium (METRICS)*, London, England, April 4-6 IEEE Computer Society (2001) 97-109.

[65] A.A. Porter and L.G. Votta, An experiment to assess different defect detection methods for software requirements inspections, *Proceedings of the 16th international conference on Software engineering (ICSE)*, (1994) 103-112.

[66] A.A. Porter and P.M. Johnson, Assessing software review meetings: results of a comparative analysis of two experimental studies, *IEEE Transactions on Software Engineering* 23 (3) (1997) 129-145.

[67] A.A. Porter and L. Votta, Comparing detection methods for software requirements inspections: a replication using professional subjects, *Empirical Software Engineering* 3 (4) (1998) 355-379.

[68] A.A. Porter, L.G. Votta, and V.R. Jr. Basili, Comparing detection methods for software requirements inspections: a replicated experiment, *IEEE Transactions on Software Engineering* 21 (6) (1995) 563-575.

[69] A.A. Porter, H.P. Siy, C.A. Toman, and L.G. Votta, An experiment to assess the cost-benefits of code inspections in large scale software development, *IEEE Transactions on Software Engineering* 23 (6) (1997) 329-346.

[70] A.A. Porter, H. Siy, A. Mockus, and L. Votta, Understanding the sources of variation in software inspections, *ACM Transactions on Software Engineering and Methodology* 7 (1) (1998) 41-79.

[71] L. Prechelt, Accelerating learning from experience: avoiding defects faster, *IEEE Software* 18 (6) (2001) 56-61.

[72] L. Prechelt and W.F. Tichy, An experiment to assess the benefits of inter-module type checking, *3rd IEEE International Software Metrics Symposium (METRICS)*, Berlin, Germany, March 25-26 IEEE Computer Society (1996) 112-119.

[73] L. Prechelt and W.F. Tichy, A controlled experiment to assess the benefits of procedure argument type checking, *IEEE Transactions on Software Engineering* 24 (4) (1998) 302-312.

[74] L. Prechelt and B. Unger, An experiment measuring the effects of personal software process (PSP) training, *IEEE Transactions on Software Engineering* 27 (5) (2000) 465-472.

[75] L. Prechelt, B. Unger-Lamprecht, M. Philippsen, and W.F. Tichy, Two controlled experiments assessing the usefulness of design pattern documentation in program maintenance, *IEEE Transactions on Software Engineering* 28 (6) (2002) 595-606.

[76] L. Prechelt, B. Unger, W.F. Tichy, P. Brossler, and L.G. Votta, A controlled experiment in maintenance: comparing design patterns to simpler solutions, *IEEE Transactions on Software Engineering* 27 (12) (2001) 1134-1144.

[77] S. Ramanujan, R.W. Scamell, and J.R. Shah, An experimental investigation of the impact of individual, program, and organizational characteristics on software maintenance effort, *The Journal of Systems and Software* 54 (2) (2000) 137-157.

[78] V. Ramesh and G. Browne, Expressing casual relationships in conceptual database schemas, *The Journal of Systems and Software* 45 (3) (1999) 225-232.

[79] B. Regnell, P. Runeson, and T. Thelin, Are the perspectives really different? Further experimentation on Scenario-Based reading of requirements, *Empirical Software Engineering* 5 (4) (2000) 331-356.

[80] M. Roper, M. Wood, and J. Miller, An empirical evaluation of defect detection technique, *Information and Software Technology* 39 (11) (1997) 763-775.

[81] K.J. Rothermel, C.R. Cook, M.M. Burnett, J. Sconfeld, T.R.G. Green, and G. Rothermel, WYSIWYT testing in the spreadsheet paradigm: an empirical evaluation, *Proceedings of the 22nd International Conference on Software Engineering (ICSE)* Limerick, Ireland, June 4-11 (2000) 230-239.

[82] G. Sabaliauskaite, F. Matsukawa, S. Kusumoto, and K. Inoue, Experimental comparison of checklist-based reading and perspective-based reading for UML design document inspection reading, *International Symposium on Empirical Software Engineeering (ISESE)*, Nara, Japan, October 3-4 IEEE Computer Society (2002) 148-160.

[83] K. Sandahl, O. Blomkvist, J. Karlsson, C. Krysander, M. Lindvall, and N. Ohlsson, An extended replication of an experiment for assessing methods for software requirements inspections, *Empirical Software Engineering* 3 (4) (1998) 327-354.

[84] B.G. Silverman and T. Mehzer, A study of strategies for computerized critiquing of programmers, *Empirical Software Engineering* 2 (4) (1997) 339-359.

[85] A.E.K. Sobel and M.R. Clarkson, Formal methods application: an empirical tale of software development, *IEEE Transactions on Software Engineering* 28 (3) (2002) 308-316.

[86] M.G. Sobol, A. Kagan, and H. Shimura, Performance criteria for relational databases in different normal forms, *The Journal of Systems and Software* 34 (1) (1996) 31-42.

[87] E. Stensrud and I. Myrtveit, Human performance estimating with analogy and regression models: an empirical validation, *5th IEEE International Software Metrics Symposium (METRICS)*, March 20-21 (1998) 205-213.

[88] K. Takahashi, A. Oka, S. Yamamoto, and S. Isoda, A comparative study of structured and text-oriented analysis and design methodologies, *The Journal of Systems and Software* 28 (1) (1995) 69-75.

[89] T. Thelin, P. Runeson, and B. Regnell, Usage-based reading - an experiment to guide reviewers with use cases, *Information and Software Technology* 43 (15) (2001) 925-938.

[90] T. Thelin, P. Runeson, C. Wohlin, T. Olsson, and C. Anderson, How much information is needed for usage-based reading, *International Symposium on Empirical Software Engineering (ISESE)*, Nara, Japan, October 3-4 IEEE Computer Society (2002) 127-138.

[91] M. Tortorella and G. Visaggio, Evaluation of a scenario-based reading technique for analysing process components, *Software Maintenance and Evolution* 13 (3) (2001) 149-166.

[92] E. Tryggeseth, Report from an experiment: impact of documentation on maintenance, *Empirical Software Engineering* 2 (2) (1997) 201-207.

[93] K.G. van den Berg and P.M. van den Broek, Programmers' performance on structured versus nonstructured function definitions, *Information and Software Technology* 38 (7) (1996) 477-492.

[94] R. Vinter, M. Loomes, and D. Kornbrot, Applying software metrics to formal specifications: a cognitive approach, *5th IEEE International Software Metrics Symposium (METRICS)*, Maryland, USA, March 20-21 IEEE Computer Society (1998) 216-223.

[95] G. Visaggio, Assessing the maintenance process through replicated, controlled experiments, *The Journal of Systems and Software* 44 (3) (1999) 187-197.

[96] R.J. Walker, E.L.J. Baniassad, and G.C. Murphy, An initial assessment of aspect-oriented programming, *21st International Conference on Software Engineering (ICSE)*, Los Angeles, USA, May 16-22 ACM (1999) 120-130.

[97] L. Williams, R.R. Kessler, W. Cunningham, and R. Jeffries, Strengthening the case for pair programming, *IEEE Software* 17 (4) (2000) 19-25.

[98] C. Wohlin, Is prior knowledge of a programming language important for software quality?, *International Symposium on Empirical Software Engineering (ISESE)*, Nara, Japan, October 3-4 IEEE Computer Society (2002) 27-36.

[99] M.Y.M. Yen and R.W. Scamell, A human factors experimental comparison of SQL and QBE, *IEEE Transactions on Software Engineering* 19 (4) (1993) 390-409.

[100] C.S. Yoo and P.H. Seong, Experimental analysis of specification language diversity impact on NPP software diversity, *The Journal of Systems and Software* 62 (2) (2002) 111-122.

[101] A. Zendler, T. Pfeiffer, M. Eicks, and F. Lehner, Experimental comparison of coarse-grained concepts in UML, OML, and TOS, *The Journal of Systems and Software* 57 (1) (2001) 21-30.

[102] Z. Zhang, V. Basili, and B. Shneiderman, Perspective-based usability inspection: an empirical validation of efficacy, *Empirical Software Engineering* 4 (1) (1999) 43-69.

[103] S.H. Zweben, S.H. Edwards, B.W. Weide, and J.E. Hollingsworth, The effects of layering and encapsulation on software development cost and quality, *IEEE Transactions on Software Engineering* 21 (3) (1995) 200-208.

Appendix B. A preliminary systematic review of experiments published in 2007

B.1. Purpose. In order to assess whether the findings from the systematic review of experiments published in 1993-2002 are representative for contemporary practise, I performed a review of the experiments published in 2007.

B.2. Method. The review assessed the experiments published in 2007 in *Empirical Software Engineering* (EMSE), The *Journal of Systems and Software* (JSS), *IEEE Transactions on Software Engineering* (TSE), and *Information and software Technology* (IST). The results from this review are to be regarded as preliminary and a more thorough investigation will be performed later. A more thorough investigation will include independent review by several researchers both regarding extraction of articles and data gathering. In addition, all the variables reported in this thesis will be investigated, whereas this preliminary investigation only assessed a few.

In this preliminary investigation, the articles were selected by automatic search on the word "experiment" in the title, abstract and keywords in the journals' overviews of the articles. Then these articles were manually investigated to reveal whether they described an experiment according to the definition used in this thesis work, see section 4.4 in the summary.

B.3. Results. A total of 258 articles were published in the four journals (Table B.1). Among these, I found eight articles (3.1%) reporting 10 experiments [1-6, 8, 9]. Two articles [4, 6] reported two experiments. Another article described two experiments, which were analysed as one [8]. Hence, the article is regarded as reporting one experiment.

The extent of experiments found in these four journals in 2007 is quite similar to the average extent found for the same four journals in 1993-2002 (2.9%).

The findings from the review comprised the following:

- Hypothesis testing was performed for seven experiments; hence three experiments reported the results descriptively, only.
- Two experiments included professionals [2, 5]; seven included students.

Table B.1. Articles that report controlled experiments

	Review of articles in1993-2002			Review of articles in 2007		
Journal	Total no. of articles investigated	Articles reporting experiments		Total no. of articles investigated	Articles reporting experiments	
		N	Row %		N	Row %
EMSE	124	22	17.7	24	2	8.3
JSS	886	24	2.7	108	5	4.6
TSE	687	17	2.5	48	0	0
IST	745	8	1.1	78	1	1.3
All	2442	71	2.9	258	8	3.1

- The average number of participants was 32.4, the minimum number was nine and the maximum number was 128.
- Statistical power was reported for one of the seven experiments that performed hypothesis testing (14.3%) [9].
- Standardized effect size was not reported in any of the articles as part of experimental results. However, one experiment reported the observed standardized effect size in the discussion of statistical power [9].
- Unstandardized effect size was reported for three experiments (30.0%) [1, 4].
- Seven experiments described a randomization procedure (70.0%), one experiment used a self-selection assignment procedure (quasi-experiment) (10.0%) [3] and two experiments (20.0%) did not clearly describe whether a randomization procedure was performed or not. One of these [8] was apparently randomized, as described in another article [7]. The other experiment is probably a quasi-experiment, because a pretest score was used to divide the subjects into groups with as similar characteristics as possible [4].
- The quasi-experiment compared the experimental groups with respect to a pretest score in order to control for selection bias.
- None of the randomized experiment described the randomization procedure.
- The participants' background information was reported for seven experiments (70.0%):
    - Age, task related knowledge (course about software development and management) [1]
    - Task related experience (UML knowledge), work experience [2]

  - Age, sex, task related experience (programming experience in years and lines of code, course credits) [3]
  - Task related knowledge (knowledge and opinions) [4]
  - Gender [4]
  - Age, work experience, task related experience (project management) [5]
  - Task related experience (java experience in years and number of courses, experience in static analysis tools) [9]

  In addition, the participants' background information for one experiment [8] was reported in another paper:

  - Years of education, task related experience (java programming experience in loc and years) [7]

- Eight experiments reported *threats to validity/limitations* (80.0%). The two experiments that did not report any limitations did not perform hypothesis testing.

B.4. Conclusion:

- The reporting of statistical power and effect size is still unacceptably low.
- There are still needs for improvements regarding reporting of assignment procedures.
- The one quasi-experiment that was evaluated in this review controlled the experimental groups for a potential selection bias in the analysis. However, this is insufficient evidence to conclude that the SE community has improved regarding quasi-experimental design and analysis compared to research conducted in previous years.
- Background information is still reported in an unstandardized manner.

These preliminary findings indicates that there are little improvements regarding the quality of experimentation in SE, today, compared to the findings from the review of the experiments published in 1993-2002. Hence, the guidelines provided in this thesis are still relevant for current experimentation in software engineering.

References for the reviewed articles from 2007

[1] S. Abrahão and G. Poels, Experimental evaluation of an object-oriented function point measurement procedure, *Information and Software Technology* 49 (4) (2007) 366-380.

[2] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, and C.A. Visaggio, Evaluating performances of pair designing in industry, *Journal of Systems and Software* 80 (8) (2007) 1317-1327.

[3] A. Karahasanovic, A.K. Levine, and R. Thomas, Comprehension strategies and difficulties in maintaining object-oriented systems: An explorative study, *The Journal of Systems & Software* 80 (9) (2007) 1541-1559.

[4] L. Karlsson, T. Thelin, B. Regnell, P. Berander, and C. Wohlin, Pair-wise comparisons versus planning game partitioning—experiments on requirements prioritisation techniques, *Empirical Software Engineering* 12 (1) (2007) 3-33.

[5] M. Keil, L. Li, L. Mathiassen, and G. Zheng, The influence of checklists and roles on software practitioner risk perception and decision-making, *The Journal of Systems & Software* doi:10.1016/j.jss.2007.07.035 (2007)

[6] H. Liu and H.B.K. Tan, Testing input validation in Web applications through automated model recovery, *The Journal of Systems & Software* doi:10.1016/j.jss.2007.05.007 (2007)

[7] M.M. Muller, Two controlled experiments concerning the comparison of pair programmin to peer review, *The Journal of Systems & Software* 78 (2005) 166-179.

[8] M.M. Muller, Do Programmer pairs make different mistakes than solo programmers? *The Journal of Systems & Software* 80 (9) (2007) 1460-1471.

[9] M.A. Wojcicki and P. Strooper, Maximising the information gained from a study of static analysis technologies for concurrent software, *Empirical Software Engineering* 12 (6) (2007) 617-645.

References for the summary

[1] American Psychological Association (APA), Publication Manual of the American Psychological Association (4th ed.), 1994.

[2] L.S. Aiken, S.G. West, D.E. Schwalm, J.L. Carroll, and S. Hsiung, Comparison of a randomized and two quasi-experimental designs in a single outcome evaluation, *Evaluation Review* 22 (2) (1998) 207-244.

[3] D.G. Altman, K.F. Schulz, D. Moher, M. Egger, F. Davidoff, D. Elbourne, P.C. Gøtzsche, and T. Lang, The revised CONSORT statement for reporting randomized trials: explanation and elaboration, *Annals of Internal Medicine* 134 (8) (2001) 663-694.

[4] J.J. Baroudi and W.J. Orlikowski, The Problem of Statistical Power in MIS Research, *MIS Quarterly* 13 (1) (1989) 87-106.

[5] V. Basili, D. Rombach, K. Schneider, B. Kitchenham, D. Pfahl, and R. Selby, eds. Empirical Software Engineering Issues: Critical Assessment and Future Directions, *Proceedings from Int. Workshop, Dagstuhl Castle,* June 26-30, 2006, *Lecture Notes in Compute Science 4336*. Springer, 2007.

[6] V.R. Basili, The experimental paradigm in software engineering, in: H.D. Rombach, V.R. Basili, and R.W. Selby (Ed.), Experimental software engineering issues: critical assessment and future directions, *Proceedings from Int. Workshp,* Dagstuhl castle, Germany, September 14-18, 1992, *Lecture Notes in Computer Science 706*, Springer (1993) 3-12.

[7] V.R. Basili, The role of experimentation in software engineering: past, current and future *Proceedings of International Conference on Software Engineering (ICSE-18)*, Berlin, Germany, March 25-30 (1996) 442-449.

[8] V.R. Basili, R.W. Selby, and D.H. Hutchens, Experimentation in software engineering, *IEEE Transactions on Software Engineering* 12 (7) (1986) 733-743.

[9] V.R. Basili, F. Shull, and F. Lanubile, Building knowledge through families of experiments, *IEEE Transactions on Software Engineering* 25 (4) (1999) 456-473.

[10] G.K. Bhattacharyya and R.A. Johnson, *Statistical Concepts and Methods*, John Wiley & Sons, Inc., Singapore, 1977.

[11] K.S. Bordens and B.B. Abbott, *Research Design and Methods. A Process Approach*, McGraw-Hill, NeW York, 2008. Seventh Edition.

[12] S.C. Borkowski, M.J. Welsh, and Q. Zhang, An analysis of statistical power in behavioral accounting research, *Behavioral Research in Accounting* 13 (2001) 63–84.

[13] J.A. Breaugh, Effect size estimation: factors to consider and mistakes to avoid, *Journal of Management* 29 (1) (2003) 79-97.

[14] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, Lessons from applying the systematic literature review process within the software engineering domain, *The Journal of Systems & Software* 80 (4) (2007) 571-583.

[15] J.K. Brewer, On the power of statistical tests in the " American Educational Research Journal", *American Educational Research Journal* 9 (3) (1972) 391-401.

[16] J.K.-U. Brock, The 'power' of international business research, *Journal of International Business Studies* 34 (1) (2003) 90-99.

[17] D.T. Campbell, Factors relevant to the validity of experiments in social settings, *Psychological Bulletin* 54 (1957) 297-312.

[18] D.T. Campbell and J.C. Stanley, *Experimental and Quasi-Experimental Designs for Research*, Houghton Mifflin Company, Boston, 1963.

[19] L.H. Cashen and S.W. Geiger, Statistical power and the testing of null hypotheses: a review of contemporary management research and recommendations for future studies, *Organizational Research Methods* 7 (2) (2004) 151-167.

[20] T.C. Chalmers, P. Celano, H.S. Sacks, and H. Smith, Bias in treatment assignment in controlled clinical trials, *The New England Journal of Medicine* (1983).

[21] L.J. Chase and R.K. Tucker, A power-analytic examination of contemporary communication research, *Speech Monographs* 42 (3) (1975) 29-41.

[22] L.J. Chase and R.B. Chase, A statistical power analysis of applied psychological research, *Journal of Applied Psychology* 61 (2) (1976) 234-237.

[23] L.B. Christensen, *Experimental methodology*, Allyn & Bacon, 2006. 10th Edition.

[24] R. Christensen, *Analysis of Variance, Design and Regression - Applied Statistical Methods*, Chapman & Hall /CRC, USA, 1998. First Edition.

[25] D. Clark-Carter, The account taken of statistical power in research published in the British Journal of Psychology, *British Journal of Psychology* 88 (1997) 71-83.

[26] W.G. Cochran, Problems arising in the analysis of a series of similar experiments, *Journal of the Royal Statistical Society (Suppl.)* 4 (1937) 102-118.

[27] J. Cohen, The statistial power of abnormal-social psychological research: a review, *Journal of Abnormal and Social Psychology* 65 (3) (1962) 145-153.

[28] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, 1969. First Edition.

[29] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Lawrence Erlbaum Associates, 1988. Second Edition.

[30] J. Cohen, Things I have learned (so far), *American Psychologist* 45 (12) (1990) 1304-1312.

[31] J. Cohen, A power primer, *Psychological Bulletin* 112 (1) (1992) 155-159.

[32] G.A. Colditz, J.N. Miller, and F. Mosteller, How study design affects outcomes in comparisons of therapy. I: Medical, *Statistics in Medicine* 8 (1989) 441-454.

[33] T.D. Cook and D.T. Campbell, *Quasi-Experimentation: Design & Analysis Issues for Field Settings*, Houghton Mifflin, 1979.

[34] H. Cooper and L.V. Hedges, *The Handbook of Research Synthesis*, Russel Sage Foundation, New York, 1994.

[35] H.M. Cooper, On the significance of effects and the effects of significance, *Journal of Personality and Social Psychology* 41 (5) (1981) 1013-1018.

[36] B. Curtis, Measurement and experimentation in software engineering, *Proceedings of the IEEE* 68 (9) (1980) 1144-1157.

[37] T. Dybå, B.A. Kitchenham, and M. Jørgensen, Evidence-based software engineering for practitioners, *IEEE Software* 11 (1) (2005) 58-65.

[38] A. Endres and D. Rombach, *A Handbook of Software and Systems Engineering: Empirical Observations, Laws and Theories*, Pearson Education Ltd., London, 2003.

[39] N. Fenton, How effective are software engineering methods?, *Journal of Systems and Software* 22 (2) (1993) 141-146.

[40] N. Fenton, S.L. Pfleeger, and R.L. Glass, Science and substance: a challenge to software engineers, *IEEE Software* 1994 (July) (1994) 86-95.

[41] T.D. Ferguson and D.J. Ketchen Jr, Organizational configurations and performance: the role of statistical power in extant research, *Strategic Management Journal* 20 (4) (1999) 385-395.

[42] G.V. Glass, Primary, secondary, and meta-analysis of research, *Educational Researcher* 10 (1976) 3-8.

[43] R.L. Glass, I. Vessey, and V. Ramesh, Research in software engineering: an analysis of the literature, *Information & Software Technology* 44 (8) (2002) 491-506.

[44] E.E. Grant and H. Sackman, An exploratory investigation of programmer performance under on-line and off-line conditions, *IEEE Transactions on Human Factors in Electronics* 8 (1) (1967) 33-48.

[45] R.J. Grissom and J.J. Kim, *Effect Size for Research. A Broad Practical Approach*, Lawrence Erlbaum Associates, Inc., 2005.

[46] J.E. Hannay, D.I.K. Sjøberg, and T. Dybå, A systematic review of theory use in software engineering experiments, *IEEE Transactions on Software Engineering* 33 (2) (2007) 87-107.

[47] A.D. Harris, E. Lautenbach, and E. Perencevich, A systematic review of quasi-experimental study designs in the fields of infection control and antibiotic resistance, *Antimicrobial Resistance* 41 (1 July) (2005) 77-82.

[48] A.D. Harris, D.D. Bradham, M. Baumgarten, I.H. Zuckerman, J.C. Fink, and E.N. Perencevich, The use and interpretation of quasi-experimental studies in infectious diseases, *Antimicrobial Resistance* 38 (1 June) (2004) 1586-1591.

[49] A.D. Harris, J.C. McGregor, E.N. Perencevich, J.P. Furuno, J. Zhu, D.E. Peterson, and J. Finkelstein, The use and interpretation of quasi-experimental studies in medical informatics, *Journal of the American Medical Informatics Association* 13 (2006) 16-23.

[50] L.V. Hedges and I. Olkin, *Statistical Methods for Meta-Analysis*, Academic Press, Inc., 1985.

[51] D.T. Heinsman, *Effect Sizes in Meta-Analysis: Does Random Assignment Make a Difference?* Doctoral Thesis, 1993, Memphis State University.

[52] D.T. Heinsmann and W.R. Shadish, Assignment methods in experimentation: When do nonrandomized experiments approximate answers from randomized experiments?, *Psychological Methods* 1 (2) (1996) 154-169.

[53] C.R. Hill and B. Thompson, Computing and interpreting effect sizes, in: J.C. Smart (Ed.), *Higher Education: Handbook of Theory and Research*, Kluwer Academic Publishers, (2004) 175-196.

[54] F. Houdek, External experiments - a workable paradigm for collaboration between industry and academia, in: N. Juristo and A.M. Moreno (Ed.), *Lecture Notes on Empirical Software Engineering*, World Scientific Publishing Singapore, (2003).

[55] A. Jedlitschka and D. Pfahl, Reporting guidelines for controlled experiments in 5oftware engineering, *International Symposium on Empirical Software Engineering (ISESE)*, Noosa Heads, Australia, November 17-18 (2005) 92-101.

[56] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, Reporting experiments in software engineering, in: F. Shull, J. Singer, and D.I.K. Sjøberg (Ed.), *Advanced Topics in Empirical Software Engineering (forthcoming)*, Springer, (2008).

[57] N. Juristo and A.M. Moreno, *Basics of Software Engineering Experimentation*, Kluwer Academic Publishers, Boston, 2003.

[58] M. Jørgensen and D. Sjøberg, Generalization and theory-building in software engineering research, *Empirical Assessment in Software Engineering* Edinburgh, Scotlans, May 24-25 IEE Proceedings (2004) 29-36.

[59] M. Jørgensen, T. Dybå, and B.A. Kitchenham, Teaching evidence-based software engineering to university students, *11th IEEE International Software Metrics Symposium*, Como, Italy, September 19-22 (2005).

[60] H.J. Keselman, C.J. Huberty, L.M. Lix, S. Olejnik, R.A. Cribbie, B. Donahue, R.K. Kowalchuk, L.L. Lowman, M.D. Petosky, J.C. Keselman, and J.R. Levin,

Statistical practices of educational researchers: an analysis of their ANOVA, MANOVA, and ANCOVA analyses, *Review of Educational Research* 68 (3) (1998) 350-386.

[61] R.E. Kirk, Practical significance: a concept whose time has come, *Educational and Psychological Measurement* 56 (5) (1996) 746-759.

[62] B. Kitchenham, Procedures for performing systematic reviews, *Keele University, UK, Technical Report TR/SE-0401 and National ICT Australia, Technical Report 0400011T.1.* (2004).

[63] B. Kitchenham, H. Al-Khilidar, M.A. Babar, M. Berry, K. Cox, J. Keung, F. Kurniawati, M. Staples, H. Zang, and L. Zhu, Evaluating guidelines for empirical software engineering studies, *5th IEEE International Symposium on Empirical Software Engineering (ISESE)*, Rio de Janeiro, Brazil, September 21-22 IEEE Computer Society (2006).

[64] B.A. Kitchenham, T. Dybå, and M. Jørgensen, Evidence-based software engineering, *International Conference on Software Engineering*, Edinburgh, Scotland, 23-28 May IEEE Computer Society (2004) 273-281.

[65] B.A. Kitchenham, S.G. Linkman, and J.S. Fry, The impact of human experimenters and human subjects on empirical studies, , *Keele University, UK, Technical Report 0400013T.1 and National ICT Australia, Technical Report 0400013T.1* (2004).

[66] B.A. Kitchenham, S.L. Pfleeger, L.M. Pickard, P.W. Jones, D.C. Hoaglin, K. ElEmam, and J. Rosenberg, Preliminary guidelines for empirical research in software engineering, *IEEE Transactions on Software Engineering* 28 (8) (2002) 721-734.

[67] R.B. Kline, *Beyond Significance Testing. Reforming Data Analysis Methods in Behavioral Research*, American Psychological Association, Washington, DC, 2004.

[68] H.C. Kraemer and S. Thiemann, *How Many Subjects? Statistical Power Analysis in Research*, Sage, Newbury Park, CA, 1987.

[69] O. Laitenberger and D. Rombach, (Quasi-)experimental studies in Industrial setting, in: N. Juristo and A.M. Moreno (Ed.), *Series on Software Engineering and Knowledge Engineering (12), Lecture Notes on Empirical Software Engineering*, World Scientific Singapore, (2003) 167-227.

[70] Leslie L. Roos Jr, Quasi-experiments and environmental policy, *Policy Science* 6 (1975) 249-265.

[71] M.W. Lipsey, *Design Sensitivity: Statistical Power for Experimental Research*, Sage, Newbury Park, CA, 1990.

[72] M.W. Lipsey and D.B. Wilson, *Practical Meta-Analysis*, Sage, Thousand Oaks, 2001.

[73] C. Lott and D. Rombach, Repeatable software engineerng experiments for comparing defect-detection techniques, *Empirical Software Engineering* 1 (3) (1996) 241-277.

[74] A.M. Mazen, L.A. Graf, C.E. Kellogg, and M. Hemmasi, Statistical power in contemporary management research, *The Academy of Management Journal* 30 (2) (1987) 369-380.

[75] J.R. McKay, A.I. Alterman, A.T. McLellan, C.R. Boardman, F.D. Mulvaney, and C.P. O'Brien, Random versus nonrandom assignment in the evaluation of treatment for cocaine abusers, *Journal of Consulting and Clinical Psychology* 6 (4) (1998) 697-701.

[76] B.D. Meyer, Natural and quasi-experiments in economics, *Technical Working Paper no. 170,* National Bureau of Economic Research, Cambridge, MA (1994).

[77] J. Miller, Applying meta-analytical procedures to software engineering experiments, *Journal of Systems and Software* 54 (2000) 29-39.

[78] J. Miller, Statistical significance testing - a panacea for software technology experiments?, *Journal of Systems and Software* 73 (2004) 183-192.

[79] J. Miller, Replicating software engineering experiments: a poisoned chalice or the Holy Grail, *Information and Software Technology* 47 (2005) 233-244.

[80] J. Miller, J. Daly, M. Wood, M. Roper, and A. Brooks, Statistical power and its subcomponents - missing and misunderstood concepts in empirical software engineering research, *Information and Software Technology* 39 (1997) 285-295.

[81] J.N. Miller, G.A. Colditz, and F. Mosteller, How study design affects outcomes in comparisons of therapy. II: Surgical, *Statistics in Medicine* 8 (1989) 455-466.

[82] T. Moher and G.M. Schneider, Methods for improving controlled experimentation in software engineering, *IEEE* (1981) 224-233.

[83] T. Moher and G.M. Schneider, Methodology and experimental research in software engineering, *International Journal of Man-Machine Studies* 16 (1982) 65-87.

[84] M.A. Mone, G.C. Mueller, and W. Mauland, The perceptions and usage of statistical power in applied psychology and management research, *Personnel Psychology* 49 (1) (1996) 103-120.

[85] D.C. Montgomery, *Design and Analysis of Experiments*, John Wiley & Sons, Inc, 2001, 5th ed.

[86] S.B. Morris and R.P. DeShon, Correcting effect size computed from factorial analysis of variance for use in meta-analysis, *Psychological Methods* 2 (2) (1997) 192-199.

[87] S. Olejnik and J. Algina, Generalized eta and omega squared statistics: measures of effect size for some common research designs, *Psychological Methods* 8 (4) (2003) 434-447.

[88] D.E. Perry, A.A. Porter, and L.G. Votta, Empirical studies of software engineering: a roadmap, *International Conference on Software Engineering. Proceedings of the Conferance on The Future of Software Engineering*, Limerick, Ireland, ACM Press (2000) 345-355.

[89] S.L. Pfleeger, Design and analysis in software engineering. Part 1: The language of case studies and formal experiments, *ACM Sigsoft Software Engineering Notes,* ACM Press 19 (4) (1994) 16-20.

[90] S.L. Pfleeger, Design and analysis in software engineering. Part 2: How to set up an experiment, *ACM Sigsoft Software Engineering Notes,* ACM Press 20 (1) (1995) 22-26.

[91] S.L. Pfleeger, Design and analysis in software engineering. Part 5: Analysing the data, *ACM Sigsoft Software Engineering Notes,* ACM Press 20 (5) (1995) 14-17.

[92] S.L. Pfleeger, Design and analysis in software engineering. Part 3: Types of experimental design, *ACM Sigsoft Software Engineering Notes,* ACM Press 20 (2) (1995) 14-16.

[93] S.L. Pfleeger, Design and analysis in software engineering. Part 4: Choosing an experimental design, *ACM Sigsoft Software Engineering Notes,* ACM Press 20 (3) (1995) 13-16.

[94] C. Potts, Software-engineering research revisited, *IEEE Software* 10 (5) (1993) 19-28.

[95] R.A. Rademacher, Statistical power in information system research: application and impact on the discipline, *Journal of Computer Information Systems* 39 (4) (1999) 1-7.

[96] R. Rosenthal, Effect sizes in behavioral and biomedical research: estimation and interpretation, in: L. Bickman (Ed.), *Validity & Social Experimentation: Donald Campbell's Legacy*, Sage, Thousand Oaks, CA, (2000) 121-139.

[97] R. Rosenthal and D.B. Rubin, A simple, general purpose display of magnitude of experimental effect, *Journal of Educational Psychology* 74 (2) (1982) 166-169.

[98] R. Rosenthal and D.B. Rubin, The counternull value of an effect size: a new statistic, *Psychological Science* 5 (6) (1994) 329-334.

[99] R. Rosenthal, R.L. Rosnow, and D.B. Rubin, *Contrasts and Effect Sizes in Behavioral Research. A Correlational Approach*, Cambridge University Press, 2000.

[100] A.G. Sawyer and A.D. Ball, Statistical power and effect size in marketing research, *Journal of Marketing Research* 18 (3) (1981) 275-290.

[101] L. Sechrest and W.H. Yeaton, Empirical bases for estimating effect size, in: R.F. Boruch, P.M. Wortman, and D.S. Cordray (Ed.), *Reanalyzing Program Evaluations*, Jossey-Bass, San Francisco, (1981).

[102] P. Sedlmeier and G. Gigerenzer, Do studies of statistical power have an effect on the power of studies?, *Psychological Bulletin* 105 (2) (1989) 309-316.

[103] J. Segal, A. Grinyer, and H. Sharp, The type of evidence produced by empirical software engineers, *Proceedings of the Workshop on Realising Evidence-Based Software Engineering*, St. Louis, Missouri, USA, May 17 (2005) 1-4.

[104] W.R. Shadish, The empirical program of quasi-experimentation, in: L. Bickman (Ed.), *Reseach Design: Donald Campbell's Legacy*, Sage, Thousand Oaks, CA, (2000) 13-35.

[105] W.R. Shadish and K. Ragsdale, Random versus nonrandom assignment in controlled experiments: Do you get the same answer?, *Journal of Consulting and Clinical Psychology* 64 (6) (1996) 1290-1305.

[106] W.R. Shadish, T.D. Cook, and D.T. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*, Houghton Mifflin, Boston, 2002.

[107] D.A. Shapiro and D. Shapiro, Meta-analysis of comparative therapy outcome studies: a replication and refinement, *Psychological Bulletin* 92 (3) (1982) 581-604.

[108] M. Shaw, Writing good software engineering research papers, *25th International Conference on Software Engineering*, Portland, Oregon, IEEE Computer Society (2003) 726-736.

[109] F. Shull, J. Singer, and D.I.K. Sjøberg, eds. *Guide to Advanced Empirical Software Engineering (forthcoming)* Springer, 2008.

[110] F. Shull, V. Basili, J. Carver, J.C. Maldonado, G.H. Travassos, M. Mendonca, and S. Fabbri, Replicating software engineering experiments: addressing the tacit knowledge problem, *International Symposium on Empirical Software Engineering*, Nara, Japan, October 3-4 IEEE Computer Society (2002) 7-16.

[111] F. Shull, M.G. Mendonca, V. Basili, J. Carver, J.C. Maldonado, S. Fabbri, G.H. Travassos, and M.C. Ferreira, Knowledge-sharing issues in experimental software engineering, *Empirical Software Engineering* 9 (2004) 111-137.

[112] J. Singer, Using the american psychological association (APA) style guidelines to report experimental results, *Proceedings of the Workshop on Empirical Studies in Software Maintenance*, Oxford, England, (1999) 71-75.

[113] D.I.K. Sjøberg, T. Dybå, and M. Jørgensen, The future of empirical methods in software engineering research, in: L. Briand and A. Wolf (Ed.), *Future of Software Engineering* IEEE Computer Society, (2007) 358-378.

[114] D.I.K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanovic, E. Koren, and M. Vokac, Conducting realistic experiments in software engineering *International Symposium on Empirical Software Engineering*, Nara, Japan, IEEE Computer Society (2002) 17-26.

[115] N.J. Smelser and P.B. Baltes, eds. *International Encyclopedia of the Social & Behavioral Sciences*. Elsevier Science Ltd., Oxford, UK 2001.

[116] M.L. Smith, G.V. Glass, and T.I. Miller, *The Benefits of Psychotherapy*, The Johns Hopkins University Press, USA, 1980.

[117] B. Thompson, "Statistical", "Practical", and "Clinical": How many kinds of significance do counselors need to consider?, *Journal of Counseling & Development* 80 (2002) 64-71.

[118] B. Thompson and P.A. Snyder, Statistical significance and reliability analyses in recent Journal of Counseling & Development research articles, *Journal of Counseling & Development* 76 (4) (1998) 436-41.

[119] W.L. Thompson, 402 citations questioning the indiscriminate use of null hypothesis significance tests in observational studies, *Retrieved July 11, 2007, from*

http://biology.uark.edu/coop/courses/thompson5.html (2001).

[120] W.F. Tichy, Should computer scientists experiment more?, *Computer* 31 (5) (1998) 32-40.

[121] W.F. Tichy, P. Lukowicz, L. Prechelt, and E.A. Heinz, Experimental evaluation in computer science: a quantitative study, *Journal of Systems and Software* 28 (1) (1995) 9-18.

[122] J. Trusty, B. Thompson, and J.V. Petrocelli, Practical guide for reporting effect size in quantitative research in the Journal of Counseling & Development, *Journal of Counseling & Development* 82 (2004) 107-110.

[123] D. Weisburd, C.M. Lum, and A. Petrosino, Does research design affect study outcomes in criminal justice?, *Annals of the American Academy of Political and Social Science* 578 (2001) 50-70.

[124] L. Wilkinson and the Task Force on Statistical Inference, Statistical methods in psychology journals: guidelines and explanations, *American Psychologist* 54 (8) (1999) 594-604.

[125] D.B. Wilson and M.W. Lipsey, The role of method in treatment effectiveness research: evidence from meta-analysis, *Psychological Methods* 6 (4) (2001) 413-429.

[126] C. Wohlin, P. Runeson, M. Høst, M.C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in software engineering: an introduction*, Kluwer Academic Publishers, 1999.

[127] C. Zannier, G. Melnik, and F. Maurer, On the success of empirical studies in the international conference on software engineering, *International Conference on Software Engineering*, Shanghai, China, May 20-28 ACM Press (2006) 341-350.

[128] M.V. Zelkowitz and D. Wallace, Experimental validation in software engineering, *Information and Software Technology* 39 (11) (1997) 735-743.

[129] A. Zendler, A preliminary software engineering theory as investigated by published experiments, *Empirical Software Engineering* 6 (2001) 161-180.