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

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.

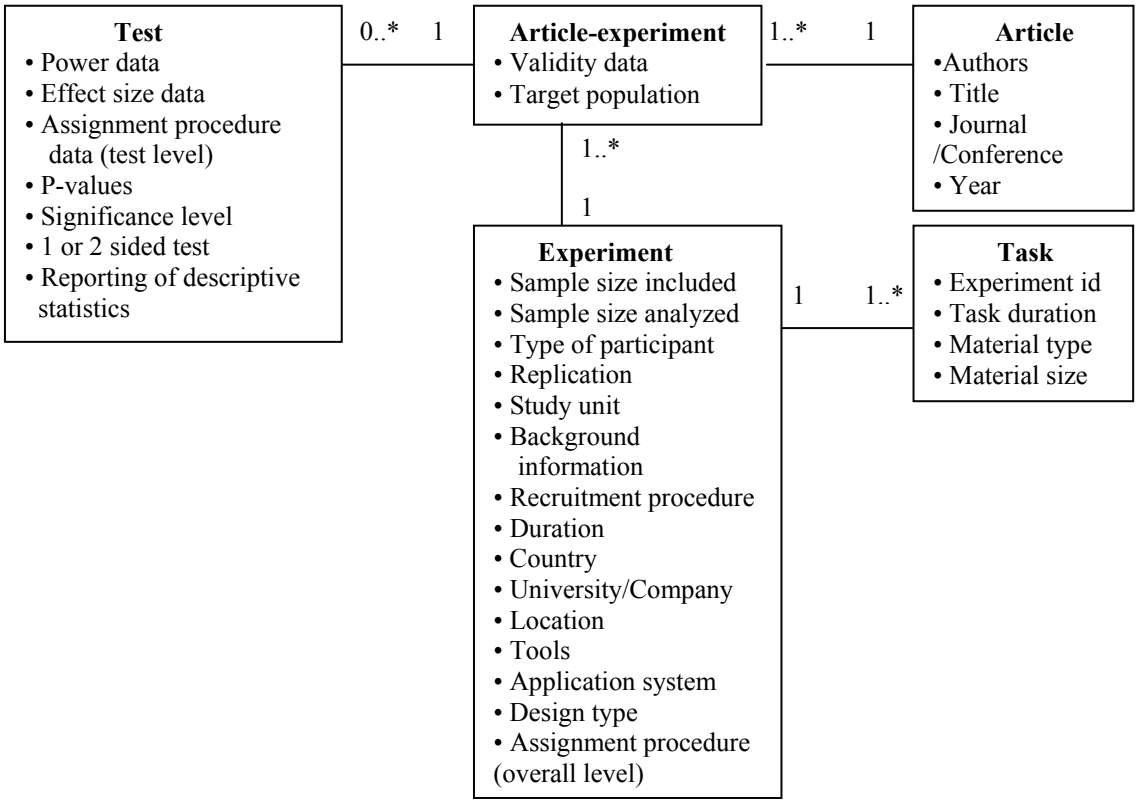

Figure 1. The data model for the review

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

Journal	Review of articles in1993-2002			Review of articles in 2007		
	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.

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.

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.

Keywords: Controlled experiments, survey, research methodology, empirical software engineering.

1 Introduction

There is an increasing understanding in the software engineering community that empirical studies are needed to develop or improve processes, methods and tools for software development and maintenance [6, 4, 35, 16, 43, 5, 32, 41, 50, 15]. An important category of empirical study is that of the controlled experiment, the conducting of which is the classical scientific method for identifying cause-effect relationships.

This paper reports on a survey that quantitatively characterises the controlled experiments in software engineering published in nine journals and three conference proceedings in the decade from 1993 to 2002. The journals are ACM Transaction on Software Engineering Methodology (TOSEM), Empirical Software Engineering (EMSE), IEEE Computer, IEEE Software, IEEE Transactions on Software Engineering (TSE), Information and Software Technology (IST), Journal of Systems and Software (JSS), Software Maintenance and Evolution (SME), Software: Practice and Experience (SP&E). The conferences are the International Conference on Software Engineering (ICSE), IEEE International Symposium on Empirical Software Engineering (ISESE), and IEEE International Symposium on Software Metrics (METRICS). The conference Empirical Assessment & Evaluation in Software Engineering (EASE) is partially included in that ten selected articles from EASE appear in special issues of JSS, ESE, and IST. We consider the above journals to be leaders in software engineering. 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.

Research in empirical software engineering should aim to acquire general knowledge about which *technology* (process, method, technique, language or tool) is useful for *whom* to conduct which (software engineering) *tasks* in which *environments*. Hence, this survey focuses on the kind of technology being studied in the experiments investigated (which reflects the topics of the experiments), the subjects that took part, the tasks they performed, the type of application systems on which these tasks were performed, and the environments in which the experiments were conducted. This survey also includes data on experiment replication and the extent to which internal and external validity is discussed.

The paper is organised as follows. Section 2 describes related work. Section 3 defines the research method for the survey. Section 4 reports the extent of controlled experiments,

and Sections 5–10 report our main findings. Section 11 discusses threats to validity of this survey. Section 12 summarises.

2 Related Work

Table 1 summarises the purpose, scope and extent of four major surveys in addition to this survey. Tichy *et al.* [43] compare the amount of experimental work published in a few computer science journals and conference proceedings with the amount of experimental work published in a journal on artificial neural network and a journal on optical engineering. In total, 403 articles were surveyed and classified into five categories: *formal theory*, *design and modeling*, *empirical work*, *hypothesis testing* and *other*. Zelkowitz and Wallace [49] propose a taxonomy of empirical studies in software engineering and report a survey in which 612 articles were classified within this taxonomy. Glass *et al.* [20] investigate 369 articles with respect to topic, research approach, research method, reference discipline and level of analysis.

The above surveys give a comprehensive picture of research methods used in software engineering. They differ in purpose, selection criteria and taxonomies. Nevertheless, their results suggest the same conclusions: the majority of published articles in computer science and software engineering provide little or no empirical validation, and the proportion of controlled experiments is particularly low. The surveys propose means to increase the amount of empirical studies and their quality.

The major difference between those surveys and ours is that they describe the extent and characteristics of various types of empirical study, while we provide an in-depth study of controlled experiments only. A comparison of those surveys and ours regarding the extent of controlled experiments is provided in Section 4.

In addition to the general surveys described above, there are several surveys within subdisciplines of software engineering, for example, object-oriented technology [14], testing techniques [28] and software effort estimation [25]. Furthermore, Shaw [38] categorises the research reported in articles submitted and accepted for ICSE 2002, and Zendler [51] reports a survey of 31 experiments with the aim of developing a preliminary theory about software engineering.

Table 1. Surveys of empirical studies in software engineering

	(Tichy *et al*. 1995)	(Zelkowitz *et al*. 1997)	(Glass *et al.* 2002)	(Zendler 2001)	Our survey
Purpose	Compares the extent of empirical studies in computer science with other fields	Classifies empirical studies in SE and validates the taxonomy of empirical studies proposed by the authors	Surveys topics, research approaches, research methods, reference disciplines and level of analysis.	Develops a preliminary SE theory from the results of various SE experiments	Surveys topics, subjects, tasks, environments, and internal and external validity of controlled experiments in SE
Scope	Comp. Sci., incl. SE	SE	SE	SE	SE
Journals	ACM (random publications), TSE, PLDI Proc., TOCS, TOPLAS	ICSE Proc., IEEE Software, TSE	IEEE Software, IST, JSS, SP&E, TOSEM, TSE	Various journals and conference proceedings	EASE, EMSE, ICSE, IEEE Computer, IEEE Software, ISESE, IST, JSME, JSS, METRICS, SP&E, TOSEM, TSE
Sampling of papers	1991-1994, one to four volumes per journal, random selection of work published by ACM in 1993	All papers in 1985, 1990 and 1995	Every fifth paper in the period 1995-1999	Not reported	All papers in the period 1993-2002
Number of investigated papers	403	612	369	49 papers assessed, 31 papers analysed in depth	5453 papers scanned, 103 papers analysed in depth

3 Research Method

This section describes the kind of experiments that are considered in this survey, and the procedure for identifying and analysing the relevant articles.

3.1 Controlled experiments in software engineering

Shadish *et al.* [37] provide the following definitions:

- *Experiment*: A study in which an intervention is deliberately introduced to observe its effects.
- *Randomised experiment*: An experiment in which units are assigned to receive the treatment or an alternative condition by a random process such as the toss of a coin or a table of random numbers.
- *Quasi-Experiment*: An experiment in which units are not assigned to conditions randomly.

- *Correlation study*: Usually synonymous with non-experimental or observational study; a study that simply observes the size and direction of a relationship among variables.

To identify the effects of the deliberate intervention in an experiment, factors that may influence the outcome, in addition to the treatment, should also be controlled[1]. This is the challenge of internal validity (see Section 10.1). Note that control is not an all or nothing condition; the degree of control varies on a continuous scale. Based on the definitions given above, we present an operational definition used for this survey. Since the term 'experiment' is inconsistently used in the software engineering community (often used synonymously with empirical study), we use the term 'controlled experiment':

Controlled experiment in software engineering (operational definition):
A randomised 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).

We do not distinguish between randomised experiments and quasi-experiments in this survey, because both experimental designs are relevant to empirical software engineering experimentation. Random assignment of experimental units to treatments may not always be feasible, *e.g.,* for logistic reasons. For example, one of the surveyed experiments used units formed from existing training groups in a company – random assignment would, in this case, have disturbed the training process.

We exclude several types of study that share certain characteristics with controlled experiments, because while these may be highly relevant for the field, they do not apply the deliberate intervention essential to controlled experiments. Thus, we exclude correlation studies, studies that are solely based on calculations on existing data (*e.g.,* from data mining), and evaluations of simulated teams based on data for individuals. The last category falls outside our operational definition, because the units are constructed after the run of the experiment.

[1] Some definitions are very explicit on the aspect of control, for example, Zimney [52] defines a psychological experiment as "objective observation of phenomena which are made to occur in a strictly controlled situation in which one or more factors are varied and the others are kept constant", see discussion of this definition in [10].

Studies that use projects or companies as treatment groups, in which data is collected at several levels (treatment defined, but no experimental unit defined) are also excluded because we consider these to be multiple case studies [47] (even though the authors might refer to them as experiments). Our survey focuses on articles that provide the main reporting of experiments. This excludes articles that at the outset would not provide sufficient data for our analyses (*e.g.,* summaries of research programs). Moreover, usability experiments are not included since we regard those as part of another discipline (human computer interaction).

3.2 Identification of articles that report controlled experiments

In order to identify and extract controlled experiments, one researcher systematically read the titles and abstracts of 5453 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. In the end, 103 articles were selected. 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.

3.3 Analysis of the articles

The survey data is stored in a relational database (MS SQL Server 2000)[2]. Some data is specific to an article, some is specific to an experiment and some information concerns the combination of article and experiment. Moreover, an article might describe several experiments and an experiment might be described in several articles, typically with a different focus in each article. Consequently, we defined a data model with the entities *article*, *experiment* and *article-experiment* with a set of attributes relevant to our survey. In addition to the survey database, a catalogue of all the articles in searchable pdf-format was generated. (About 3/4 of the articles were provided in searchable pdf-format by the journal

[2] MS SQL Server 2000 is a registered trademark of Microsoft Corp.

publishers; the remaining 1/4 were OCR-scanned.)[3] The articles were analysed according to the six aspects listed in Table 2. Each aspect encompasses a set of attributes for data extraction.

Six researchers analysed the articles, so that each aspect above 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 over the article to a third person. The main analysis tool was SAS[4].

Table 2. Aspects and their attributes for data extractions

Aspect	Attributes
Extent	Authors, Affiliation, Country, Year, Journal/Conference.
Topic	Treatment, Title, Keywords.
Subjects	Number of subjects categorised into (subcategories of) students and professionals, Subject selection mode, Subject background and Subject recruitment information (voluntary, part of course, paid, *etc.*).
Task and Environment	Location of experiment, Development tool, Work mode (individual or team), Duration, Application type (commercial or constructed), Application/Task size.
Replication	Replication indicator, *Subjects*, *Topic, Extent.*
Internal validity	Category of threat to internal validity, Explicitness.
External validity	Category of threat to external validity, Explicitness.

4 Extent

Controlled experiments, as defined in Section 3.1, are reported in 103 (1.9%) of the 5453 articles scanned for this survey, see Table 3. The 103 articles report a total of 113 controlled experiments. Twelve articles report more than one experiment and four experiments are reported in several articles.

EMSE, ISESE and METRICS, which focus specifically on empirical software engineering, report a higher proportion of controlled experiments than the other journals and the ICSE conference. The mean proportion of controlled experiments across years varies between 0.6 and 3.5, but we see no marked trend over years. An overview of the trend for the individual journals/conferences is presented in the appendix.

[3] The survey database and catalogue of articles may be provided upon request to the corresponding author and under the conditions of a signed agreement towards the use of the data.

[4] SAS is a registered trademark of SAS Institute Inc.

Table 3. Articles reporting controlled experiments

		Articles reporting controlled experiments	
Journal/Conference	Total no. of articles investigated	N	Row %
EMSE	124	22	17.7
ISESE	20	3	15.0
METRICS	177	10	5.6
JSS	886	24	2.7
TSE	687	17	2.5
ICSE	520	12	2.3
IST	745	8	1.1
SME	186	2	1.1
IEEE SW	532	4	0.8
TOSEM	125	1	0.8
IEEE Comp	780	0	0
SP&E	671	0	0
All	5453	103	1.9

The surveys summarised in Table 1 also report extent. Tichy *et al.* have the study type definition with the broadest scope and report that 14% of the articles published in the specific software engineering journals TSE and TOPLAS (*Transactions on Programming Languages and Systems*) describe empirical work. In Glass *et al.*, the authors classify 3% of the articles as laboratory experiments using human subjects and <1% as field experiments. According to the survey by Zelkowitz and Wallace, experiments defined as controlled methods are reported in 2.9% of the articles. Our survey finds the lowest percentage of articles (1.9%) that report controlled experiments. This might be because our study type definition is narrower than those of the other studies or because our investigation spans more sources and years.

We rank institutions and scholars according to the number of *experiments* published (not the quality), but relative to their fractional representation on the article(s) that reports the experiment. Glass and Chen [18] also ranked institutions and scholars but according to *publication* in systems and software engineering, and they used a more complex ranking scheme for scholars.

In total, 207 scholars are involved in the experiments of our survey. Table 4 presents the top 20 ranked scholars. Due to the fractional distribution, the number of experiments in which a scholar has actually been involved, is typically higher than the scores in Table 4. For instance, the top ranked scholar, Giuseppe Visaggio, was involved in six experiments described in four papers authored by one to three scholars, resulting in a fractional score of 4.2 experiments. Among the 20 top ranked scholars, three (Laitenberger, Roper, Wood) were involved in eight experiments, one was involved in seven, four in six, two in five, nine in four, and one was involved in three experiments.

There are 109 institutions from 19 countries involved in the experiments of our survey. The scores for institutions are accumulated from the scores of affiliated authors. Table 5 presents the top 10 ranked institutions.

The institution that has used most professionals as subjects throughout the surveyed time period is Fraunhofer Institute, Kaiserslautern. In total, they used 100 professionals in six experiments, ranging from 11 to 20 in a single experiment. The institution that conducted the experiment involving the largest number (68) of professionals as subjects was Andersen Consulting (now Accenture), Norway.

Table 4. Top 20 scholars conducting controlled experiments in software engineering 1993-2002

Rank	Experiments	Scholar	Affiliation
1	4.2	Visaggio G	Dipartimento di Informatica, University of Bari
2	2.7	Prechelt L	abaXX Technology AG; Fakultät für Informatik, Universität Karlsruhe
3	2.6	Laitenberger O	Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern
3	2.6	Porter A A	Department of Computer Science, University of Maryland
5	2.4	Wohlin C	Dept. of SE and Comp. Sci., Blekinge Inst. of Technology; Dept. of Com. Systems, Lund University
6	2.3	Roper M	Department of Computer Science, University of Strathclyde
6	2.3	Wood M	Department of Computer and Information Sciences, University of Strathclyde
8	2.0	Votta L G	Software Production Research Department, AT&T Bell Laboratories/Lucent Technologies
8	2.0	Koskinen J	Department of Computer Science and Information Systems, University of Jyväskylä
10	1.8	Miller J	Department of Computer Science, University of Strathclyde
10	1.8	Jørgensen M	Department of Informatics, University of Oslo; Simula Research Laboratory, Oslo
10	1.8	Sjøberg D	Department of Informatics, University of Oslo; Simula Research Laboratory, Oslo
13	1.3	El Emam K	Canadian National Research Council, Institute for Information Technology
13	1.3	Regnell B	Department of Communication Systems, Lund University
13	1.3	Höst M	Department of Communication Systems, Lund University
16	1.2	Daly J W	Agilent Technologies, Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern
16	1.2	Tichy W F	Fakultät für Informatik, Universität Karlsruhe
16	1.2	Unger B	sd&m GmbH and Co.; Fakultät für Informatik, Universität Karlsruhe
19	1.1	Basili V R	Department of Computer Science, University of Maryland
19	1.1	Lanubile F	Dipartimento di Informatica, University of Bari
…	…	…	…
Total	113	Total number of scholars	207

Table 5. Top 10 institutions conducting controlled experiments in software engineering 1993-2002

Rank	Experiments	Institution			Country
1	8.7	Department of Computer and Information Sciences, University of Strathclyde			Scotland
2	7.6	Fraunhofer Institute for Experimental Software Engineering, Kaiserslautern			Germany
3	6.3	Department of Communication Systems, Lund University			Sweden
4	6.2	Department of Computer Science, University of Maryland			USA
5	5.2	Dipartimento di Informatica, University of Bari			Italy
6	4.1	Fakultät für Informatik, Universität Karlsruhe			Germany
7	4.0	Department of Informatics, University of Oslo			Norway
8	2.3	Department of Computer and Information Science, The Ohio State University			USA
9	2.1	Software Production Research Dept., AT&T Bell Labs/Lucent Technologies			USA
10	2.0	Cleveland State University			USA
…	…	…			…
Total	113	Total number of institutions	109	Total number of countries	19

5 Topics

This section describes two classifications of the 103 analysed articles according to their main topic. The first classification illustrates the experiments' discipline coverage relative to software engineering as a whole, while the second classification has a more technical focus on software engineering method and methodology. The analysis is with respect to article, rather than experiment, this is adequate since no two experiments on different topics are reported in the same article. Both classifications emphasise the treatment of an experiment, since treatment, being the intervention of interest (Section 3) indicates the de facto topic under investigation.

5.1 Classification scheme: Glass *et al.*

There are a number of classification schemes for computing science and software engineering, *e.g.,* SWEBOK [1] and Glass *et al.* [20]. The classification scheme of Glass *et al.* is aimed at positioning software engineering research relative to a backdrop of overall computing disciplines, *i.e.,* computer science, software engineering, and information systems, and their classification categories are meant to give uniformity across all three fields [19]. The scheme is, therefore, somewhat general. On the other hand, this scheme has actually been used in classifying work undertaken in software engineering, and can therefore be used for illustrating the relative topic coverage of controlled experiments in software engineering.

Fig. 1 shows the distribution to topic categories of controlled experiments in software engineering relative to software engineering research in general. Controlled experiments seem to cover at least the categories that are well represented by general SE research, but remember that the overall number of controlled experiments performed is low (Section 2). Recall that experiments on topics purely within human computer interaction are not included in this survey, as is the case for topics purely within information systems. Our focus on experiments with human subjects also excludes a range of software engineering topics.

The two most prominent categories are Software life-cycle/engineering (49%) and Methods/Techniques (32%) due to respectively, the relatively large number of experiments on inspection techniques and object-oriented design techniques.

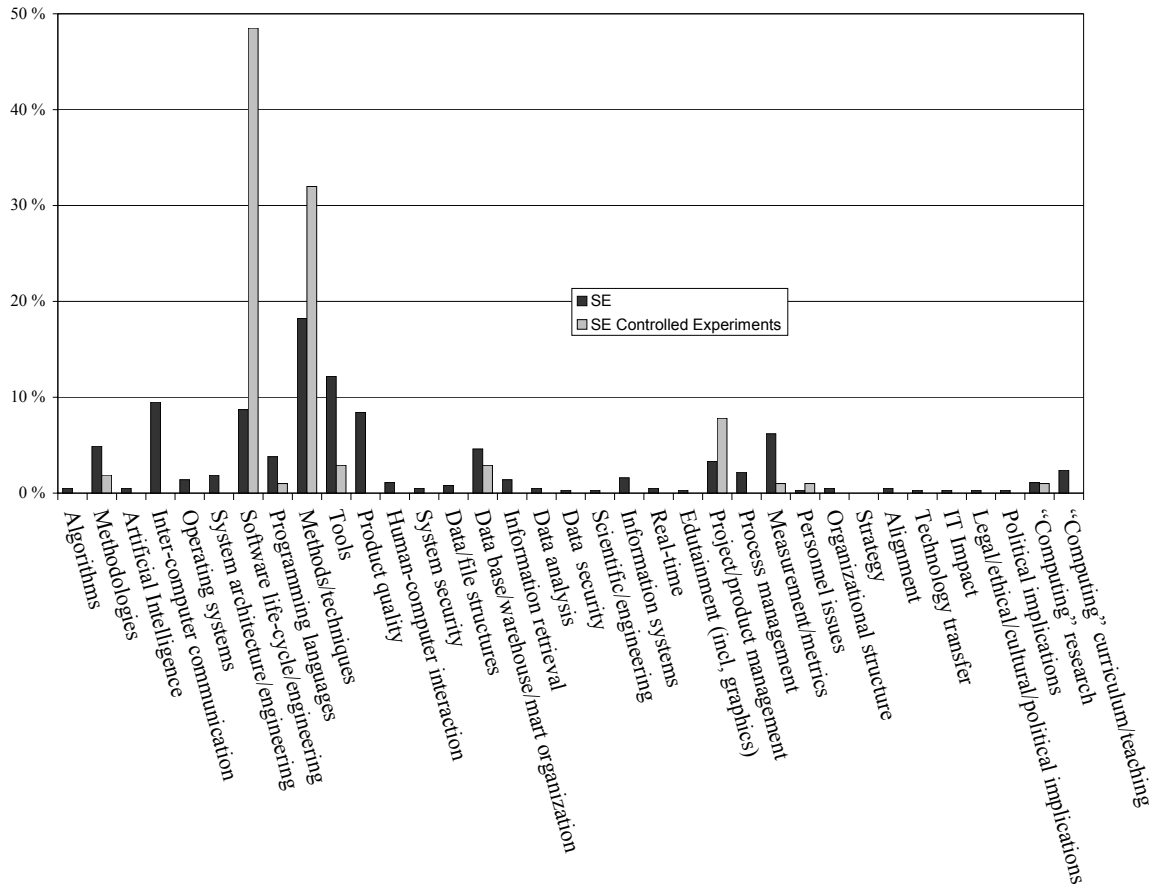

Fig. 1. Comparative distribution to topic of software engineering research and software engineering experiments using the scheme of Glass et al. Only nonvoid categories are shown.

5.2 Classification scheme: IEEE Keyword Taxonomy

The IEEE Keyword Taxonomy [24] provides a more technical perspective than the scheme of Glass *et al.* [20]. This taxonomy is recommended for authors of IEEE articles, and is an extended version of the ACM Computing Classification [2].

We use the IEEE keywords to denote topic categories. The classification according to the IEEE Keyword Taxonomy is given in Table 6. The two prominent technical areas are *Code inspections and walkthroughs* (35%) and *Object-oriented design methods* (8%). The numbers of experiments are limited for other areas.

6 Subjects

This section describes the kind of subjects that take part in the experiments, the kind of information that is reported about them, and how they are recruited.

6.1 Number and Categories of Subjects in the Experiments

In total, 5,88 subjects took part in the 113 experiments investigated in this survey. Eighty-seven percent were students and nine percent were professionals. The reported subject types are divided into the categories given in Table 7.

The number of participants per experiment ranges from four to 266, with a mean value of 48.6 (Table 8). Students participated in 91 (81%) of the experiments, either alone or together with professionals and/or scientists, and professionals took part in 27 experiments (24%). The use of professionals as subjects has been relatively stable over time. Undergraduates are used much more often than graduate students. For one experiment, no information about subject type was provided; for eight experiments, no details about type of students were given; and for five experiments with mixed types of subject, no information about the number in each category was provided.

The issue of differences between students and professionals has been discussed in the literature [13, 12, 48, 36]. Interestingly, while seven articles describe experiments using both students and professionals, only three of them measure the difference in performance between the two groups. In the first experiment, categorised as *Software psychology*, three programming tasks were performed. For two of the tasks, there was no difference between the groups, whereas for the third task, the professionals were significantly better. In the second experiment, also in *Software psychology*, there was no difference. In the third experiment, categorised as *Maintenance process*, the professionals were significantly better.

The performance in an experiment may differ between subcategories of subjects, that is, there may be an interaction effect between treatment and subcategory [3]. However, none of the surveyed experiments distinguished between subcategories of professionals or students.

The experiment with the highest number of professionals (68) was classified as *Cost estimation* in the IEEE-taxonomy (Table 6). Then there were five experiments with 29-35 professionals, of which one also employed 20 students. These five were categorised in descending order (regarding number of subjects) as *Modules and Interfaces*, *Code Inspections and walkthroughs*, *Maintenance process*, *Software Psychology* (understanding code), and *Patterns*.

The total number of participants was reported in all the articles, either explicitly or implicitly; in the latter case we could roughly calculate the number (for instance, from the information that 10 teams averaging four subjects participated). Subject mortality (drop-outs) was reported in 24 experiments (2% mortality on average). Even in experiments with as many as 266 subjects (as well as many other experiments with a relatively high number of subjects), no mortality was reported. One article states that "*Non-random drop-out of subjects has been avoided by the experimental design, i.e. assignment of groups only on the second day of the experiment, i.e. directly before the treatment, and not before the pre-test already on the first day of the experiment*." However, most articles say nothing about how mortality was managed.

There are good reasons for conducting experiments with students as subjects, for example, for testing experimental design and initial hypotheses, or for educational purposes [42]. Depending on the actual experiment, students *may* also be representative of junior/inexperienced professionals. However, the low proportion of professionals used in software engineering experiments reduces experimental realism, which in turn may inhibit the understanding of industrial software processes and consequently technology transfer from the research community to industry. Hence, to break the trend of few professionals as subjects, new strategies are needed to overcome these challenges, see *e.g.,* discussions in [39, 40].

Table 6. Classification of articles according to IEEE taxonomy

	Articles			
IEEE Taxonomy	N (group)	N	% (group)	%
General	**3**		**2.9**	
Software psychology		3		2.9
Requirements/Specifications	**4**		**3.9**	
Languages		1		1.0
Methodologies		2		1.9
Validation		1		1.0
Design Tools and Techniques	**1**		**1.0**	
Modules and interfaces		1		1.0
Coding Tools and Techniques	**2**		**1.9**	
Object-oriented programming		1		1.0
Structured programming		1		1.0
Software/Program Verification	**3**		**2.9**	
Formal methods		3		2.9
Testing and Debugging	**40**		**35.4**	
Code inspections and walkthroughs		37		35.9
Debugging aids		1		1.0
Testing strategies		1		1.0
Testing tools		1		1.0
Programming Environments/Construction Tools	**2**		**1.9**	
Graphical environments		2		1.9
Distribution, Maintenance, and Enhancement	**3**		**2.9**	
Documentation		1		1.0
Maintenance process		2		1.9
Metrics/Measurement	**1**		**1.0**	
Complexity measures		1		1.0
Management	**8**		**7.1**	
Cost estimation		1		1.0
Productivity		1		1.0
Programming teams		1		1.0
Project control & modeling		1		1.0
Risk management		1		1.0
Time estimation		3		2.9
Design	**15**		**13.3**	
Design notations and documentation		2		1.9
Representation		2		1.9
Methodologies		3		2.9
Object-oriented design methods		8		7.8
Software Architectures	**7**		**6.2**	
Domain-specific architectures		3		2.9
Languages		2		1.9
Patterns		2		1.9
Reusable Software	**4**		**3.9**	
Reuse models		4		3.9
Software and System Safety	**1**		**1.0**	
Software and System Safety		1		1.0
Software Construction	**4**		**3.9**	
Error processing		1		1.0
Programming paradigms		3		2.9
Software Engineering Process	**5**		**4.9**	
Software process models		5		4.9
Total		**103**		**100**

Table 7. Subject categories

Subject Category	Reported Subject Types	N	%
Undergraduates	Undergraduates, Bachelors, Third and fourth-year students, Last-year students, Honors and Majors.	2969	54.1
Graduates	Graduate students, Students following graduate courses or Master's programs, MSc and PhD students.	594	10.8
Students, type unknown	Students in computer science, Students.	1203	21.9
Professionals	Developers, Practitioners, Software engineers, Analysts, Domain experts, Business managers, Facilitators, Professionals.	517	9.4
Scientists	Professors, Post-doctorates, Staff members of educational institutions.	74	1.3
Unknown		131	2.3
Total		5488	100

Table 8. Participants in experiments

		Experiments		Subjects					
Category of subjects		N	%	Mean	Std	Min	Median	Max	Sum
Students only	Undergraduates only	43	38.1	63.2	61.1	10	43	266	2719
	Graduates only	15	13.3	25.1	11.1	9	24	48	377
	Undergraduates and graduates	16	14.2	60.6	57.8	6	42	208	970
	Students, type unknown	8	7.1	65.5	70.3	13	43	231	524
		82	72.6	56.0	56.8	6	36	266	4590
Professionals only		21	18.6	20.0	14.0	4	20	68	420
Mixed group of subjects		9	8.0	49.3	37.2	12	42	120	444
Unknown		1	0.9	34.0	-	34	34	34	34
Total		113	100	48.6	51.6	4	30	266	5488

Number and size of experiments in terms of subjects. The mixed group of subjects include students with scientists and/or professionals.

6.2 Information about subjects

In order to generalise from an experiment with a given group of subjects (sample population), one needs information about various characteristics and their variation both in the sample and in the group to which the results will be generalised (target population) [7]. For professionals, depending on what we wish to study, it would be relevant to know the variations regarding competence, productivity, education, experience (including domains), age, culture, *etc.* However, there is no generally accepted set of background variables for guiding data collection in a given type of study, simply because the software engineering community does not know which variables are the important ones. We have chosen to

focus on the variables that are reported in the analysed articles, that is, gender, age, education, experience and task-related training.

The analysed articles vary to a large extent on how they report such information. For 14 of the 113 experiments, no information about the subjects was reported. Moreover, the level of detail reported varies 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% of the subjects had some previous experience with object-oriented programming, 58% with programming GUIs.*" An example of a high level description without figures is: "*Some of the students had industrial programming experience.*"

For the 91 experiments with students, the following information was reported: gender (seven experiments), age (six experiments), grades (six experiments), programming experience (general description: 17 experiments, number of years/languages: 11 experiments), work experience in industry (general description: nine experiments, number of years: nine experiments), task-related experience (64 experiments) and task-related training (27 experiments). The training was either tailored specifically for the experiment or was part of a course, or the experiment could be conducted as part of a training session.

For the 27 experiments with professionals, more details on the subjects' background were given. Categories of professional such as reviewers, analysts, programmers and managers were given for seven experiments. Subjects' degrees were described for three experiments. Gender and age were given for, respectively, two and three experiments. Language and nationality were given for oneexperiment (subjects from two countries participated). A general description of programming experience was given for two experiments. Programming experience in years/languages was given for seven experiments. Self-assessment of programming experience was reported for two experiments. Work experience in years was given for five experiments. A general description of task-related experience was reported in one experiment. Task-related experience was measured in years for 13 experiments. Task-related training was reported for 12 experiments.

The relatively low and arbitrary reporting on context variables is a hindrance for meta-studies, which are needed to identify which context factors influence which kinds of performance. The impact of the various context factors will, of course, depend on the treatments and actual tasks to be performed in the experiments. Future work should

investigate the extent to which the variation in 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.

6.3 Recruitment of subjects

Recruiting subjects to experiments is not a trivial task; either from a methodological or a practical point of view. For example, volunteers may bias the results because they are often more motivated, skilled, *etc.* than subjects who take part because it is mandatory in some way [8]. Information about whether participation was mandatory is reported for 41 (36%) of the experiments. For 12 of them (all student experiments), participation was mandatory. Information about subject compensation for taking part in experiments is reported for 39 (35%) of the experiments. The grades of students were affected by the participation in 10 cases, and they received extra credits in nine cases (Table 9). In three cases, students were paid to take part, and in one case, students were sponsored for a trip to an exhibition. No compensation to professionals is reported. Typically, the experiments with professionals were organised as part of normal projects or training programmes, and payment was thus implicitly provided by the employer. Hence, it seems that none of the researchers or research teams paid companies or professionals for taking part in experiments.

If one applies statistical hypothesis testing, a requirement is to have a well-defined population from which the sample is drawn: "*If you cannot define the population from*

Table 9. Subject reward data

	Experiment		Participant	
Reward	N	%	N	%
Grade	10	8.8	732	13.3
Extra credits	9	8.0	660	12.0
Payment	3	2.7	121	2.2
Other rewards	1	0.9	24	0.4
No reward	16	14.4	458	8.3
Unknown	74	65.5	3493	64.6
Total	113	100	5488	100

which your subjects/objects are drawn, it is not possible to draw any inference from the results of your experiment" [30].[5] Nevertheless, none of the experiments in this survey that apply statistical hypothesis testing actually reported sampling from a well-defined target population.

For only a couple of experiments, random sampling of subjects was claimed. How the random sampling was carried out, was not described. The dominant approach was convenience sampling: "*Subjects are selected because of their convenient accessibility to the researcher. These subjects are chosen simply because they are the easiest to obtain for the study. This technique is easy, fast and usually the least expensive and troublesome. ... The criticism of this technique is that bias is introduced into the sample.*" [34]. This does not mean that convenience sampling is generally inappropriate. For example, Ferber [17] refers to the *exploratory*, the *illustrative*, and the *clinical* situations in which convenience sampling may be appropriate. In software engineering, the most convenient way of recruiting subjects is to use the students that are taught by the researcher. (Note that convenience sampling is also common in other disciplines such as clinical medicine [34] and social sciences [33].)

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

7 Tasks

The tasks that subjects are asked to carry out are an important characteristic of a software engineering experiment. Tasks may include building a software application from scratch or performing various operations on an existing application. This section reports on the surveyed experiments according to a high-level categorisation of their tasks and the duration of those tasks. Moreover, we describe the total *magnitude* of the experiments by

[5] This claim should, as we understand it, not be interpreted outside the context of statistical hypothesis testing. Obviously, even a study without a well-defined population (but with a well-defined sample) may enable the researcher to infer about similar projects, *e.g.,* based on argumentation by analogy or by theory, see further discussion in [26].

reporting the product of the number of subjects and the duration of the tasks. Finally, we describe the kinds and size of application and materials used in the experiments.

7.1 Task categorisation

We categorise tasks given to subjects according to the main tasks in a software process. We have defined four general categories, *Plan, Create, Modify and Analyse* that reflect major tasks on software artefacts. Table 10 shows subcategories within these major categories that have been identified in the surveyed experiments.

Task categorisation is somewhat different from topic categorisation. Tasks of a certain category can be used to test hypotheses within various topics. For example, a maintenance task can be used to test a certain design, or an experiment assigned to the *Patterns* category in the IEEE taxonomy might have design, coding or maintenance tasks.

Table 10 shows the number of experiments deploying each kind of task. Note that tasks of several categories might be involved in a single experiment. A task is represented by its fraction of all tasks in an experiment, for example, an experiment with one *Design* task and one *Coding* task gives a contribution of 0.5 to each of the two task categories. (Note also that we do not distinguish between tasks and sub-tasks because there is no commonly agreed definition of the unit of task. Hence, in the relatively few cases in which the experimenters have divided their tasks into subtasks, we have considered them as one task as long as they fall within the same category.) Due to experiment design, a task may be performed several times by the same subjects but with different treatments. In such cases, however, the task is only counted once.

The proportion of planning, creation, modification and analysis tasks is, respectively, 10%, 20%, 16% and 54%. *Inspection* tasks occur in 37 (33%) of the experiments, and are by far the most prominent. This is in accordance with the topic classification of articles reported in Section 5. Thirty-six of these experiments involve individual inspections, 29 involve team inspections. Twenty-eight experiments involve both individual and team inspections. Inspection tasks are typically conducted using pen and paper, although some use support tools.

Document comprehension tasks form the basis of various software engineering tasks, and typically involve answering questions about system structure and functionality. Twenty-three experiments involve document comprehension tasks, 12 of these pertain to

code documents, nine are design comprehension, one concerns a requirements document and one concerns process components.

Maintenance tasks pertain to all document types. The surveyed experiments, however, only deal with design and code maintenance tasks; 19 experiments give code maintenance tasks, and three give change tasks on design documents (including impact analyses). None give both. In 10 of the code maintenance tasks, new functionality was added. One of the code maintenance tasks is a pen-and-paper maintenance task performed jointly with a comprehension task.

Table 10. Task categorization, duration, and material size

				Duration						Materials§
	Experiments			Subject level			Slot level			
Task category	N*	%	Occ.†	N*	Occ.†	*median* (*h*)‡	N*	Occ.†	*median* (*h*)‡	Occ.†
Plan	**11.0**	**9.7**	**11**							
Project planning	4.5	4.0	5				1.5	2	0.5	3
Requirements analysis	1.0	0.9	1				1.0	1	0.7	
Estimation	5.5	4.9	6				0.5	1	0.5	3
Create	**22.8**	**20.2**	**25**							
Design	7.4	6.6	11	2.8	4	0.91	3.8	5	1.0	6
Coding	15.4	13.6	19	4.8	6	3.53	1.3	2	0.9	1
Modify	**18.6**	**16.5**	**22**							
Maintenance -	(18.6)	(16.5)	(22)							(15)
Change design	1.3	1.2	3	0.5	1	0.50	0.5	1	1.0	3
Change code	17.3	15.3	19	12.3	13	0.92	1.5	2	1.7	12
Analyse	**60.7**	**53.7**	**98**							
Inspection -	(35.1)	(31.1)	(37)							(28)
Individual	21.4	19.0	36	6.3	8	2.29	8.0	14	2.0	27
Team	13.7	12.1	29	1.3	3	1.00	6.0	12	2.0	24
Testing	6.6	5.9	10	4.0	6	0.99	1.0	1	0.3	7
Document compreh. -	(19.0)	(16.8)	(23)							(17)
Process doc.	1.0	0.9	1							
Req. Doc.	1.0	0.9	1	1.0	1	1.01				1
Design doc.	6.8	6.0	9	3.5	4	0.37	1.0	2	1.5	7
Code doc.	10.2	9.0	12	4.3	5	0.06	1.8	3	2.1	9
All experiments	**113**	**100**	**-**	**41**	**-**	**1.03***	**28**	**-**	**2.0***	**-**

***** The fraction of experiments.
† Occurrences of experiments. One experiment might be represented in several task categories.
‡ Median duration of tasks by category. The last row shows the median total duration for all the tasks of an experiment.
§ The occurrences of experiments in each task category that report size of materials. The total number of experiments that report size of materials is 67.

Coding and *Design* are tasks in which new artefacts are produced. Modifying existing code or design documents is classified as maintenance.

Most of the *Testing* tasks involve the generation of test harnesses and test cases. Testing here also includes debugging using debugging tools, but excludes inspections.

Three experiments investigate the effects of preplanning estimates on detailed estimates (anchoring). In one of these, the *Estimation* task is part of a *Project planning* exercise. One experiment involves estimation in a larger project, although project planning as such is not a task in the experiment in question. Two experiments issue estimation tasks in order to compare various estimation techniques.

Four of the five experiments with *Project planning* tasks are all-student experiments in which the subjects were asked to role-play in project planning or to simulate projects. The fifth one involves both professionals and students assessing how 10 different factors affect the lead-time of software development projects.

In the experiment involving *Requirements analysis,* the subjects were asked to negotiate a software requirements meeting with a customer.

Forty experiments deploy tasks in several categories. Among these experiments, five involve three tasks, two involve four tasks: one has comprehension, maintenance and inspection (individual and team) tasks, and one has design, coding, team inspection and testing.

7.2 Task duration

An important task characteristic is duration. Accurate duration data per subject (typically in dependent variables) is reported in 41 (36%) of the experiments and at *slot level* in 28 (25 %) of the experiments. (Time slots are coarse-grained indications, typically upper bounds, of how much time the subjects took to perform a task. For example, we chose a slot of two hours from the information that "*We gave each subject up to three hours to review each document (i.e., one document in the morning, and one in the afternoon). Only one subject took more than two hours*".) Duration data that is not considered sufficient for analysis is contained in phrases like "*Six days, no time limit*", "*From 1 to 11 days depending on the subjects' skills*", and "*Non-programming subjects had 30 min. to finish their task. Programming subjects had one week*".

Fig. 2 shows the frequency by time interval of the 41 experiments with detailed subject-level time information. It appears that about 2/3 of the experiments last less than two hours.

The two leftmost 'Subject level' columns of Table 10 show, for each task category respectively, the fraction of and the number of experiments with subject-level duration data that include tasks of this category. The third 'Subject level' column shows the median

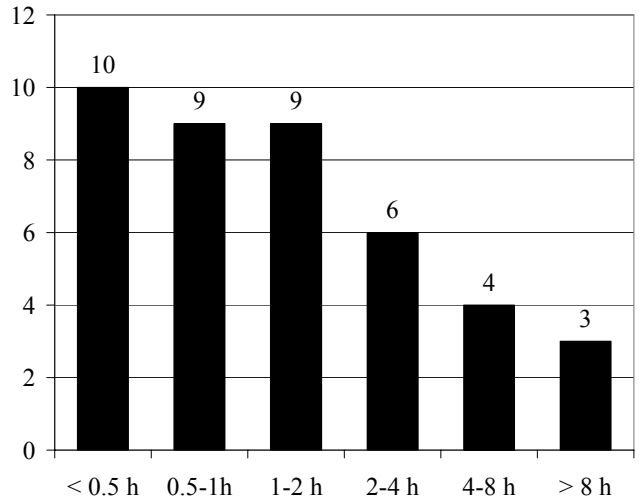

Fig. 2. Distribution of experiments with subject-level duration data to time intervals.

duration in hours for these tasks. For example, three experiments have duration data for design tasks. The fraction of the time spent on design activities in these three experiments, relative to the total time for all experiments (with subject-level time data), is 2.3. The median time used on design tasks in these experiments is 0.85 hours. The median duration of all experiments with subject-level data is 1.0 hours and 2.0 hours for the experiments with slot-level data.

Fig. 3 shows the actual task duration for the subject-level occurrences of Table 10. In the interests of saving space, four data points at, respectively, 25 (*Change code*), 18.5 (*Coding*), 18.5 (*Design*) and 55 hours (*Coding*) are omitted from the figure. It appears that there is large variance in duration, and that it seems independent of the type of task being performed.

Little is mentioned in the articles about control over context variables in experiments with multiple sessions, idle periods, or that span several days or weeks. Although the issue in principle concerns all experiments, it would be particularly interesting to know how experimenters have ensured control in experiments that involve long tasks.

The data described above reflects the duration of explicitly measured software engineering-specific tasks as described in Section 7.1. Often however, subjects perform additional tasks (training, preparation, post-mortem questionnaires, etc.) whose durations are not captured in dependent variables or are otherwise measured explicitly. If one wants to reflect the total time spent by subjects (perhaps in the interest of logistics), information at a different level must be gathered. Although most experiments (close to 80%) provide

some sort of information about total experiment duration, the data is, in general, measured and reported arbitrarily, and is consequently difficult to summarise here.

The median duration of the tasks of 1.0/2.0 hours is, of course, a very small fraction of the time of a typical industrial development project. The extent to which short tasks are a threat to external validity, is difficult to judge in general. The actual tasks in the experiments *may* be representative of typical industrial (sub)tasks. However, the lack of studies that describe "typical" tasks within certain categories and contexts makes such a judgement difficult. More studies are needed to investigate the relevance of the tasks being conducted in software engineering experiments.

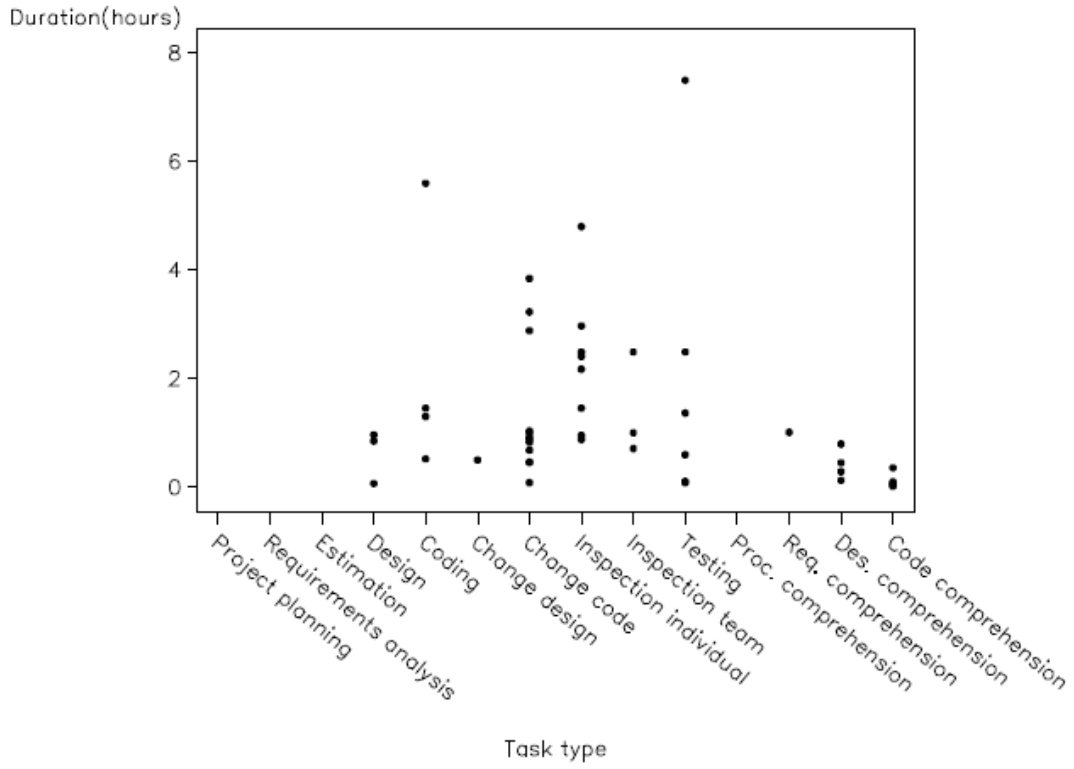

Fig. 3. Task categories and subject-level duration data.

7.3 Magnitude of experiments – combination of number of subjects and duration

Many aspects of the complexity of software engineering only manifest themselves in controlled experiments if the experiments involve a sufficiently large number of subjects and tasks, for example, differences among subgroups of subjects [3]. Hence, we can characterise the experiments in terms of the scale of the combination of subjects and tasks

(here, measured in terms of duration of the task). The magnitude of an experiment can be described in terms of the total number of person-hours or person-days; that is, the number of subjects multiplied with the length of the tasks.

In this survey, the experiment with the largest number of professionals lasts less than one hour. However, in general, there seems to be no significant relationship between duration and the number of subjects.

We here categorise the 69 experiments with duration data (41 with subject-level data and 28 slot-level data), according to subject numbers and task duration into, respectively, S (small), M (medium), and L (large), such that each category contains roughly 1/3 of the experiments. In practice, this gives the following categories. For subject numbers, S: ≤ 23, M: 23–47, and L: > 47. For duration, S: ≤ 0.96 hours, M: 0.96–2.9, and L: > 2.9 hours. (The subject groups cannot be made completely even because there are six experiments with 24 subjects.) The person-hours categorisation is obtained by crossing these two categorisations in configurations (subject category, duration category) as follows, S (small): (S,S), (S,M), (M,S); M (medium): (S,L), (M,M), (L,S); L (large): (M,L), (L,L), (L,M). Table 11 shows that experiments with professionals use a smaller number of subjects than do experiments with students. Both experiments with students and experiments with professionals have a uniform distribution for the three levels of duration. Regarding magnitude, most student experiments are in the middle category and a fair number are large, while most experiments with professionals are small and only one experiment is large.

Table 11. Distribution of experiments to subject number, duration, and subject-duration categories

Measure	Subjects (N)			Duration (*h*)			Person-Hours		
Level	S	M	L	S	M	L	S	M	L
Students (only)	15	18	21	18	19	17	14	24	16
Professionals (only)	5	3	0	3	2	3	4	3	1
Combination/Other	1	3	3	2	2	3	2	2	3
Total	21	24	24	23	23	23	20	29	20
Sum			69			69			69

7.4 Application and materials

Applications may be of various types, such as commercial, open source, the result of a student project, or custom-built for the purpose of the experiment. Table 12 shows that 75% of the surveyed experiments involved applications that were either constructed for the purpose of the experiment or were parts of student projects. Commercial applications were used in 16 experiments, of which 10 included inspection tasks (eight of these had team inspections in addition to individual inspections), two included design tasks, one had coding and maintenance (change code), one had coding only, one had (design) comprehension and maintenance (change design), and one had estimation. For 12 experiments, the reporting is

Table 12. Distribution of experiments to application type

Application type	N	%
Constructed	80	70.8
Commercial	16	14.2
Student project	5	4.4
Open source	0	0.0
Unclear	12	10.6
Total	113	100

unclear in this respect, but 11 of these appear to have used custom-built applications. There are no open source applications registered in this survey. The small fraction of commercial or industrial applications used in current software engineering experiments puts in question the possibility of generalizing the experimental results to an industrial setting.

The size of the materials presented to subjects gives some indications of the comprehensiveness of the experiments. Size in the form of pages, lines of code (LOC) or other quantities is reported in 67 (59%) of the experiments. The diversity of the surveyed experiments and how they report information about materials makes it difficult to give a systematic overview of the size of the experiment materials. Nevertheless, below we describe in brief the size of materials per task category, cf. the rightmost column of Table 10.

Three experiments with *Project planning* tasks report materials size: a one-page case scenario, a scenario in terms of 500 adjusted function points, and a four-page program representation, respectively. The three *Estimation* experiments are based on a 1,000

person-hour project, a four-page program representation, and on the creation of 10 programs (no sizes on these), respectively.

Five of the six experiments with size indications for *Design* tasks are on requirements documents (1-2 pages, six modules). The sixth experiment gives a one-page task description. Materials size is reported in one instance for a coding task (specification given in three tables).

Two experiments with *Change design* tasks have a 30 page design document as material (one experiment is an internal replication of the other; the materials of the former are improved, but their sizes are more or less the same), and one has a 1,500 LOC system as input. In the experiments with *Change code* tasks, the applications to be maintained range from 54 to 2,700 LOC. The largest application also involves 100 pages of documentation. Three experiments report the number of classes (6-10).

Twenty-eight experiments give materials size for *Inspection* tasks (individual or team). Fourteen give LOC (ranging from 135-3,955 LOC). In one instance, the materials size is given as 300 LOC, but the size of the entire system is 65,000 LOC. Page counts (ranging from 16-47 pages) are given in 14 instances (all different from the 14 with LOC). Materials size for *Testing* tasks (25-2,000 LOC) is reported in seven experiments (one also reports 10 classes). Reported materials sizes for *Document comprehension* tasks are varied (five diagrams, seven screens, 16 screenshots, etc.), but five experiments give LOC (92–2,700 LOC) for *Code comprehension* tasks, and five experiments give page counts (2–30 pages) for *Design comprehension* tasks.

In addition, some experiments (with tasks in the *Create* category) report the size of produced task solutions. Five experiments with *Coding* give LOC (in the range 86-2,000 LOC) for produced code, and in one experiment the size for a *Design* task is provided implicitly, in that the solution design document written by the experimenters is two pages. Also, the amount of added code is given for two maintenance tasks: 50–150 LOC and 35–79 LOC, respectively.

8 Environments

The strength of controlled experiments lies in that they may be used to isolate causal relationships. However, controlled experiments in the field of software engineering are often conducted in artificially designed environments that make it difficult to generalise the results to industrial contexts. In short, “Internal and external validity can be negatively

related" [37]. This section describes the surveyed experiments according to their location and tools used.

8.1 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.

For the 27 experiments with professionals or with professionals and students, 17 report no explicit information about the experimental setting. Only one experiment is reported to have been run in a usual office environment. The pilot of another experiment was run in an office environment, but the main experiment was run in a classroom setting in order to increase the internal validity. Three other experiments were run in a classroom setting, two of which were run as part of a training course. Seven experiments are explicitly reported to have been run in a laboratory environment.

Of the 85 experiments with students or with students and scientists, 56 report no explicit information about the experimental setting. For, respectively, 13 and seven of those experiments, it was explicitly stated they were conducted in a laboratory and classroom. For another group of nine experiments, some sort of university setting was stated, for example, "*academic settings*", "*conducted under exam conditions*" and "*supervised setting*". However, one may assume that all the experiments with students were carried out in a laboratory or classroom. Moreover, we believe that the distinction between a classroom and laboratory setting for students may be blurred and may depend on cultural differences, apart from the fact that a laboratory usually would include the use of PCs or workstations (2/3 of the experiments that report the use of a laboratory also report the use of PC or workstation, see the next section).

Approximately half of all the experiments with students report the name of the actual university/college. For the 27 experiments that include professionals, the name of the company is reported in 12 cases. For four experiments, the company is not named, but the type of company is specified. Note that, depending on the actual experiment, certain companies have a policy such that they must remain anonymous in the reporting of the

experiment. In five cases, the professionals are described as coming from "several" companies or organisations. The exact number of companies is not given.[6]

8.2 Tools

It is a challenge to configure the experimental environment with an infrastructure with supporting tools that resembles an industrial development environment. Among the surveyed experiments, 55% report on tools to support the tasks of the experiments (Table 13). This includes both explicit descriptions, *e.g.,* "*Sun-4, GNU C compiler* ", and implicit, but clear indications, *e.g.,* "*Developed programs were run against a set of test data*".

Table 13. Distribution of experiments to specific tool

Tool	N	%
PC or workstation (only)	32	28.3
Pen and paper (only)	25	22.1
Combination	5	4.4
Unknown	51	45.1
Total	113	100

Table 13 shows that the use of computer tools is slightly higher than the use of pen and paper. However, it is likely that a larger proportion of those experiments that do not report on tools are actually pen and paper experiments, because the added effort and administrative overhead of using computer tools might inspire researchers to report the use of tools more than the use of pen and paper.

The task types that give the largest and smallest contribution to the *PC or workstation* category are, respectively, *Coding* and *Inspection*. Other than that, there is little correlation between task type and tool for the experiments that actually report on this issue. Moreover, there was no difference between experiments with professionals and experiments with students regarding the use of tools.

Three of the five experiments with *Combination* in Table 13 explicitly test the effects of computerised tool use versus pen and paper.

[6] Before we decided to rely exclusively on the information reported in the articles, we approached the corresponding authors of these five experiments to acquire more information about the extent of companies involved in the experiments. It turned out that in two experiments, the subjects attended a course aimed at people from industry (the number of companies of the participants was unknown). One author replied that it was a mistake in the article; all participants actually came from the same company. One replied that he did not know, but our impression was that it was only two companies. The last one did not respond to our request.

The relatively meagre proportion of experiments that report on the use of tools to support assigned tasks may be due to an unawareness of, or a lack of interest in, the relevance of this issue. For example, most of the experiments in the *Unknown* category are inspection experiments, for which it may be normal to use pen and paper. However, for most design, coding, testing and maintenance tasks, computer tools would have been used in an industrial setting, although the line is not clear-cut. For example, designers may sketch preliminary versions by hand, but the final design would be made using a tool.

In general, increasing the realism of software engineering experiments entails an increased use of industrial supporting tools. The community should thus recognise the effort and resources needed to set up PC or workstation environments with the right licences, installations, access rights, *etc.*, and to familiarise the subjects with the tools. Moreover, the tools must be checked to demonstrate acceptable performance and stability when many subjects are working simultaneously.

In the experiments of this survey, there is almost no discussion of the relationships among the three dimensions *subject*, *task* and *environment*. For the community to progress, this issue needs to be investigated. For example, a professional development tool will probably become more useful the larger and more complex the tasks and application systems become, assuming that the subjects are sufficiently proficient with the tool.

9 Replication

In this survey, 20 of the experiments are described by the authors themselves as replications. These experiments constitute 14 series of replications. Table 14 summarises the series including both the original experiments and the replications, and reports differences between them. Most replications (35%) are conducted in the area of *Inspection* (seven replications in series 1, 2 and 3) and *Maintenance* (five replications in series 4, 5 and 6). Among the 20 replications, five can be considered as *close* replications in the terminology of Lindsay and Ehrenberg [31], i.e., one attempts to retain, as much as is possible, most of the known conditions of the original experiment. The other replications are considered to be *differentiated* replications, i.e., they involve variations in essential aspects of the experimental conditions. One prominent variation involves conducting the experiment with other kinds of subject; three replications use professionals instead of students, three use undergraduates instead of graduates, and one uses students instead of

professionals. Other variations include conducting the experiment on different application systems (four), and with different tasks (three).

Table 14. Replicated experiments

Series	Topic	Exp.	Stud.	Prof.	Con.	Rej.	Authors	Repl. Type	Other differences
1	Perspective-Based Reading (requirements inspection)	0	X		-	-	-	-	
		1		X	X		same	differentiated	
		2	X			X	others	differentiated	undergrads, (originally graduates)
		3	X			X	others	differentiated	undergrads, more, time extended
		4	X			X	others	differentiated	undergraduate
2	Perspective-Based Reading	0		X	-	-	-		
		1	X			X	others	differentiated	
3	Perspective-Based reading	0		X	-	-	-		
		1		X	X		same	differentiated	Diff. applications
		2		X	X		same	close	
4	Maintenance Process	0	X		-	-	-		
		1	X		X		same	differentiated	More tasks than in Exp. 0
		2		X	X		same	differentiated	Same as Exp. 1
5	Maintainability of OO systems (inheritance depth)	0	X		-	-	-		
		1	X		X		same	close	
6	Maintainability of OO systems (inheritance depth)	0	X		-	-	-		
		1	X			X	others	differentiated	diff. appl. and tasks, added hypotheses
		2	X			X	others	differentiated	Diff. appl.
7	Quality guidelines (maintainability of OO systems)	0	X		-		-		
		1	X		X		same	differentiated	more subjects, diff. Tasks
8	DB referential integrity metrics	0*	X		-	-	-		
		1		X	X	X	same	differentiated	
9	Layering and encapsulation	0	X		-	-	-		
		1	X		X		same	close	
10	Comprehension of OO models	0	X		-	-	-		
		1	X		X		same	differentiated	Diff. applications
11	Visual depiction of decision stmt.	0*		X	-	-	-		
		1		X	X		others	close	
12	Defect detection	0*	X		-	-	-		
		1	X		X		others	close	
13	Use Case guidelines	0*	X		-	-	-		
		1	X		X	X	others	differentiated	Diff. eval. criteria
14	Design Patterns	0	X				-		
		1	X		X		same	differentiated	diff. prog. lang. and rating scale.

Column *Exp.* presents the number in the replication series. The original experiments are denoted by '0'. Columns *Stud.* and *Prof.* indicate whether the subjects were students or professionals. Columns *Con.* and *Rej*. Indicate whether the replications confirm or reject the findings of the original experiment. '*' indicate that the original experiment was published in a journal or conference proceedings not included in the survey.

In all the five close replications, the results of the original experiment were confirmed (three were conducted by the same authors, two by others). Among the 15 differentiated replications, seven were conducted by other authors. Six of these reported results differing from the original experiment, and one partly confirmed the results of the original experiment. Among the differentiated replications conducted by the original authors, we found the opposite pattern; seven replications confirmed the results of the original experiment, and only one reported partly different results.

"Methodological authorities generally regard replication, or what is also referred to as 'repeating a study', to be a crucial aspect of the scientific method" [31]. However, only

18% of the surveyed experiments were replications. A discussion of the form and extent ofreplication that would be benefit software engineering is beyond the scope of this paper, but should be an issue of discussion for the research community.

10 Threats to internal and external validity

Two important aspects of the quality of an experiment are their internal and external validity. This section discusses how, and the extent to which, threats to internal and external validity are reported for the surveyed experiments. Descriptions of such threats are made in various ways and under different headings. For 54 experiments (48% of all experiments), there is a special section entitled "Threats to (internal/external) validity" or other combinations that include the terms "threats" or "validity". Nine other experiments (8%) have 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.

10.1 Internal validity

Internal validity of an experiment is "*the validity of inferences about whether observed co-variation between A (the presumed treatment) and B (the presumed outcome) reflects a causal relationship from A to B as those variables were manipulated or measured*" [37]. Changes in *B* may have causes other than, or alternative to, the manipulation of *A*. Such an alternative cause for the outcome is called a *confound* or *confounding factor*. For further discussions (including formal definitions) of concepts of confounding, see [21].

Threats to internal validity are addressed explicitly for 71 experiments (63%). (We did not include threats that are addressed implicitly as part of the experimental design.) We classified the reporting of threats to internal validity according to the scheme of Shadish *et al.* [37] shown in Table 15. That table also shows examples of threats in the various categories reported for the surveyed experiments. A version of this scheme, with other examples from software engineering, is presented in [46].

Table 16 shows the distribution of the experiments according to the scheme of Table 15. Almost half of all experiments report on selection threats (46%) and/or instrumentation threats (40%). The distribution of number of threat categories reported is as follows: 22 experiments report one threat, 11 experiments report two threats, 23 report three, 10 report four, four report five, and one experiment reports seven threats.

Table 16 also shows whether the authors consider the threats to be present but not handled by the authors, or reduced or eliminated due to actions taken by the authors or due to other circumstances. Only 18% of the threats (reported in 23% of the experiments) are not handled, but it may be the case that threats that are not reduced or eliminated are under-reported.

Classifying internal validity is not always straightforward. For example, "learning effects" are often classified as "maturation" in the experiments, while this should be "testing" according to the categories given in Shadish *et al.* [37]. Maturation threats refer to "*natural changes that would occur even in the absence of treatment, such as growing older, hungrier, wiser, stronger, or more experienced*", while testing threats refer to effects of practice and familiarity within the experiment that could be mistaken for treatment effects [37]. Moreover, threats that by this scheme pertain to *statistical conclusion validity* or *construct validity* were, for a few experiments, reported as internal validity threats. In part, this may be due to non-trivial subtleties in threat classification, illustrated by the fact that the line of development starting with Campbell *et al.* [9], via Cook *et al.* [11] to the present classification scheme in [37], shows considerable variation. For example, the notions of statistical conclusion validity and construct validity appeared for the first time in 1979 [11].[7]

[7] In addition, there are threats that scholars put in different main categories. For example, what Trochim [44] and Wohlin [46] refer to as "social threats" are categorised as threats to internal validity by them, but as threats to construct validity by Shadish *et al.* [37]. Four experiments address "social threats" in our survey, but since we follow the scheme of Shadish *et al.*, such threats are not included in our survey.

Table 15. Threats to internal validity: reasons why inferences that the relationship between two variables is causal may be incorrect

	Description given by Shadish *et al*.	Examples from the survey
1.	*Ambiguous Temporal Precedence*: Lack of clarity about which variable occurred first may yield confusion about which variable is the cause and which is the effect.	None
2.	*Selection*: Systematic differences over conditions in respondent characteristics that could also cause the observed effect.	Random assignment and blocking, in combination with randomisation or alone, and within-subject design were often mentioned as reducing factors.
3.	*History*: Events occurring concurrently with treatment could cause the observed effect.	Most cases concerned individuals or teams communicating during the experiments. Attempts to reduce this effect include: "The subjects were instructed not to discuss the experiment or otherwise do anything between the tests that could cause an unwanted effect on the results."
4.	*Maturation*: Naturally occurring changes over time could be confused with a treatment effect.	Most cases concerned boredom, fatigue, demotivation and loss of enthusiasm, for example: "The boredom effect might have affected the second run of the experiment, because subjects had to perform a second complete inspection using the same review technique", "Demotivation may also play a part as subjects become bored with three weeks of testing"*
5.	*Regression*: When units are selected for their extreme scores, they will often have less extreme scores on other variables, an occurrence that can be confused with a treatment effect.	"The absence of pretest scores to assign subjects to groups, the use of simple tasks, and the presence of multiple groups control for statistical regression"
6.	*Attrition*: Loss of respondents to treatment or to measurement can produce artifactual effects if that loss is systematically correlated with conditions.	"A threat to the internal validity that was considered in the analysis is that the subjects did not have enough time to apply all the use cases", "Of the twenty subjects who expressed an interest in the study only thirteen of them actually turned up to participate"
7.	*Testing*: Exposure to a test can affect scores on subsequent exposures to that test, an occurrence that can be confused with a treatment effect.	"We cannot exclude that learning was still in progress during the experiment. We tried to minimize the learning effect by teaching requirements specification and review and having a training session before the experiment itself."
8.	*Instrumentation*: The nature of a measure may change over time or conditions in a way that could be confused with a treatment effect.	"Instrumentation effects may result from differences in the specification documents. Such variation is impossible to avoid, but we controlled for it by having each team inspect both documents."
9.	*Additive and Interactive Effects of Threats to Internal Validity*: The impact of a threat can be added to that of another threat or may depend on the level of another threat.	None

Further discussions of the impact of motivation of subjects in software engineering experiments may be found in [23].

Table 16. Threats to internal validity

	No of experiments				
Category	Threat not handled	Threat reduced	Threat eliminated	Total	% of all exp.
Selection	10	35	7	52	46.0
Instrumentation	9	30	6	45	39.8
Maturation	3	14	6	23	20.4
Testing	2	22	4	28	24.8
History	3	9	6	18	15.9
Attrition	5	3	4	12	10.6
Regression	0	1	1	2	1.8
Ambiguous Temporal Precedence	0	0	0	0	0.0
Additive and Interactive Effects of Threats to Internal Validity	0	0	0	0	0.0
No of threats*	32 (17.8%)	114 (63.3%)	34 (18.9%)	180 (100%)	
No of Experiments	26 (23.0%)	55 (48.7%)	19 (16.8%)	71† (62.8%)	

* We do not distinguish between one or more threats within a category for a given experiment; that is, only one threat per category is counted per experiment.
† Note that the total number of experiments is not the sum of the previous three columns because one experiment may be represented in more than one category.

10.2 External validity

External validity concerns inferences about the extent to which a causal relationship holds over variations in persons, settings, treatments and outcomes [37]. This section summarises how the authors report threats to external validity regarding these issues.

Threats to external validity are reported for 78 experiments (69%). Table 17 shows a categorisation of the threats based on Shadish *et al.* [37]. Threats regarding subjects are discussed in a total of 67 experiments (rows one, four, five, six and seven), regarding task in 60, environment in 23 and treatment in six.

Most threats regarding subjects deal with difficulties of generalising from students to professionals (45 experiments). Another category of experiments (14) also uses students, but the experimenters argue that this may not be a threat to validity because the students for this kind of task would (probably) have the same ability as professionals (seven), because the students were close to finalising their education and start working in industry (four), or because one other study showed no difference between students and professionals (three). few experiments (three) that use professionals claim that threats to external validity were not critical because the experiment was conducted with professionals (they did not discuss the representativeness of their actual sample of professionals). A few experiments (three) considered that running an experiment within a

single organization was a threat to the generalisation to other organizations. Very few experiments (two) explicitly described the lack of random sampling as a threat to validity.

Table 17. Threats to external validity

Factors addressed as threats to external validity	Experiments	%
Subject (only)	14	12.4
Task (only)	10	8.8
Environment (only)	1	0.9
Subject and environment	2	1.8
Subject and task	31	27.4
Subject, environment and task	14	12.4
Treatment and subject, task or environment	6	5.3
Threats to external validity not addressed	35	31.0
Total	113	100

Most of the task-related threats concern size and complexity of the tasks (16 experiments) and experimental material (34), such as program code, inspection documents and database systems. For experiments on inspection, one threat discussed was that the inspection process applied was not considered representative for industrial practice (nine). The (short) duration of the experiment was also regarded as a threat (three). One experiment stated that "*all our results were obtained from one project, in one application domain, using one language and environment, within one software organisation. Therefore, we cannot claim that our conclusions have general applicability, until our work has been replicated.*" Another experiment stated that the subjects might not have used the technique intended to be studied in the experiment.

Threats regarding environment were either stated as a problem of generalising from the experimental setting with no specific reasons (five experiments) or stated with concrete reasons for the difficulties: use of laboratory or classroom (nine), individual work (five), and use of pen and paper (six).

A major finding is that the reporting is vague and unsystematic. The community needs guidelines that provide significant support for how to draw conclusions from the experimental results and on how to address threats to internal and external validity and their consequences.

11 Threats to Validity of this Survey

The main threats to validity for this study are publication selection bias, inaccuracy in data extraction and misclassification.

11.1 Selection of journals and conferences

We consider the 12 surveyed journals and conferences to be leaders in software engineering in general and empirical software engineering in particular. (Our selection of journals is a superset of those selected by others, as shown in Table 1.) Nevertheless, a systematic survey that included, in addition, grey literature (theses, technical reports, working papers, *etc.*) describing controlled experiments in software engineering would, in principle, provide more data and allow more general conclusions to be drawn [29].

11.2 Selection of articles

To help ensure an unbiased selection process, we defined research questions in advance, organised the selection of articles as a multistage process, involved several researchers in this process, and documented the reasons for inclusion/exclusion as suggested in [29].

The initial investigation of the titles and abstracts of 5,453 articles resulted in 140 survey articles. Based on recorded comments, 80 of these were reanalysed by one or two other researchers and discussed in the project group. Seventeen further articles were then excluded because they described studies without a treatment. Moreover, three articles were found to be exploratory, observational or constituting a pre-study. Eight were found to fall outside the field of software engineering five were excluded on the grounds that they were summary articles, while four articles described multiple case studies. We used Inspec and various search engines to check the completeness of our inclusion, and cross-checked for inclusion with other surveys [51, 22, 25]. Still, the process was difficult and we may not have managed to detect all articles that we would have liked to include.

Another challenge was that there is no keyword standard that we are aware of that distinguishes between methods in empirical software engineering and that could be used to extract controlled experiments in a consistent manner. For example, none of the selected articles matched the IEEE keyword taxonomy; indeed, this taxonomy has no appropriate keywords for the methods of empirical software engineering. (*MIS Quarterly* has ceased to use their keyword classification scheme due to the presence of full-text search engines and the difficulty of keeping keyword classification schemes up to date [45].)

Moreover, article and experiment duplication is a potential threat to frequency counts and the statistics in this survey. Among the 113 experiments covered in the 103 articles, 109 are reported in one article, two are reported in two articles, one is reported in three articles, and one is reported in four. Among the 103 surveyed articles, 91 report a single experiment, seven report two experiments, and five report three experiments. We detected one case of near article duplicates in different journals. The structure of the database is designed to handle duplication, but a threat would be that duplication goes undetected. However, at least three people have read through all relevant articles without detecting further duplicates.

11.3 Data extraction

The data was extracted from the articles independently by two researchers. The inter-rater agreement varied from 73% to 100%. Disagreements were resolved by discussion and, when necessary, by involving other project members. Data extraction from prose is difficult at the outset and the lack of standard terminology and standards for reporting experiments in software engineering may have resulted in some inaccuracy in the data.

11.4 Classification to topics

The classification of articles to topics was done in two steps. First, the articles were classified automatically on the basis of title, list of keywords, and registered treatment. Then, this classification was double-checked by two researchers. The inter-rater agreement between the algorithm and the two researchers was 75% for the comparative classification using Glass *et al.*'s scheme, and 66% for the IEEE-classification. The topic classification was difficult, due to the lack of a well-defined method of classifying according to the schemes used.

12 Summary

This paper reported a survey that characterized quantitatively the controlled experiments in software engineering published in nine journals and three conference proceedings in the decade from 1993 to 2002. Included were randomised experiments or quasi-experiments in which individuals or teams (the experimental units) applied a process, method, technique, language or tool (the treatments) to conduct one or more software engineering tasks. Out of 5,453 articles scanned, we identified 103 articles that reported 113 controlled experiments.

Although as many as 108 institutions from 19 countries were involved in conducting the experiments, a relatively low proportion of software engineering articles (1.9%) report controlled experiments, given that controlled experiments is the classical scientific method for identifying cause-effect relationships. One reason may be the large effort and resources needed to run well-designed experiments.

An important issue that pertains to all software engineering research is its relevance to the industry. For experiments, both the topics under investigation and how representative of an industrial setting an experiment is will influence industrial relevance. The two major areas investigated in the experiments were inspection techniques and object-oriented design techniques. This survey also gave some indications as to how realistic the experiments were relative to the subjects that took part, the tasks they performed, the types of applications on which these tasks were done, and the environment in which the subjects worked.

In total, 5,488 subjects participated in the experiments. The number of participants ranged from 4 to 266, with a mean value of 49. In total, 87% of the subjects were students, whereas only 9% were professionals. This indicates that one may question how representative the experimental results are for an industrial setting.

The same applies to the kind of application used in the experiments. In 75%, the applications were constructed for the purpose of the experiment or constituted student projects. Commercial applications were used in 14% of the experiments.

Threats to internal and external validity were addressed in respectively, 63% and 69% of the experiments. Among the threats to internal validity, about 1/5 were not handled, 3/5 were reduced and 1/5 were eliminated. This could either mean that the experiments all over had a high degree of internal validity or that the internal threats that were not reduced or eliminated were underreported. Threats to external validity regarding subject and task were discussed in more than half of the experiments, regarding environment in about 1/4 of the experiments and regarding treatment in only a few. Threats to internal validity regarding selection and instrumentation were most frequently reported.

A major finding of this survey is that the reporting is often vague and unsystematic, and there is often a lack of consistent terminology. The community needs guidelines that provide significant support on how to deal with the methodological and practical complexity of conducting and reporting high-quality, preferably realistic, software engineering experiments. We recommend that researchers should accurately report the following: the type and number of subjects, including the mortality rate; context variables

such as general software engineering experience and experience specific to the tasks of the experiments; how the subjects were recruited; the application areas and type of tasks; the duration of the tasks; and internal and external validity of the experiments, including being specific about the sample and target population of the experiment. A more uniform way of reporting experiments will help to improve the review of articles, replication of experiments, meta-analysis and theory building.

Appendix

See table 18 and Table 19.

Table 18. Total number of articles investigated

	Year										
Journal	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	Total
EMSE	-	-	-	10	24	14	17	19	24	16	124
ISESE	-	-	--	-	-	-	-	-	-	20	20
METRICS	15	11	-	17	18	32	31	-	30	23	177
JSS	87	78	76	74	82	91	95	112	101	90	886
TSE	85	74	77	65	52	83	55	68	62	76	687
ICSE	48	31	32	59	51	64	56	64	58	57	520
IST	67	69	62	69	76	80	87	83	78	74	745
SME	12	16	22	21	18	18	20	19	19	21	186
IEEE SW	50	56	45	51	52	48	59	60	55	56	532
TOSEM	13	12	10	13	12	13	13	14	11	14	125
IEEE Comp	70	76	74	83	91	79	78	73	81	75	780
SP&E	69	59	68	68	71	72	68	65	65	66	671
Total	516	482	466	530	547	584	579	577	584	588	5453

Table 19. Number of articles that report controlled experiments

	Year										
Journal	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	Total
EMSE	-	-	-	2	6	5	1	5	1	2	22 (17.7% of 124)
ISESE	-	-	-	-	-	-	-	-	-	3	3 (15.0% of 20)
METRICS	0	0	-	2	0	4	0	-	3	1	10 (5.6% of 177)
JSS	1	1	1	4	0	4	5	6	1	1	24 (2.7% of 886)
TSE	2	1	2	0	2	1	1	3	3	2	17 (2.5% of 678)
ICSE	0	1	0	1	1	1	1	3	3	1	12 (2.3% of 520)
IST	0	0	0	1	2	2	0	0	3	0	8 (1.1% of 745)
SME	0	0	0	0	0	0	0	0	1	1	2 (1.1% of 186)
IEEE SW	0	0	0	0	0	0	0	3	1	0	4 (0.8% of 532)
TOSEM	0	0	0	0	0	1	0	0	0	0	1 (0.8% of 125)
IEEE comp	0	0	0	0	0	0	0	0	0	0	0 (0% of 780)
SP&E	0	0	0	0	0	0	0	0	0	0	0 (0% of 671)
Total	3 0.6% Of 516	3 0.6% of 482	3 0.6% Of 466	10 1.9% of 530	11 2.0% of 547	18 3.1% of 584	8 1.4% of 579	20 3.5% of 577	16 2.7% of 584	11 1.9% of 588	103 (1.9% of 5453)

Acknowledgements

The authors grateful to Barbara Kitchenham, Ray Welland, Erik Arisholm, Reidar Conradi, Tore Dybå, Magne Jørgensen, James Dzidek and the anonymous reviewers for insight and feedback to several key issues covered in this survey. Thanks are due to Jørgen Busvold, Ingeborg Nygard, Ragnfrid Sjøberg and Jørn Inge Vestgården for assistance in compiling data. Thanks to Chris Wright for proof-reading the paper.

References

[1] A. Abran and J.W. Moore, SWEBOK – Guide to the Software Engineering Body of Knowledge, *2004 Version, IEEE CS Professional Practices Committee*, 2004.

[2] ACM Computing Classification, http://theory.lcs.mit.edu/~jacm/CR/1991., 1995.

[3] E. Arisholm and D.I.K. Sjøberg, Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software, *IEEE Transactions on Software Engineering*, 30 (8) (2004) 521–534.

[4] V.R. Basili, The experimental paradigm in software engineering. In D. Rombach, V. Basili, and R.W. Selby, (Eds), Experimental Engineering Issues: Critical Assessment and Future Directions, *Proc. of Int'l Workshop, Dagstuhl Castle (Germany)*, Springer Verlag, vol. 706 (1993) 3–12.

[5] V.R. Basili, The role of experimentation in software engineering: past, current, and future, *Proc. of the 18th Int'l Conf. on Software Engineering (ICSE), Berlin (Germany)*, IEEE Computer Society, (1996) 442–449.

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

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

[8] D.M. Berry and W.F. Tichy, Response to "Comments on Formal Methods Application: An Empirical Tale of Software Development", *IEEE Transactions on Software Engineering* 29 (6) (2003) 572–575.

[9] D.T. Campbell and J.C. Stanley, Experimental and quasi-experimental designs for research on teaching. In N.L. Cage, (Ed.), *Handbook of Research on Teaching*, Rand McNally, Chicago, 1963.

[10] L.B. Christensen, *Experimental Methodology*, Pearson/Allyn & Bacon, Boston, 8th edition, 2001.

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

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

[13] B. Curtis, By the way, did anyone study real programmers? In *Empirical Studies of Programmers, Proc. of the First Workshop*, Ablex Publishing Corp. (1986) 256–262.

[14] I.S. Deligiannis, M. Shepperd, S.Webster, and M. Roumeliotis, A review of experimental investigations into object-oriented technology, *Empirical Software Engineering* 7 (3) (2002) 193–231.

[15] A. Endres and D. Rombach, *A Handbook of Sofware and Systems Engineering. Empirical Observations, Laws and Theories*, Fraunhofer IESE Series on Software Engineering. Pearson Education Limited, 2003.

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

[17] R. Ferber. Editorial: Research by convenience, *Journal of Consumer Research*, 4 (1977) 57–58.

[18] R.L. Glass and T.Y. Chen, An assessment of systems and software engineering scholars and institutions (1998-2002), *Journal of Systems and Software* 68 (1) (2003) 77–84.

[19] R.L. Glass, V. Ramesh, and I. Vessey, An analysis of research in computing disciplines, *Communications of the ACM* 47 (6) (2004) 89–94.

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

[21] S. Greenland, J.M. Robins, and J. Pearl, Confounding and collapsibility in causal inference, *Statistical Science* 14 (1) (1999) 29–46.

[22] W. Hayes, Research synthesis in software engineering: A case for meta- analysis, In *Software Metrics. Proc. of the 6th Int'l Symposium on Software Metrics (METRICS'03), Boca Raton, FL (USA),* IEEE Computer Society (2003) 143–151.

[23] M. Höst, C. Wohlin, and T. Thelin, Experimental context classification: Incentives and experience of subjects, In *Proc. of the 27th Int'l Conf. on Software Engineering (ICSE '05), St. Louis, MO (USA),* IEEE Computer Society (2005) 470–478.

[24] IEEE Keyword Taxonomy, http://www.computer.org/mc/keywords/software.htm, 2002.

[25] M. Jørgensen, A review of studies on expert estimation of software development effort, *Journal of Systems and Software* 70(1,2) (2004) 37–60.

[26] M. Jørgensen and D.I.K. Sjøberg, Generalization and theory building in software engineering research. In *Empirical Assessment in Software Engineering, Proc. of EASE 2004*, IEE, (2004) 29–36.

[27] M. Jørgensen, K.H. Teigen, and K. Moløkken, Better sure than safe? Over-confidence in judgement based software development effort prediction intervals, *Journal of Systems and Software* 70 (1,2) (2004) 79–93.

[28] Juristo, A.M. Moreno, and S. Vegas, Reviewing 25 years of testing technique experiments, *Empirical Software Engineering* vol. 9 (2004) 7–44.

[29] B.A. Kitchenham, Procedures for Performing Systematic Reviews, Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.

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

[31] R.M. Lindsay and A.S.C. Ehrenberg, The design of replicated studies, *The American Statistician*, vol. 47 (1993) 217–228.

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

[33] J.W. Lucas, Theory-testing, generalization, and the problem of external validity, *Sociological Theory*, vol. 21 (2003) 236–253.

[34] T.R. Lunsford and B.R. Lunsford, The research sample, part I: Sampling. *Journal of Prosthetics and Orthotics* vol. 7 (1995) 105–112.

[35] H.D. Rombach, V.R. Basili, and R.W. Selby, (Eds.) *Experimental Software Engineering Issues: Critical Assessment and Future Directions, Int'l Workshop Dagstuhl Castle (Germany), September 14-18, 1992, Proceedings*, vol. 706 of *Lecture Notes in Computer Science*. Springer Verlag, 1993.

[36] P. Runeson, Using students as experimental subjects – an analysis of graduate and freshmen PSP student data, In *Empirical Assessment in Software Engineering. Proc. of EASE 2003* (2003) 95–102.

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

[38] Mary Shaw, Writing good software engineering research papers: minitutorial, *25th Int'l Conf. on Software Engineering (ICSE '03), Portland, OR (USA)*, IEEE Computer Society, (2003) 726–736.

[39] D.I.K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanović, E. Koren, and M. Vokáč, Conducting realistic experiments in software engineering, In *Proc. of the 18th Int'l Symposium on Empirical Software Engineering (ISESE), Nara (Japan)*, IEEE Computer Society (2002) 17–26.

[40] D.I.K. Sjøberg, B. Anda, E. Arisholm, T. Dybå, M. Jørgensen, A. Karahasanović, and M. Vokáč, Challenges and recommendations when increasing the realism of controlled software engineering experiments, *Empirical Methods and Studies in Software Engineering (ESERNET), 2001–2002*, vol. 2765 of *Lecture Notes in Computer Science*, Springer Verlag (2003) 24–38.

[41] W.F. Tichy, Should computer scientist experiment more? 16 excuses to avoid experimentation. *IEEE Computer* 31 (5) (1998) 32–40.

[42] W.F. Tichy, Hints for reviewing empirical work in software engineering, *Empirical Software Engineering* 5 (4) (2000) 309–312.

[43] 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.

[44] W.M.K Trochim, *The Research Methods Knowledge Base*, second ed., Atomic Dog Publishing, Cincinnati, 2001.

[45] R. Weber, Editor's comments, *MIS Quarterly*, 27 (3) (2003) iii–xii.

[46] C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell, and A. Wesslen, *Experimentation in Software Engineering: An Introduction,* John Wiley & Sons Inc., 1999.

[47] R.K. Yin, *Case Study Research: Design and Methods,* Thousand Oaks, California, Sage, 2003.

[48] E. A. Youngs, Human errors in programming, *International Journal of Man-Machine Studies* 6 (3) (1974) 361–376.

[49] M.V. Zelkowitz and D. Wallace, Experimental validation in software engineering, *Journal of Information and Software Technology* vol. 39 (1997) 735–743.

[50] M.V. Zelkowitz and D. Wallace, Experimental models for validating technology, *Theory and Practice of Object Systems*, 31 (5) (1998) 23–31.

[51] A. Zendler, A preliminary software engineering theory as investigated by published experiments, *Empirical Software Engineering* 6 (2) (2001) 161–180.

[52] G.H. Zimney, *Method in Experimental Psychology,* Ronald Press, New York, 1961.

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

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 based on 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 enhanced by also reporting effect sizes and confidence intervals.

Keywords: Empirical software engineering, controlled experiment, systematic review, statistical power, effect size.

1 Introduction

An important use of statistical significance testing in empirical software engineering (ESE) research is to test hypotheses in controlled experiments. An important component of such testing is the notion of *statistical power*, which is defined as the probability that a statistical test will correctly reject the null hypothesis [12]. 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.

Knowledge of statistical power can influence both 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 the outcomes of the study will likely be insignificant. Furthermore, if the study fails to provide information about the statistical power of its tests, we cannot determine whether the insignificant results were due to insufficient power or if the phenomenon 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 satisfactory level of statistical power, the research is probably not worth the time, effort, and cost of inferential statistics." ([1], p. 96).

These considerations have prompted researchers in disciplines such as social and abnormal psychology [8,10,38], applied psychology [6,30], communication [7], behavioral accounting [2], marketing [37], management [5,16,25,30], international business [4], and information systems research [1,36] to determine the *post hoc* statistical power of their respective literature.

Within software engineering (SE), Miller *et al*. [29] discussed the role of statistical power analysis in ESE research, suggesting that there is inadequate reporting and attention afforded 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 (p.286).*

Although Miller *et al*. [29] 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. There is, therefore, a need 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.

The purpose of this paper is thus (1) to perform a systematic review and quantitative assessment of the statistical power of ESE research in a sample of published controlled experiments, (2) to discuss the implications of these findings, and (3) to discuss techniques that ESE researchers can use to increase the statistical power of their studies in order to improve the quality and validity of ESE research.

In section 2, we present a brief background on statistical power and its determinants. In Section 3, we provide an overview of the research method employed to review and determine the statistical power in controlled software engineering experiments. Section 4 reports the results of the review, while Section 5 provides a discussion of the results, their implications, and some recommendations that should improve the quality and validity of future ESE research. Section 6 provides some concluding comments.

2 Background: statistical power

2.1 Power and errors in statistical inference

According to Neyman and Pearson's [31,32] method of statistical inference, testing hypotheses requires that we specify an acceptable level of statistical error, or the risk we are willing to take regarding the correctness of our decisions. Regardless of which decision rule we select, there are generally two ways of being correct and two ways of making an error in the choice between the null (H_0) and the alternate (H_A) hypotheses (see Table 1).

A Type I error is the error made when H_0 (the tested hypothesis) is wrongly rejected. In other words, a Type I error is committed whenever the sample results fall into the rejection region, even though H_0 is true. Conventionally, the probability of committing a Type I error is represented by the level of statistical significance, denoted by the lowercase Greek letter alpha (α). Conversely, the probability of being correct, given that H_0 is true is equal to $1-\alpha$.

Table 1. Ways of being correct or making an error when choosing between two competing hypotheses.

		Unknown true state of nature	
		H_0: No Difference	H_A: Difference
Statistical conclusion	Accept H_0	1–α: Correct	β: Type II error
	Reject H_0	α: Type I error	1–β: Correct (power)

The probability of making an error of Type II, also known as beta (β), is the probability of failing to reject the null hypothesis when it is actually false. Thus, when a sample result does not fall into the rejection region, even though some H_A is true, we are led to make a Type II error. Consequently, the probability of correctly rejecting the null hypothesis, i.e., the probability of making a correct decision given that H_A is true, is 1–β; the *power* of the statistical test. It is literally the probability of finding out that H_0 is wrong, given the decision rule and the true H_A.

As can be seen from Table 1, statistical power is particularly important when there is a true difference in the population. In this situation, when the phenomenon actually exists, the statistical test must be powerful enough to detect it. If the test reveals a non-significant result in this case, the conclusion of "no effect" would be misleading and we would thus be committing a Type II error.

Traditionally, α is set to .05 to guard against Type I error, while β is set to .20 to guard against Type II error. Accepting these conventions also means that we are guarded four times more against Type I errors than we are against Type II errors. However, the distribution of risk between Type I and Type II errors need to be appropriate to the situation at hand. An illustrative case is made by Mazen *et al*. [25] regarding the ill-fated Challenger space shuttle, in which NASA officials faced a choice between two types of assumptions, each with a distinctive cost:

> *The first [assumption] was that the shuttle was unsafe to fly because the performance of the O-ring used in the rocket-booster was different from that used on previous missions. The second was that the shuttle was safe to fly because there would be no difference between the performance of the O-rings in this and previous missions. If the mission had been aborted and the O-ring had indeed been functional, Type I error would have been committed. Obviously the cost of the Type II error, launching with a*

defective O-ring, was much greater than the cost that would have been incurred with Type I error (ibid., p. 370).

2.2 Determinants of statistical power

The fundamental approach to statistical power analysis was established by Cohen [12], who described the relationships among the four variables involved in statistical inference: significance criterion (α), sample size (*N*), population effect size (*ES*), and statistical power (1–β). For any statistical model, these relationships are such that each is a function of the other three. Thus, we can determine the power for any statistical test, given α, *N*, and *ES* (Table 2).

The appropriate sections of Cohen [12] or Kraemer and Thiemann [21] should be consulted for details on how to perform statistical power analysis. Specifically, Chapter 12 in Cohen's book provides the computational procedures that are used to determine the power and sample size values of the commonly used power tables and power charts.

As mentioned, the significance criterion (α) is the probability of incorrectly rejecting the null hypothesis. Power increases with larger α. A small α will, thus, result in relatively small power. The directionality of the significance criterion also affects the power of a statistical test. A non-directional two-tailed test will have lower power than a directional one-tailed test at the same α, provided that the sample result is in the predicted direction. Note that a directional test has no power to detect effects in the direction opposite to the one predicted (see Figure 1).

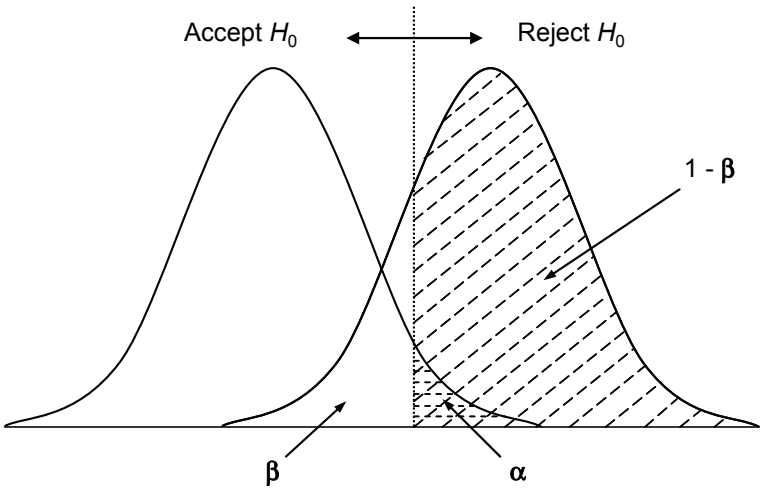

Figure 1: Statistical power and the probability of Type I and Type II error in testing a directional hypothesis.

The second determinant of power is sample size (*N*). At any given α level, increased sample size reduces the standard deviations of the sampling distributions for H_0 and H_A. This reduction results in less overlap of the distributions, increased precision, and thus increased power (see Figure 1).

The final determinant of power is the effect size (*ES*), which refers to the true size of the difference between H_0 and H_A (the null hypothesis is that the effect size is 0), i.e., the degree to which the phenomenon is present in the population. The larger the effect size, the greater the probability that the effect will be detected and the null hypothesis rejected.

The nature of the effect size will vary from one statistical procedure to the next (e.g., a standardized mean difference or a correlation coefficient), but its function in power analysis is the same in all procedures. Thus, each statistical test has its own scale-free and continuous effect size index, ranging upward from zero (see Table 3). So, whereas *p* values reveal whether a finding is *statistically* significant, effect size indices are measures of *practical* significance 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* [13,23].

Effect size is probably the most difficult aspect of power analysis to specify or estimate. It can sometimes be determined by a critical assessment of prior empirical research in the area. However, due to a lack of empirical studies and cumulative findings in software engineering, the best option for a reasonable estimation of effect size is expert judgment [29].

Cohen [12] has facilitated such estimation of effect size. Based on a review of prior behavioral research, he developed operational definitions of three levels of effect sizes (small, medium, and large) with different quantitative levels for the different types of statistical test. In information systems (IS) research and in the behavioral sciences, the operationalized definitions of the effect size for each of these categories have become a research standard for the most commonly used statistical tests [1,36].

Table 2. Determinants of statistical power.

Significance criterion (α)	The chosen risk of committing a Type I error (e.g. $\alpha = 0.05$).
Sample size (N)	The total number of subjects included in the analysis of data.
Effect size (ES)	The magnitude of the effect under the alternate hypothesis (e.g. $d = 0.5$).

Table 3. Effect-size indexes and their values for small, medium, and large effects for the most common statistical tests ([13], p. 157).

Statistical Test	Effect-Size Index	Effect Size		
		Small	Medium	Large
1. The *t*-test for the difference between two independent means	$d = \frac{m_A - m_B}{\sigma}$	.20	.50	.80
2. The *t*-test for the significance of a product-moment correlation coefficient, *r*	R	.10	.30	.50
3. The test for the difference between two independent *r*s	$q = z_A - z_B$	.10	.30	.50
4. The normal curve test for the difference between two independent proportions	$h = \phi_A - \phi_B$	.20	.50	.80
5. The chi-square test for goodness of fit (one-way) or association in two-way contingency tables	$w = \sqrt{\sum_{i=1}^{k} \frac{(P_{1i} - P_{0i})^2}{P_{0i}}}$	.10	.30	.50
6. One-way analysis of variance	$f = \frac{\sigma_m}{\sigma}$	.10	.25	.40
7. Multiple and multiple partial correlation	$f^2 = \frac{R^2}{1 - R^2}$	.02	.15	.35

Cohen established these conventions in 1977 [11], and they have been fixed ever since. His intent was that "medium [effect size] represents an effect likely to be visible to the naked eye of a careful observer ... small [effect size] to be noticeably smaller than medium but not so small as to be trivial, and ... large [effect size] to be the same distance above medium as small was below it." ([13],p.156). Table 3 gives the definition of the ES indices and the corresponding ES values for the most common statistical tests. These ES values enable the comparison of power levels across studies in this survey, as well as across surveys conducted in other disciplines. As an example, the ES index for the *t*-test of the difference between independent means, *d*, is the difference expressed in units of the within-population standard deviation. For this test, the small, medium, and large ESs are,

respectively, d = .20, .50, and .80. Thus, an operationally defined medium difference between means is half a standard deviation.

3 Research Method

We assessed all the 103 papers on controlled experiments (of a total of 5453 papers), identified by Sjøberg *et al*. [40], published in nine major software engineering journals and three conference proceedings during the decade 1993-2002 (Table 4). These journals and conference proceedings were chosen because they were considered to be representative of ESE research. Furthermore, since controlled experiments are empirical studies that employ inferential statistics, they were considered a relevant sample in this study.

Since the term "experiment" is used inconsistently in the software engineering community (often being used synonymously with empirical study), we use the term "controlled experiment". A study was defined as a controlled experiment if individuals or teams (the experimental units) conducted one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages, or tools (the treatments). We did not distinguish between randomized experiments and quasi-experiments in this study, because both designs are relevant to ESE experimentation.

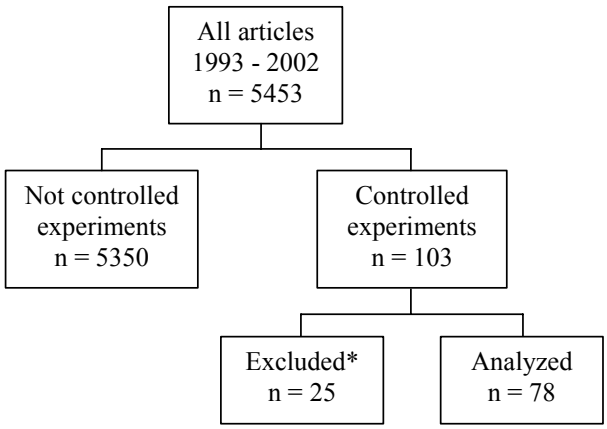

Figure 2: Results of the literature review. *25 articles were excluded due to duplicate reporting, no statistical analysis or unspecified statistical tests.

We excluded several types of study that share certain characteristics with experiments. While these might be highly relevant for the field, they are not controlled experiments as defined above. Thus, we excluded correlation studies, studies that are based solely on calculations on existing data, and simulated team evaluations that use data for individuals.

Studies that used projects or companies as treatment groups, in which data was collected at several levels (treatment defined, but no experimental unit defined) were also excluded because we consider these to be multiple case studies [43].

In order to identify and extract controlled experiments, one researcher systematically read the titles and abstracts of the 5453 scientific articles. 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 complete article was read by two researchers.

These criteria were met by 103 articles, which reported 113 experiments, (Table 4). All of them involved a number of significance tests. However, not all of these were equally relevant to the hypotheses of the studies. In fact, it was not always clear from the reporting of the studies which hypotheses were actually tested or which significance tests corresponded to which hypotheses.

Table 4. 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 (TOSEM)	1	1.0
Software: Practice and Experience (SP&E)	–	–
IEEE Computer	–	–
TOTAL:	103	100%

The first two authors read all 103 articles in detail and made separate extractions of the power data. Based on these two data sets, all three authors reviewed all tests in all experiments to reach a consensus on which experiments and tests to include. For 14 experiments, no statistical analysis was performed and for seven experiments, we did not manage to track which tests answered which hypothesis or research question. Five experiments were reported in more than one article. In these cases, we included the one most recently published. This assessment resulted in 78 articles (Figure 2). Of these articles, we identified 459 statistical tests corresponding to the main hypotheses or research questions of 92 experiments.

Similar to the methodology used by Baroudi and Orlikowski for MIS research [1], both parametric and nonparametric tests of the major hypotheses were included in this study. Table 5 shows the distribution of the 459 statistical tests in the final sample for which statistical power could be determined *post hoc*. The main parametric tests were Analysis of Variance (ANOVA) and *t*-tests. The main nonparametric tests were Wilcoxon, Mann-Whitney, Fisher's exact test, Chi-square, and Kruskall-Wallis. Other tests include Tukey's pairwise comparison (18), nonparametric rank-sum test (6), Poisson (3), regression (3), Mood's median test (2), proportion (2), and Spearman rank correlation (2).

The power of the nonparametric tests was determined by using analogous parametric tests where appropriate [9,10,18,21]. For example, the *t*-test for means approximates to the Mann-Whitney *U* test and the Wilcoxon rank test, the parametric *F* test to the Kruskal-Wallis *H* test, and Pearson's *r* to the Spearman Rank Correlation. Chi-square approximations were not needed since Cohen provided separate tables to determine its power.

Following the *post hoc* method, the power of each test was determined by using the stated sample size, setting the α level to the conventional level of .05, and choosing the nondirectional critical region for all power computations. Furthermore, power was calculated in relation to Cohen's definitions of small, medium, and large effect sizes [12]. This is similar to that of past surveys of statistical power in other disciplines, such as IS research [1,36]. All power calculations were made using SamplePower 2.0 from SPSS[8].

[8] See www.spss.com/samplepower/

Table 5. Distribution of statistical tests employed in 92 controlled SE experiments.

Statistical test	Number	Percent
ANOVA	179	39.0
t-test	117	25.5
Wilcoxon	41	8.9
Mann-Whitney	39	8.5
Fisher's exact test	15	3.3
Chi-square	14	3.1
Kruskall-Wallis	8	1.7
Other tests	46	10.0
TOTAL:	459	100%

4 Results

The 78 articles selected for this study with available data for calculating power yielded 459 statistical tests of the major hypotheses being investigated in the 92 reported controlled experiments. Table 6 shows the distribution of sample size for the experiments by type of statistical test. On average, the statistical tests covered 55 observations. However, the high standard deviation for several of the tests reveals a large amount of variation in sample sizes. For example, among the ANOVA subsample the average sample size was 79, yet 165 of the 179 tests examined had an average sample size of 50, while the remaining 14 tests had an average of 450. Similarly, for the Chi-square subsample the average sample size was 126. However, two of the tests had a sample size of 531 observations, while the average sample size of the remaining 12 tests was 58 observations. Also, in the group of other tests, with an average sample size of 39 observations, the three regression tests had a sample size of 242 observations, while the average sample size for the remaining 43 tests was 25 observations.

Table 6. Distribution of sample sizes (observations) occurring in 92 controlled SE experiments.

Statistical test	Mean	Std.	Min	Median	Max
ANOVA	79	118	6	65	800
t-test	34	29	5	30	136
Wilcoxon	40	23	10	34	78
Mann-Whitney	34	13	6	32	66
Fisher's exact test	40	27	16	20	74
Chi-square	119	180	10	30	531
Kruskall-Wallis	26	19	15	15	69
Other	38	57	10	16	242
TOTAL:	55	87	5	34	800

Several of the experiments surveyed in this study used within-subject designs so that each subject contributed several observations to the sample size of a statistical test. The most extreme cases were as follows: one study that used 800 observations from 100 subjects for an ANOVA test; another study that used 564 observations from 94 subjects for an ANOVA test; and yet another study that used 531 observations from 266 subjects in a Chi-square test. The latter study was also the one with the highest number of subjects in our sample.

So, while the average sample size of all 459 statistical tests in this study was 55 observations, with a standard deviation of 87, the median sample size was as low as 34 observations. Correspondingly, the average number of subjects in the surveyed experiments was 48, with a standard deviation of 51 and a median of 30. As a comparison, the average sample size of all tests in Rademacher's power study in IS research was 179 subjects (with a standard deviation of 196) [36].

Table 7 presents the power distribution of the 459 statistical tests in the 92 experiments using Cohen's conventional values for small, medium, and large effect sizes (see Table 3).

Small effect size: The average statistical power of the tests when we assumed small effect sizes was as low as .11. This means that if the phenomena being investigated exhibit a small effect size, then, on average, the SE studies examined have only a one in ten chance of detecting them. Table 7 shows that only one test is above the .80 conventional

power level and that 97% have a less than 50 percent chance of detecting significant findings.

Medium effect size: When we assume medium effect sizes, the average statistical power of the tests increases to .36. Although this is an improvement over the .11 power level achieved by tests of small effect sizes, the studies only have, on average, just about a one-third chance of detecting phenomena exhibiting a medium effect size. Table 7 indicates that only 6% of the tests examined achieve the conventional .80 power level or better, and that 78% of the tests have a less than 50 percent chance of detecting significant results.

Large effect size: Assuming large effect sizes, the average statistical power of the tests increases further, to .63. This means that, on average, the studies still have slightly less than a two-thirds chance of detecting their phenomena. As can be seen from Table 7, 31% of the tests attain or exceed the .80 power level, and 70% obtain a greater than 50 percent chance of correctly rejecting their null hypotheses. Thus, even when we assume that the effect being studied is so large as to make statistical testing unnecessary, as much as 69% of the tests fall below the .80 level.

Table 8 presents the power of the studies by type of statistical test employed. None of the tests reaches the conventional .80 power level; not even when we assume large effect sizes. ANOVA and *t*-tests account for almost two-thirds of all statistical analyses in controlled SE experiments, yet their mean power level for detecting large effect sizes is only .67 and .61 respectively, while the corresponding power levels assuming medium effect sizes are as low as .40 and .33.

In summary, this quantitative assessment revealed that controlled SE experiments, on average, only have a two-thirds chance of detecting phenomena with large effect sizes. The corresponding chance of detecting phenomena with medium effect sizes is around one in three, while there is only a one in ten chance of detecting small effect sizes.

Finally, a qualitative assessment of the treatment of power within the sampled studies revealed an interesting pattern. Of the 78 papers in our sample, 12 discussed the statistical power associated with the testing of null hypotheses. Of these studies, nine elaborated on the specific procedures for determining the statistical power of tests. Three of the nine performed *a priori* power analysis, while six performed the analysis *a posteriori*. Only one of the papers that performed an *a priori* power analysis used it to guide the choice of sample size. In this case, the authors explicitly stated that they were only interested in large effect sizes and that they regarded a power level of 0.5 as sufficient. Still, they included

Table 7. Frequency and cumulative percentage distribution of power in 92 controlled SE experiments.

	Small effect size		Medium effect size		Large effect size	
Power level	Frequency	Cum. %	Frequency	Cum. %	Frequency	Cum. %
.91 - .99	–	–	18	100	69	100
.81 - .90	1	100	11	96	75	85
.71 - .80	–	100	14	94	49	69
.61 - .70	2	100	13	91	70	58
.51 - .60	9	99	44	88	58	43
.41 - .50	2	97	50	78	21	30
.31 - .40	–	97	76	67	43	25
.21 - .30	13	97	107	51	43	16
.11 - .20	120	94	94	27	31	7
.00 - .10	312	68	32	7	–	–
TOTAL:	459	–	459	–	459	–
Average power:	0.11		0.36		0.63	

Table 8. Power analysis by type of statistical test in 92 controlled SE experiments.

	Small effect size		Medium effect size		Large effect size	
Statistical test	Means	Std. Dev.	Means	Std. Dev.	Means	Std. Dev.
ANOVA	.12	.11	.40	.24	.67	.28
t-test	.10	.03	.33	.17	.61	.23
Wilcoxon	.12	.05	.46	.24	.74	.24
Mann-Whitney	.09	.02	.29	.10	.59	.19
Fisher's exact test	.06	.05	.25	.22	.49	.34
Chi-square	.18	.20	.43	.33	.64	.28
Kruskall-Wallis	.09	.02	.31	.15	.59	.28
Other	.10	.11	.26	.25	.44	.24

so few subjects in the experiment that the average power to detect a large effect size of their statistical tests was as low as 0.28. Of the six papers that performed *a posteriori* power analysis, two gave recommendations for the necessary sample sizes in future replication studies. Thus, overall, 84.6% of the sampled experimental studies did not reference the statistical power of their significance tests.

5 Discussion

In this section, we discuss the implications of the findings in this study for the interpretation of experimental SE research. We suggest several ways to increase statistical power, and we provide recommendations for future research. First, however, we compare the main findings in the current study with the related discipline of IS research.

5.1 Comparison with IS research

We compared the results of the current study with two corresponding reviews of the statistical power levels in IS research [1,36]. In the former study, 63 statistically-based studies were identified from the issues of *Communications of the ACM*, *Decision Sciences*, *Management Science*, and *MIS Quarterly* over the five-year period from January 1980 to July 1985. The final sample included 149 statistical tests from 57 studies. In the latter study, 65 statistically-based studies that employed 167 statistical tests were selected from *MIS Quarterly* over the seven-year period from January 1990 to September 1997. In comparison, the current study included 92 controlled experiments that comprised 459 statistical tests published in nine major software engineering journals and three conference proceedings during the decade 1993-2002 (see Tables 4 and 5).

Statistical power in the two IS research studies and the current SE research study for small, medium, and large effect sizes are compared in Table 9. The results of the two IS studies indicate that the power levels for all effect sizes have improved substantially in the decade between the two studies. Furthermore, the results show that IS research now meets the desired power level of .80 specified by Cohen [12] for medium effect sizes, which is assumed as the target level by most IS researchers [36].

Table 9: Comparison of current survey with statistical power values in prior IS research.

		Means for different effect-size assumptions		
Related IS study	No. of Articles	Small	Medium	Large
Baroudi and Orlikowski [1]	57	.19	.60	.83
Rademacher [36]	65	.34	.81	.96
Current study	78	.11	.36	.63

The results of the current study show that the power of experimental SE research falls markedly below the levels attained by IS research. One reason for this difference might be that the IS field has benefited from the early power review of Baroudi and Orlikowski [1], and thus explicit attention has been paid to statistical power, which has paid off with contemporary research displaying improved power levels, as demonstrated by Rademacher [36]. What is particularly worrying for SE research is that the power level displayed by the current study not only falls markedly below the level of 1999 study by Rademacher, but that it also falls markedly below the level of the 1989 study by Baroudi and Orlikowski.

While medium effect sizes are considered the target level in IS research [36], and the average power to detect these effect sizes are .81 in IS research, Table 7 indicates that only 6% of the tests examined in the current research achieve this level, and that as much as 78% of the tests in the current research have a less than 50 percent chance of detecting significant results for medium effects. Unless it can be demonstrated that medium (and large) effect sizes are irrelevant to SE research, this should be a cause for concern for SE researchers and practitioners. Consequently, we should explore in more depth what constitutes meaningful effect sizes within SE research, in order to establish specific SE conventions.

A comparison of power data for the two most popular types of statistical test in experimental SE research, with the corresponding tests in IS research, is provided in Table 10. As can be seen from Table 5, these tests (ANOVA and *t*-test) constitute about two-thirds of the statistical tests in our sample. The results show that, on average, IS research employ sample sizes that are twice as large as those found in SE research for these tests. In fact, the situation is a little worse than that, since *observations* are used as the sample size in the current study, while the IS studies refer to *subjects*. Moreover, the power levels of

the current study to detect medium effect sizes are only about half of the corresponding power levels of IS research.

Table 10: Comparison of the two most popular types of tests in the current survey with corresponding power data for IS research.

Statistical test	Baroudi and Orlikowski [1]	Rademacher [36]	Current study
ANOVA (medium effect size):			
Sample size[1]	64	136	79
Power (mean value)	.56	.82	.40
Power (std. deviation)	.30	.19	.24
t-test (medium effect size):			
Sample size[1]	45	70	34
Power (mean value)	.53	.74	.33
Power (std. deviation)	.27	.18	.17

[1]Note that sample size in the two IS studies refers to subjects, while in the current study it refers to observations.

5.2 Implications for interpreting experimental SE research

An important finding of this study is that explicit consideration of power issues, e.g., in terms of discussion, use, and reporting of statistical power analysis, in experimental SE research is very limited. As mentioned above, 15.4% of the papers discussed statistical power in relation to their testing of the null hypothesis, but in only one paper did the authors perform an *a priori* power analysis. In addition, and perhaps as a consequence, the *post hoc* power analyses showed that, overall, the studies examined had low statistical power. Even for large effect sizes, as much as 69% of the tests fell below the .80 level. This implies that considerations of statistical power are underemphasized in experimental SE research.

Two major issues that are particularly important for experimental SE research arise from this underemphasis of statistical power: (1) the interpretation of results from individual studies and (2) the interpretation of results from the combination or replication of empirical studies [22,24,27,29,35]. As mentioned above, a test without sufficient statistical power will not provide the researcher with enough information to draw

conclusions regarding the acceptance or rejection of the null hypothesis. If no effects are detected in this situation, researchers should not conclude that the phenomenon does not exist. Rather, they should report that no significant findings were demonstrated in their study, and that this may be due to the low statistical power associated with their tests.

Another issue regarding the interpretation of results from individual studies with low power is the use of multiple tests. In this case, which included 91.3% of the experiments, the probability of obtaining at least one statistically significant effect might be large, even if the probability that any specific effect is statistically significant is small (see [28]). As an example, recall from Table 7 that the probability that a medium effect size is statistically significant is only .36. At the same time, the 84 experiments in this study with more than one test had an average of 5.4 tests per experiment. Thus, with this number of tests, we would expect about two statistically significant results for medium effect sizes in each of the experiments in this study. So, although power is sufficient for attaining statistical significance somewhere, it is not sufficient for any specific test. Again, this inadequate power for testing specific effects makes it difficult to interpret properly the results of any single study. It would be helpful, therefore, if researchers reporting results from statistical hypothesis testing were to distinguish between the tests of primary and secondary hypotheses.

Low statistical power also has a substantial impact on the ability to replicate experimental studies based on null hypothesis testing. Ottenbacher nicely demonstrates an apparent paradox that results from the replication of such low powered studies [34], showing that:

> *... the more often we are well guided by theory and prior observation, but conduct a low power study, the more we decrease the probability of replication! Thus a literature with low statistical power is not only committing a passive error, but can actually contribute to diverting attention and resources in unproductive directions (ibid., 273).*

Consequently, the tendency to underpower SE studies makes replication and meta-analysis troublesome, and will tend to produce an inconsistent body of literature, thus hindering the advancement of knowledge.

The results of our review also raise another important issue: the interpretation of studies with very high levels of power. Some of the studies in this review employed large

sample sizes, ranging from 400 to 800 observations. This poses a problem for interpretation, because virtually any study can be made to show significant results if the sample size is large enough, regardless of how small the true effect size may be [18]. Hence, it is of particular importance that researchers who report statistically significant results from studies with very large sample sizes, or with very large power levels, also report the corresponding effect sizes. This will put the reader in a better position to interpret the results and judge whether the statistically significant findings have practical importance.

5.3 Ways to increase statistical power

Increase the size of the sample: The most obvious way to increase the statistical power of a study is to increase the size of the sample. However, there is invariably some cost in terms of time, effort, and money per subject that must be considered. With this in mind, most researchers try to use the smallest number of subjects necessary to have a reasonable chance of obtaining significant results with a meaningful effect size [9]. However, while using only a few subjects may result in meaningful effects not being detected, trivial effects may show up as significant results when the sample size is very large. Consequently, if the researcher wants significance to reflect a sizable effect and also wants to avoid being led into a blind alley by a significant result, attention should be paid to both aspects of sample size. As a general rule, the sample size should be large enough to give confidence that meaningful effects will be detected. At the same time, the reporting of effect sizes will ensure that trivial associations will be detected even though they might be statistically significant.

Relax the significance criterion: Power can also be increased by relaxing the significance criterion. This approach is not common, however, because of widespread concern about keeping Type I errors to a fixed, low level of, e.g., .01 or .05. Still, as the example of the Challenger space shuttle showed, the significance criterion and the power level should be determined by the relative seriousness of Type I and Type II errors. Thus, researchers should be aware of the costs of both types of errors when setting the alpha and power levels, and must make sure that they explain the consequences of the raised probability of Type I errors if they relax the significance criterion. When possible, researchers should analyze the relative consequences of Type I and Type II errors for the specific treatment situation under investigation.

Choose powerful statistical tests: In general, parametric tests are more powerful than their analogous nonparametric test [21]. Thus, the power of a study can most often be increased by choosing an appropriate parametric test. It is important to note, however, that these tests make a number of assumptions about the properties (parameters) of the populations, such as the mean and standard deviation, from which samples are drawn. On the other hand, given the empirical evidence for the robustness and enhanced power provided by parametric tests, "researchers are encouraged to use the parametric test most appropriate for their study and resort to non-parametric procedures only in the rare case of extreme assumption violations" ([1], p. 98).

The power of a test can also be increased by retaining as much information as possible about the dependent variable. In general, tests comparing data categorized into groups are less powerful than tests using data measured along a continuum. As Baroudi and Orlikowski recommend [1], "statistics that permit continuous data to be analyzed in continuous form, such as regression, should be used over those that require data to be divided in groups, such as the analysis of variance" (p. 99).

Furthermore, as we have already noted, the direction of the significance criterion also affects the power of a statistical test. A directional, one-tailed test will yield higher power than a non-directional two-tailed test at the same alpha level, provided that the sample results are in the predicted direction. Note, however, that a directional test has no power to detect effects in the direction opposite to that predicted. Thus, the primary guide for the researcher deciding whether a hypothesis should be tested with a directional or non-directional test must be the comparative term of the original research question.

Reduce measurement error and subject heterogeneity: The larger the variance on the scores within the treatment and control groups, the smaller the effect size and the power will be. One source of such variance is measurement error, i.e. variability in scores that is unrelated to the characteristic being measured. Another source is the heterogeneity of subjects on the measure [23]. Thus, anything that makes the population standard deviation small will increase power, other things being equal.

In general, subject heterogeneity can be reduced by selecting or developing measures that do not discriminate strongly among subjects. If the measure, nevertheless, does respond substantially to subject differences, these could be reduced statistically during data analysis. To reduce such variance, and thus increase statistical power, the researcher can utilize a repeated measures or paired subjects design, or a factorial design that employs blocking, stratification, or matching criteria [39]. Researchers can also reduce subject

heterogeneity by employing a research design that covaries a pretest measure with the dependent variable [14].

Measurement error can be reduced by exercising careful control over experimental subjects and conditions. In addition, the researcher can use some form of aggregation, or averaging, of multiple measures that contain errors individually, to reduce the influence of error on the composite scores [33,41]. So, whenever applicable, the researcher should use reliable, multi-item measures to increase power [15].

Balance groups: The statistical power of a study is based less on the total number of subjects involved than on the number in each group or cell within the design. In addition, because the power of a test with unequal group sizes is estimated using the harmonic mean [12], the "effective" group size is skewed toward the size of the group with the fewest subjects. Thus, with a fixed number of subjects, maximal statistical power is attained when they are divided equally into treatment and control groups [23]. Researchers should, therefore, try to obtain equal, or in the case of factorial designs, proportional, group sizes rather than getting a large sample size that results in there being unequal or disproportional groups [1].

Investigate only relevant variables: One of the best strategies for increasing statistical power is to use theory and prior research to identify those variables that are most likely to have an effect [23]. Careful selection of which independent variables to include and which variables to exclude is, thus, crucial to raising the power of a study and the legitimacy of its potential findings. Kraemer and Thiemann suggested that only factors that are absolutely necessary to the research question, or that have a documented and strong relationship to the response, should be included in a study [21]. Accordingly, they recommended "Choose a few predictor variables and choose them carefully." (p. 65), or as McClelland put it [26]: "Doubling one's thinking is likely to be much more productive than doubling one's sample size." (p. 964).

In summary, when criterion significance and power levels are set, and a threshold for the minimum effect size to be detected has been decided, the two primary factors for consideration in a power analysis are the operative effect size and the sample size. Since much of what determines effect size has to do with the selection of measures, statistical analysis, treatment implementation, and other issues that are intrinsic parts of the research design, effect size enhancements are, generally, more cost-effective to engineer than are sample size increases [23]. However, determining how best to enhance the effect size requires some analysis and diagnosis of these factors for the particular research situation at

hand. A tactic that is almost always effective, though, is procedural and statistical variance control. Procedural variance control means tight standardization of treatment and control conditions, sampling, and measurement, while statistical variance control uses such techniques as covariates or blocking factors to separate variance judged irrelevant to the assessment of treatment effects from the error term for significance testing (see above). As shown by Lipsey [23], such techniques can sometimes increase the operative effect size two or threefold or even more.

Thus, when designing SE experiments, the goal should be to obtain the largest possible effect size with the smallest investment in the number of subjects studied. This presupposes that the researcher understands the factors that influence statistical power and skilfully applies that knowledge in the planning and implementation of each study undertaken. For a more in-depth treatment of these issues, see Lipsey's excellent work on design sensitivity to the statistical power of experimental research [23].

5.4 Limitations

The main limitations of this study are publication selection bias and inaccuracy in data extraction. As the basis for our investigation was the recent survey of controlled SE experiments performed by [40], the current study has the same publication selection basis as the main study. However, we consider the 12 surveyed journals and conferences to be leaders in software engineering in general and empirical software engineering in particular. Besides, Sjøberg *et al*.'s selection of journals is a superset of those selected by others (e.g., [17,44]. Nevertheless, if the main study also had included the grey literature (theses, technical reports, working papers, etc.) on controlled SE experiments, the current study could, in principle, provide more data and possibly allow more general conclusions to be drawn [19]. Regarding the selection of articles, the main study utilized a multistage process involving several researchers who documented the reasons for inclusion/exclusion as suggested in [19] (see [40]).

As described in Section 3, the first two authors read all 103 articles included in the main study in detail and made separate extractions of the power data. Based on these two data sets, all three authors reviewed all tests in all experiments to reach a consensus on which experiments and tests to include. However, because it was not always clear from the reporting of the studies which hypotheses were actually tested, which significance tests

corresponded to which hypotheses, or how many observations that were included for each test, the extraction process may have resulted in some inaccuracy in the data.

5.5 Recommendations for future research

Based on the problems that we have identified that are associated with statistical power in experimental SE research, we offer some recommendations to SE researchers who perform null hypothesis testing.

First, before embarking on studies involving statistical inference, we recommend that SE researchers plan for acceptable power on the basis of attention to the effect size, either by assessing previous empirical research in the area and using the effect sizes found in these studies as a guide, or by looking at their own studies and pilot studies for guidance. However, due to the limited number of empirical studies in SE this approach may be difficult to apply [29]. Alternatively, researchers can use a judgmental approach to decide what effect size they are interested in detecting. However, until there is a better basis for establishing conventions specific to SE, we recommend the same general target level of medium effect sizes as used in IS research, determined according to Cohen's definitions [12].

Second, we recommend that SE researchers analyze the implications of the relative seriousness of Type I and Type II errors for the specific treatment situation under investigation. Unless there are specific circumstances, we do not recommend that researchers relax the commonly accepted norm of setting alpha to .05. Similarly, we recommend that SE researchers plan for a power level of at least .80 and perform power analyses accordingly. Thus, rather than relaxing alpha, we generally recommend increasing power to better balance the probabilities of committing Type I and Type II errors.

Third, in agreement with Kitchenham *et al*. [20] and Wilkinson *et al*. [42], we recommend that significance tests of experimental studies be accompanied by effect size measures and confidence intervals to better inform readers. In addition, studies should report the data for calculating such items as sample sizes, alpha level, means, standard deviations, statistical tests, the tails of the tests, and the value of the statistics.

Finally, we recommend that journal editors and reviewers pay closer attention to the issue of statistical power. This way, readers will be in a better position to make informed decisions about the validity of the results and meta-analysts will be in a better position to perform secondary analyses.

6 Conclusion

The purpose of this research was to perform a quantitative assessment of the statistical power of current experimental SE research. Since this is the first study of its kind in SE research, it was not possible to compare the statistical power data of the current study with prior experimental SE research. Therefore, we found it useful to draw on the related discipline of IS research, because this provided convenient baseline data for measuring and validating the results of the statistical power analysis of this research.

The results showed that there is inadequate attention to power issues in general, and that the level of statistical power in SE research falls substantially below accepted norms as well as below the levels found in the related discipline of IS research. For example, only six percent of the studies in this analysis had power of .80 or more to detect a medium effect size, which figure is assumed as the target level by most IS researchers.

In conclusion, attention must be directed to the adequacy of sample sizes and research designs in experimental SE research to ensure acceptable levels of power (i.e., $1-\beta \geq .80$), assuming that Type I errors are to be controlled at $\alpha = .05$. At a minimum, the current reporting of significance tests should be enhanced by reporting the effect sizes and confidence intervals to permit secondary analysis and to allow the reader a richer understanding of, and an increased trust in, a study's results and implications.

Acknowledgements

We are grateful to Jo E. Hannay, Ove Hansen, Amela Karahasanović, Nils-Kristian Liborg and Anette C. Rekdal for providing help in identifying the 103 papers on controlled experiments used in our review. Thanks to Chris Wright for proofreading the paper.

Appendix A: A numeric guide to sample size for the *t*-test

We assume that a researcher plans to test a non-directional hypothesis that two means do not differ by conducting a controlled experiment with one experimental and one control group. Such a study can be analyzed suitably with an unpaired *t*-test with two-tailed rejection regions.

The effect size index (*d*) under these circumstances can be calculated by

$$d = \frac{M_E - M_C}{\sigma}$$

where M_E is the mean score of the experimental group;

M_C is the mean score of the control group; and

σ is the standard deviation based on either group or both.

A small effect size would be $d = .2$, a medium effect size would be $d = .5$, while a large effect size would be $d = .8$.

The sample size[9] (*N*) required for each group as a function of effect size, alpha, and power is shown in *Table A.1*[10]. As an example, if the researcher wants to be able to detect a medium difference ($d = .5$) between the two independent means at $\alpha = .05$, a sample size of $N = 64$ is required in each group. Similarly, at the same alpha level, if the researcher has 60 subjects available for the experiment, a power level of .85 will be attained for detecting a large effect size. Alternatively, by relaxing the alpha level to .10, 30 subjects in each group would yield a power of .60 to detect a medium effect size.

[9] In fact, the samples size in the table represents the harmonic mean of the sample sizes in the treatment and control groups.

[10] Calculation of the sample sizes in *Table A.1* was made with SamplePower 2.0 from SPSS.

***Table A.1*: A numeric guide to sample size for small, medium, and large effects sizes for different values of α and power for a two-tailed *t*-test.**

	α = .01			α = .05			α = .10		
Power	d = .2	d = .5	d = .8	d = .2	d = .5	d = .8	d = .2	d = .5	d = .8
.95	893	145	58	651	105	42	542	88	35
.90	746	121	49	527	86	34	429	70	28
.85	655	107	43	450	73	30	361	59	24
.80	586	96	39	394	64	26	310	51	21
.75	530	87	35	348	57	23	270	44	18
.70	483	79	32	310	51	21	236	39	16
.65	441	72	30	276	45	19	207	34	14
.60	402	66	27	246	41	17	181	30	12
.55	367	61	25	219	36	15	158	26	11
.50	334	55	23	194	32	14	136	23	10

References

[1] J. Baroudi, W. Orlikowski, The problem of statistical power in MIS research, *MIS Quarterly* 13 (1) (1989) 87–106.

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

[3] J.K. Brewer, On the power of statistical tests in the American educational research journal, *American Educational Research Journal* 9 (1972) 391–401.

[4] J.K.U. Brock, The 'Power' of International Business Research, *Journal of International Business Studies* 34 (1) (2003) 90–99.

[5] L.H. Cashen, 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.

[6] L.J. Chase, R.B. Chase, A statistical power analysis of applied psychology research, *Journal of Applied Psychology* 6 (2) (1976) 234–237.

[7] L.J. Chase, R.K. Tucker, A power-analytic examination of contemporary communica-tion research, *Speech Monographs* 42 (1) (1975) 29–41.

[8] D. Clark-Carter, The account taken of statistical power in research published in the british journal of psychology, *British Journal of Psychology* 88 (1) (1997) 71–83.

[9] B.H. Cohen, *Explaining Psychological Statistics*, second ed., Wiley, New york, 2001.

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

[11] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, Academic Press, New York, 1977, Revised Edition.

[12] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, second ed., Laurence Erlbaum, Hillsdale, New Jersey, 1988.

[13] J. Cohen, A power prime, *Psychological Bulletin* 112 (1) (1992) 155–159.

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

[15] T. Dybå, An instrument for measuring the key factors of success in software process improvement, *Empirical Software Engineering* 5 (4) (2000) 357–390.

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

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

[18] W.L. Hays, Statistics, fifth ed., Harcourt Brace, New York, 1994.

[19] B.A. Kitchenham, *Procedures for performing systematic reviews*, Keele University Technical Report TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.

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

[21] H.C. Kraemer, S. Thiemann, *How Many Subjects? Statistical Power Analysis in Research*, Sage, Beverly Hills, 1987.

[22] R.M. Lindsay, A.S.C. Ehrenberg, The design of replicated studies, *The American Statistician* 47 (3) (1993) 217–228.

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

[24] S.E. Maxwell, The persistence of underpowered studies in psychological research: causes, consequences, and remedies, *Psychological Methods* 9 (2) (2004) 147–163.

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

[26] G.H. McClelland, Increasing statistical power without increasing sample size, *American Psychologist* 55 (8) (2000) 963–964.

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

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

[29] J. Miller, J. Daly, M. Wood, M. Roper, A. Brooks, Statistical power and its subcomponents—missing and misunderstood concepts in empirical software engineering research, *Information and Software Technology* 39 (4) (1997) 285–295.

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

[31] J. Neyman, E.S. Pearson, On the use and interpretation of certain test criteria for purposes of statistical inference, *Biometrika* 20A (1928) 175–240 263–294.

[32] J. Neyman, E.S. Pearson, On the problem of the most efficient tests of statistical hypotheses, *Transactions of the Royal Society of London* Series A 231 (1933) 289-337.

[33] J.C. Nunnally, I.A. Bernstein, *Psychometric Theory*, third ed., McGraw- Hill, New York, 1994.

[34] K.J. Ottenbacher, The power of replications and the replications of power, *The American Statistician* 50 (3) (1996) 271–275.

[35] L.M. Pickard, B.A. Kitchenham, P.W. Jones, Combining empirical results in software engineering, *Information and Software Technology* 40 (14) (1998) 811–821.

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

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

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

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

[40] D.I.K. Sjøberg, J.E.Hannay,O.Hansen,V.B.Kampenes,A.Karahasanovic´, N.-K. Liborg, A.C. Rekdal, A survey of controlled experiments in software engineering, *IEEE Transactions on Software Engineering* 31(9) (2005) 733–753.

[41] P. Spector, Summated rating scale construction: an introduction, Sage University Paper series on *Quantitative Applications in the Social Sciences*, 07–082, Sage, Newbury Park, California, 1992.

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

[43] R.K. Yin, *Case Study Research: Design and Methods*, Sage, Thousand Oaks, CA, 2003.

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

Paper 3:

A Systematic Review of Effect Size in Software Engineering Experiments

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

Information and Software Technology 49 (11-12) 1073-1086

Abstract

An effect size quantifies the effects of an experimental treatment. Conclusions drawn from hypothesis testing results 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 twelve 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 gives advice 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.

Keywords: Empirical software engineering; Controlled experiments; Effect size; Statistical significance; Practical importance.

1 Introduction

Software engineering experiments investigate the cause-effect relationships between treatments applied (process, method, technique, language, tool, etc.) and outcome variables measured (time, effectiveness, quality, efficiency, etc). An *effect size* is the magnitude of the relationship between treatment variables and outcome variables, and is computed on the basis of the sample data to make inferences about a population (analogously to the concept of hypothesis testing). An effect size tells us the degree to which the phenomenon under investigation is present in the population. There are several types of effect size measures[11], for example, correlations, odds ratios and differences between means.

Wrong or imprecise conclusions might be drawn from hypothesis testing results if effect sizes are not judged in addition to statistical significance. In particular, *p*-values are insufficient for decision-making; if an experiment includes a sufficient number of subjects, it is always possible to identify statistically significant differences, or if the experiment includes too few subjects (insufficient power), *p*-values may also be misleading. So, 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* [7, 27]. Hence, as also recommended by others [12, 23, 29], effect sizes should be part of experimental results in software engineering.

There is no unambiguous mapping from an effect size to a value of practical importance. Hence, observed effect sizes must be judged in context [2, 9, 18, 21, 35, 36, 41, 42, 45]. Even small effects might be of practical importance. For example, the optimization of a defect-detection method that yields only a one percent increase in error detection would be of little practical importance for most types of software, but might be of high practical importance for safety-critical software, particularly if the added one percent belongs to the most critical type of errors. This means that a contextual, subjective judgment of observed effect sizes must be made and a ritualized interpretation avoided. Hence, not only is the reporting of effect sizes important, but also a nuanced interpretation and discussion of those values.

Effect size estimation is not a new method. An approach to determining the magnitude of the effect of agricultural treatments was published seven decades ago [3], and reporting effect sizes in addition to statistical significance has been recommended for a long time in

[11] We will refer to specific values as *effect sizes*, and ways (formulae) to compute effect sizes as *effect size measures*.

behavioural science [4, 45]. Reporting effect sizes is also urged in medical science. A group of scientists and editors have developed the CONSORT statement to improve the quality of reporting of randomized clinical trials. One recommendation is that one should report "for each primary and secondary outcome, a summary of results for each group and the estimated effect size and its precision (e.g., 95% confidence interval)" [1]: p.682].

In addition to being meaningful in the analysis and reporting of experimental results, previously published effect sizes can be used in meta-analyses [17] and in statistical power analyses [5, 27], and for comparison purpose. Such use requires the reporting of either effect sizes, or sufficient data for effect size estimation.

This article reports on a systematic review of the literature on effect size issues in controlled experiments published in empirical software engineering. A total of 113 controlled experiments were reported in the decade from 1993-2002 in 12 leading journals and conference proceedings in software engineering [39]. Of these 113 experiments, this review investigates the 92 for which statistical hypothesis testing was performed and primary tests were identifiable. The aim of this review is to investigate the following:

- *The extent of effect size reporting and the interpretation of the effect sizes given by the authors of the reviewed experiments*, i.e., the extent to which effect sizes are used to describe the experimental result as a supplement to statistical significance, and when effect sizes are reported, how they are described and interpreted. This investigation is motivated by the belief that the use of effect sizes affects conclusions made from experiments.
- *The extent to which experimental results are reported in such a way that standardized effect sizes can be estimated.* This is an assessment of the completeness of the reporting of descriptive statistics. A complete reporting of descriptive statistics will allow the reader to verify the reporting of test results and effect size estimates, and to estimate effect sizes other than those reported.
- *The standardized effect sizes detected in software engineering experiments.* The rationale for this investigation is to provide an overview of effect sizes detected in software engineering experiments so that researchers can make relative comparisons of observed effect size estimates.

The remainder of this paper is organized as follows. Section 2 summarizes relevant concepts and measures of effect size. Section 3 describes the research method applied in this

review. Section 4 reports the results. Section 5 discusses the findings, the implications for power analysis, the limitations of the study, and presents guidelines for reporting effect sizes. Section 6 concludes.

2 Background: effect size

The effect that one inspection method has on the number of defects detected compared with another inspection method is an example of an effect in software engineering that we wish to investigate by conducting experiments. This unknown effect is referred to as the *population effect size*. It cannot be computed directly as long as we do not have access to the total population of subjects that falls within the scope of the research questions of our investigation. However, the population effect size may be estimated from sample data from a single experiment. *Estimated effect sizes* from several experiments can further be aggregated and analyzed to provide even stronger foundations for inferences about the population effect size (meta-analysis).

Figure 1 gives an overview of the effect size concepts described in the next sections. Measures of effect size can be classified as *standardized* or *unstandardized*. Standardized measures are scale-free because they are defined in terms of the variability in the data. Types of standardized measures of effect size are presented in Section 2.1. Unstandardized measures encompass all other types of effect size measures and will be described in Section 2.2.

2.1 Standardized effect size

Two families of standardized effect size measures are often referred to in the literature: the *d family* and the *r family*. Below, we will emphasize Hedges' *g* in the *d family* and the point-biserial correlation in the *r family*, because these are the two types applied in this review.

2.1.1 The d family

The *d family* consists of variations over standardized mean difference. Assume that we have two groups, Group 1 and Group 2. Moreover, assume that the experimental observations in Group 1, $y_{11,}\ldots,\ y_{1n}$, are normally distributed with mean μ_1 and variance σ^2, and the observations in Group 2, $y_{21},\ldots,y_{2m}$, are normally distributed with mean μ_2 and variance σ^2.

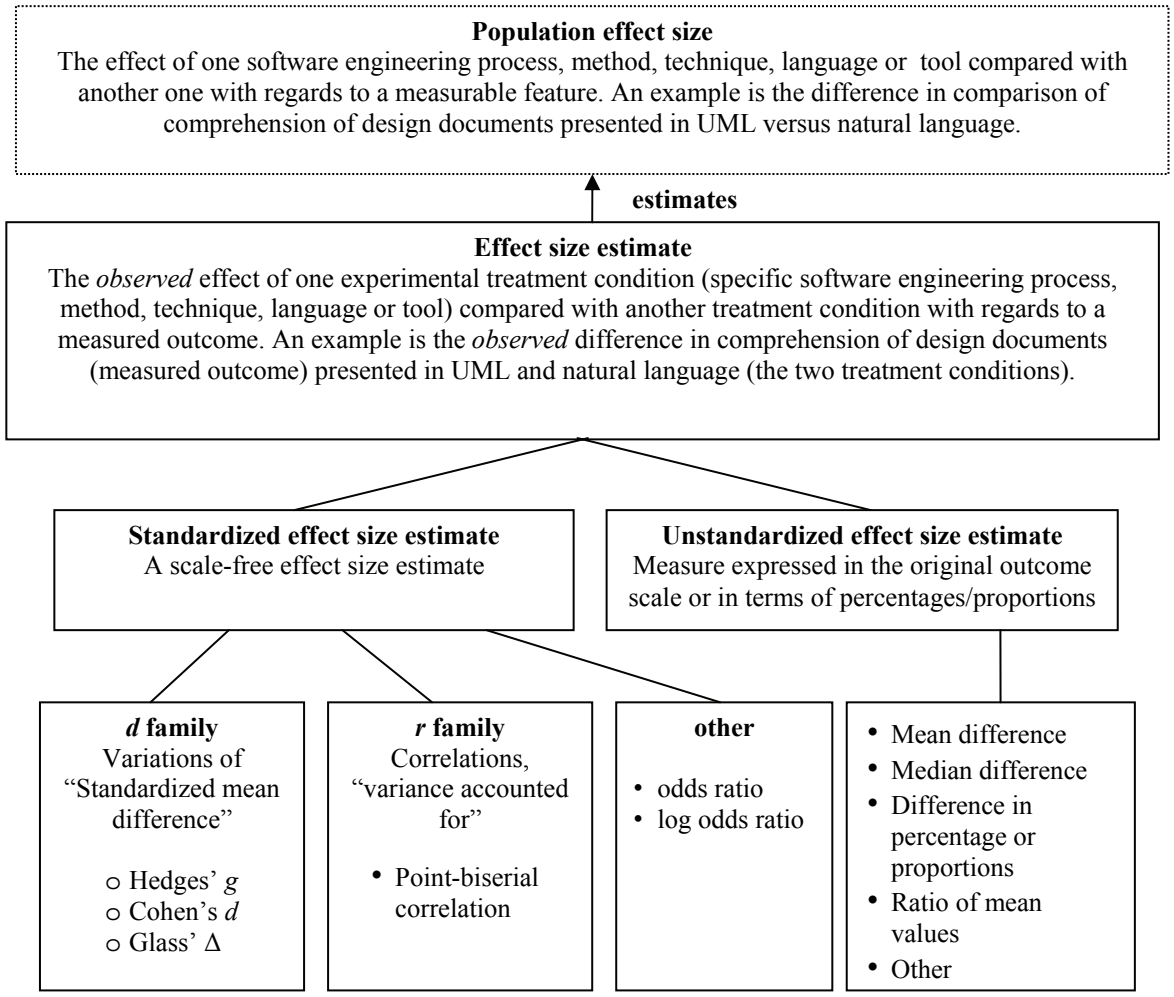

Figure 1. Population and estimated effect size as defined for software engineering and examples of types of effect size measures for the comparison of two treatment conditions.

More specifically:

$$Y_1 \sim N(\mu_1, \sigma^2)$$

and

$$Y_2 \sim N(\mu_2, \sigma^2)$$

The population standardized mean difference effect size measure, which we will call d_{pop}, is defined as

$$\textit{Population standardized mean difference,} \quad d_{pop} = \frac{\mu_1 - \mu_2}{\sigma} \tag{1}$$

The population standardized mean difference takes positive or negative values, depending on the choice of μ_1 and μ_2. It is estimated by the difference between sample means ($\overline{X}_1, \overline{X}_2$) divided by an estimate of population standard deviation. Different estimators of the population standard deviation give different effect size estimators. The three estimators most often referred to in the literature are Hedges' *g*, Cohen's *d* and Glass' Δ [24, 34]. Hedges' *g* has the pooled standard deviation, S_p, as the standardizer:

$$Hedge's\ g = \frac{\overline{X}_1 - \overline{X}_2}{s_p} \tag{2}$$

The pooled standard deviation is based on the standard deviations in both groups, s_1, s_2:

$$s_p = \sqrt{\frac{(n_1 - 1)\, s_1^2 + (n_2 - 1) s_2^2}{(n_1 - 1) + (n_2 - 1)}}, \tag{3}$$

Cohen's *d* also has the pooled standard deviation as its standardizer, but with n_i replacing $(n_i - 1)$ in Formula (3) and in the estimators of the single s_i. Glass' Δ applies the standard deviation in one group only; the one considered to be the control. According to [17], these three estimators have the same properties in large samples (i.e., they are equivalent in the limit $(n_1+n_2)\rightarrow\infty$), but Hedges' *g* has the best properties for small samples when multiplied by a correction factor that adjusts for small sample bias (Formula 4 below). Hence, we applied Hedges' *g* as the estimator for d_{pop} in our investigation and will not consider Cohen's *d* and Glass' Δ further.

$$correction\ \ factor\ for\ Hedge's\ g = 1 - \frac{3}{4(N-2)-1}, \tag{4}$$

where *N* is the total sample size.

Hedges' *g* assumes homogeneity of variance in the two experimental groups. Kline [24] suggests that if the ratio of the largest standard deviation over the smallest standard deviation is larger than four, the effect sizes should be calculated twice using each standard deviation and the diverging results discussed. Other solutions are to replace s_p with an estimate of the standard deviation of whichever sample is the reasonable baseline comparison group [14], or to use the square root of the mean of s_1, s_2 [5].

Formulas (2) above are applicable for outcomes measured on the continuous scale. When aggregating study results from several studies and the standardized mean difference is to be estimated, there is a need for estimators that approximate a standardized mean difference effect size for variables that are measured on scales other than the continuous. When the

outcome is dichotomous (binary), approximations to the standardized mean difference can be expressed in terms of an arcsine transformation [15] or an odds ratio [24, 37, 38]. When the outcome is ordinal (e.g., small, medium, large) a continuous scale might be assumed and formulas (2) applied, but note that when the number of categories is less than five, this approach will underestimate the population effect size [38]. When nominal outcomes are used, the standardized mean difference must be computed for pairs of categories applying the methods for dichotomous outcomes.

When raw data is unavailable, or means and standard deviations are not reported, effect size estimation can be based on various kinds of statistics. This is relevant for meta-analyses or statistical power analyses, or if a reader wants to judge published results in terms of effect sizes when these are not reported. Table 7 shows the set of formulas for computing Hedges' *g* that we applied in our investigation. Computation of Hedges' *g* in 40 different ways is provided by the ES software tool [37, 38]. Descriptions of computations of standardized mean difference effect size estimates for ANOVA designs are provided in [11].

2.1.2 The r family

The *r family* consists of the Pearson product-moment correlation in any of its combinations of continuous and dichotomous variables [33]. For two treatment conditions and a continuous outcome, the effect size is called the point-biserial correlation, which we will refer to as $r_{pb\text{-}pop}$. When $r_{pb\text{-}pop}$ is squared, it is also called η^2 and it can be interpreted to mean the proportion of variance accounted for by the population means. Hence, we can express the population point-biserial correlation as follows:

$$\textit{Population point-biserial correlation, } r_{pb\text{-}pop} = \sqrt{\frac{\sigma^2_{treatment}}{\sigma^2_{total}}}, \tag{5}$$

where the numerator is the variance of the population means around the grand mean, and the denominator is the variance of all scores around the grand mean. $r_{pb\text{-}pop}$ has the value range [0,1]. An estimator of, $r_{pb\text{-}pop}$, based on information from an ANOVA table, is obtained by taking the square root of the explained variance expressed in terms of the sum of squares of the treatments and the total sum of squares:

$$r_{pb} = \sqrt{\frac{SS_{Treatment}}{SS_{Total}}} \tag{6}$$

Formulas based on *t*-values and other statistics, as well as estimators that adjust for bias, are provided in [24, 28, 31, 32, 35].

The point-biserial correlation is affected by the proportion of subjects in each experimental group. It tends to be highest in a balanced design and approaches zero when the design becomes more unbalanced [24]. As a consequence, r_{pb} values from studies with different splits in the sample size will not be directly comparable. To counteract this, the following corrected r_{pb} is recommended [19]:

$$Corrected\ r_{pb} = \frac{a r_{pb}}{\sqrt{(a^2 - 1) r_{pb}^2 + 1}}, \quad (7)$$

where $a = \sqrt{0.25 / pq}$, and p and q are the proportions of subjects in each experimental group (p+q=1).

Formula (6) above is applicable for outcomes measured on a continuous scale. When both variables are dichotomous, the population point-biserial correlation is called Φ and is expressed in terms of the proportions in a 2*2 table, [14]. When reporting results from a table larger than 2*2, an effect size estimator called Cramer's *V* can be applied [14]. When a categorical outcome is measured on an ordinal scale (e.g., small, medium, large), a continuous scale can be assumed and a point-biserial correlation calculated as for continuous outcome [14]. The population effect size will be underestimated if fewer than five categories are applied [38].

It is possible to compute r_{pb} from Hedges' g, and vice versa. Information might be unavailable for computing one or the other, or one may prefer to view the results in terms of a correlation coefficient when g, say, is reported in an article. The following formula maps g to r_{pb} [5, 35]:

$$r_{pb} = \frac{g}{\sqrt{g^2 + (1/pq) * ((N-2)/N)}}, \quad (8)$$

where N is the total sample size. Note that the formula is simplified by the factor $1/pq=4$ for a balanced design, (p=q=0.5).

2.1.3 Interpretation of standardized effect sizes

It is not intuitively evident how to interpret standardized effect sizes. Some approaches are listed below and described further in this section.

- Standardized effect sizes can be interpreted in terms of the properties of the formula, for example, distributional overlap for the standardized mean difference and explained variance for the point-biserial correlation.

- Standardized effect sizes can be compared with
  - effect sizes reported in similar experiments,
  - effect sizes reported in the research field in question, for example, software engineering as a whole, and
  - standard conventions for small, medium and large effect sizes developed for research in behavioural science.

The population standardized mean difference, d_{pop}, is expressed in terms of mean difference divided by a measure of the variability in the data. We can interpret this formula as the degree of distributional overlap of values for two populations. A large degree of nonoverlap means a large effect size, and when the two distributions are perfectly superimposed, the effect size is zero [5], see Table 1.

Table 1. Distributional nonoverlap percentages for values of d_{pop} [5]

d_{pop}	0.0	0.5	1.0	1.3	2.0	3.0	4.0
Degree of non-overlap	0%	33%	55%	65%	81%	93%	98%

This is further visualized in Figure 2: The unstandardized effect sizes (represented by the differences between the full and dotted vertical line) are equal in (a) and (b). However, the standardized effect size in (a) is larger than the one in (b), because the degree of non-overlap is larger in (a) than in (b). The standardized mean difference reflects what is visualized in the figure: The effect size seems important in (a) but might be hardly noticeable in (b).

A point-biserial correlation can be interpreted in terms of the property of its square root (see Formula 5 and 6); the percentage of total variance that is explained by treatment.

The second possibility of interpretation of a standardized effect size is to take advantage of its standardized property, i.e., that it is comparable across measurement scales. The best interpretation arises from comparison with experiments that test the same hypothesis as the one in question [9]. In the absence of such experiments, an alternative is to compare the

observed effect size to effect sizes reported in the field of interest. We present effect sizes observed in software engineering experiments in Section 4.2.2. A third alternative is to compare the observed effect size against standard conventions that have been developed in

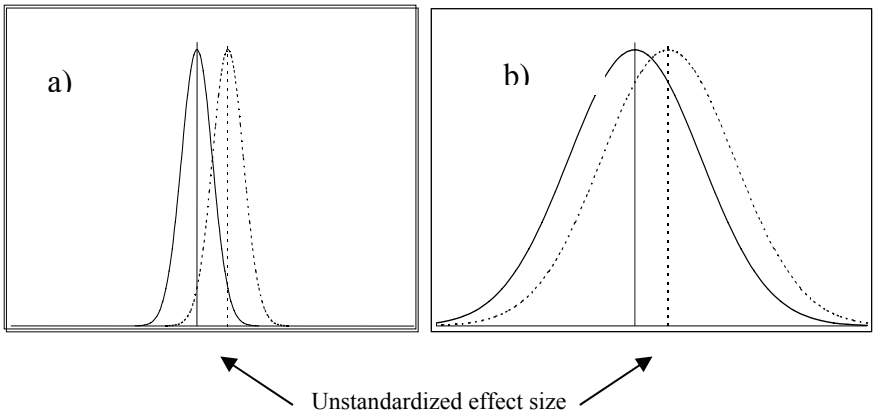

Figure 2. Illustration of how the standardized mean difference effect size can be interpreted in terms of distributional overlap

behavioural science. Values for small, medium and large population standardized effect sizes corresponding to various statistical tests and types of effect size measures are defined by Cohen (1988, 1992). His definitions are based on a combination of a subjective view of average effect sizes observed in behavioural science and a view of what small, medium and large effect sizes should mean. The definitions for d_{pop} and $r_{pb\text{-}pop}$ are shown in Table 2.

Cohen proposed his definitions for statistical power analyses, to help researchers guess on effect sizes when no other sources for effect size estimation existed, i.e., no similar experiments or pilot studies. His definitions are also used to interpret observed effect sizes, but this is also only advisable when no other sources for effect size estimation are available [43]. In later papers, Cohen recommends reporting effect size with a corresponding confidence interval, but does not himself recommend applying the small, medium and large categories in the evaluation of observed effect sizes [6, 8].

Table 2. Values for small, medium, and large d_{pop} and $r_{pb\text{-}pop}$ [5]

Effect size index		Effect size values		
		Small	Medium	Large
d_{pop}	Standardized mean difference	.20	.50	.80
$r_{pb\text{-}pop}$	Point-biserial correlation	.10	.24	.37

The interpretations described above do not include any contextual information. To evaluate whether an observed effect is of practical importance for a specific context, the effect

size must be discussed in relation to each relevant contextual factor, for example, whether the size of efficiency improvement compensates for the effort needed for learning the new method.

2.2 Unstandardized effect size

Unstandardized effect size measures are expressed in terms of raw units of whatever is being measured. This may make the effect sizes easier to interpret, but in contrast to standardized effect sizes, they are not independent of measurement scale. Examples are these: (i) the difference between mean values (e.g., the difference in time taken to perform a given task when using two different methods), (ii) percentage mean difference, and (iii) the difference in proportion of subjects (e.g., the difference between experimental groups with respect to the proportion of subjects viewing a script as correct). The concept of population effect size applies here as well, for example, the effect size measure for population mean difference is expressed as follows:

$$Population\,mean\,difference = \mu_1 - \mu_2, \tag{9}$$

where μ_i is the mean value in population i, which is estimated by the mean $\bar{x}_i$. The standardized counterpart is the standardized mean difference (Formula 1).

Unstandardized effect sizes lend themselves more directly to interpretations of practical importance than do standardized values. For example, an unstandardized effect size of eight hours difference in development effectiveness between two methods used for the same task serves as a better basis for judging the practical importance of the result than a standardized effect size of g=0.5.

2.3 Nonparametric effect size

The standardized effect size measures described in the preceding sections assume parametric models for the outcome variable. Most of the standardized effect size measures developed are parametric. However, assuming parametric models may be inappropriate in many instances, and standardized nonparametric effect size measures based on median values have been suggested in the literature [16, 25, 26]. Computation of these measures requires raw data that is seldom available in articles presenting experimental results. Hence, these nonparametric effect size measures are appropriate for reporting effect sizes, but not always useful in meta-analyses.

Alternatives or supplements to the standardized nonparametric effect size measure are the unstandardized difference in median values or graphical presentations, for example, two box plots within the same figure for easy comparison.

3 Research Method

This section describes how we identified the controlled experiments and primary tests, what kind of information we gathered, and how effect size estimates were computed.

3.1 Identification of controlled experiments and primary tests

We assessed all the 103 papers on controlled experiments (of a total of 5453 papers), identified by Sjøberg *et al*. [39]. Table 3 shows the actual journals and conference proceedings, which were chosen because they were considered to be representative of empirical software engineering research. Furthermore, since controlled experiments are empirical studies that employ inferential statistics, they were considered a relevant sample in this study. The 103 articles reported 113 controlled experiments. The article selection process was determined from predefined criteria as suggested in [22], see [39] for full details.

Since the term "experiment" is used inconsistently in the software engineering community (often being used synonymously with empirical study), we use the term "controlled experiment". A study was defined as a controlled experiment if individuals or teams (the experimental units) conducted one or more software engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages or tools (the treatments). We did not distinguish between randomized experiments and quasi-experiments in this study, because both designs are relevant to software engineering experimentation. In this article, we consistently use the term 'experiment' in the above-mentioned sense of "controlled experiment".

Results from several statistical tests were often reported in the reviewed articles; one article reported 74 tests. We therefore classified each statistical test as either *primary* or *secondary*. The *primary* test what the experiment is designed to evaluate. They were specified in the article by hypotheses or research questions. If no hypothesis or research question was stated, we classified as *primary* those tests that were described to address the main incentive of the investigation. *Secondary tests* comprised all other tests.

Two of the authors of this paper read all the 103 articles and made separate extractions of the primary tests. Then three of the authors reviewed these two data sets to reach a consensus

on which experiments and tests to include. In 14 of the experiments, no statistical testing was performed, and the corresponding articles were thus excluded from the investigation. Seven experiments were excluded because it was impossible to track which result answered which hypothesis or research question. Four experiments were reported in more than one article. In these cases, we included the most recently published. We identified 459 statistical tests corresponding to the main hypotheses or research questions of 92 experiments. Of these tests, we excluded 25 tests of interaction effects, because no well-developed procedures exist for computing effect sizes for interactions [11]. In addition, five tests were excluded because they were regression analyses and involved no treatment. Thus, the final set comprised 429 primary tests, detected in 92 experiments and 78 articles (Figure 3).

Table 3. Distribution of articles describing controlled experiments in the period Jan. 1993 – Dec. 2002

Journal/Conference Proceeding[12]	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 (TOSEM)	1	1.0
Software: Practice and Experience (SP&E)	-	-
IEEE Computer	-	-
TOTAL:	103	100%

[12] The conference *Empirical Assessment & Evaluation in Software Engineering* (EASE) is partially included in that ten selected articles from EASE appear in special issues of JSS, EMSE, and IST.

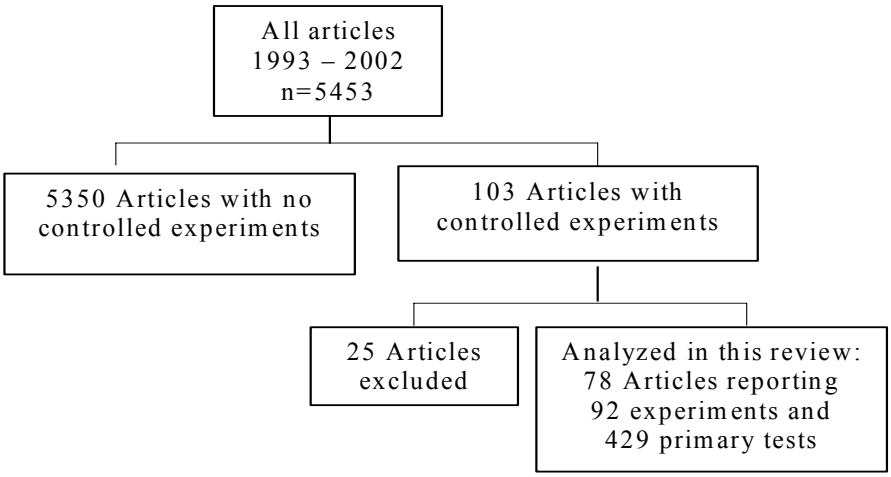

Figure 3. Results of the literature review selection process.

3.2 Information extracted

For each primary test, we recorded

- whether a standardized and/or unstandardized effect size or a graphical visualization of the effect size was reported,
- when an effect size was reported, the interpretation of the effect size and whether practical importance was discussed, and
- sample size, level of significance, *p*-value or information about rejection or acceptation of the null hypothesis, and whether the test was one or two-sided.

In addition, we registered descriptive statistics and estimated the standardized mean difference effect size for those tests with sufficient information reported. Our aim with this computation was to investigate the range of effect sizes in software engineering experiments across experimental topic, treatment and outcome. We therefore estimated the same standardized mean difference population effect size, d_{pop}, for all tests, applying the absolute value for Hedges' *g* as the estimator. Each estimate was corrected for bias by Formula 4 in Section 2.1.1.

The primary tests included parametric tests that compare mean values, nonparametric tests that compare median values or ranks, and tests of the values of dichotomous variables. The applied estimation formulas are listed in Table 7.

We investigated the effect between *two* treatment conditions. Hence, when the primary test was an overall comparison of more than two treatment conditions, we looked at the pair-wise comparisons (contrasts) for our effect size estimation.

We wanted to present the effect sizes as point-biserial correlations as well as standardized mean differences. The *g*-values were transformed into r_{pb}, by applying Formula (8) in Section 2.1.2. Then the values were corrected for unbalanced design by Formula (7). This correction did not change the values to a great extent, since half of the tests had balanced design and the split in sample size was larger than 70-30 for eight tests only (see Section 2.1.2). For those primary tests for which *g* could not be computed, there was not sufficient information to compute r_{pb}, either.

As stated in Section 2.1.1, the pooled standard deviation assumes that the standard deviations are equal in both treatment groups. To check this assumption, we calculated the ratio of standard deviations, when these were reported. The ratio of the largest standard deviation over the smallest standard deviation exceeded four (Section 2.1.1) in seven tests. Consequently, we did not include effect sizes for these tests.

Ten tests were one-sided with results in the direction opposite to the alternative hypothesis. We regarded effect sizes for these tests as real effects and included them in our analysis.

4 Results

The findings comprise two main parts: (1) How effect sizes were reported in the surveyed experiments, with respect to the extent of reporting and interpretation of the reported values and (2) the result of our estimation of standardized effect sizes from information reported in the surveyed experiments.

4.1 The reporting of effect sizes in the surveyed experiments

4.1.1 Extent of effect size reporting

Only 29% of the experiments reported at least one effect size; see Table 4. Two of the 92 experiments reported both standardized and unstandardized effect sizes, eight reported standardized effect sizes only and 17 reported unstandardized effect sizes only. Standardized and unstandardized effect sizes were reported for, respectively, 55 and 46 of the 429 primary tests of the reviewed experiments.

Table 4. Extent of effect size reporting for experiments and primary tests, presented per type of statistical test method

Levels of effect size reporting	Experiments		Primary tests							
			Total		Parametric tests		Non-parametric tests		Tests of dichotomous variables	
	N	%	N	%	n	%	n	%	n	%
Both standardized and unstandardized	2	2.2	3	0.7	3	1.0	0	0	0	0
Standardized (only)	8	8.7	52	12.1	46	15.7	6	6.4	0	0
Unstandardized (only)	17	18.5	43	10.0	32	10.9	6	6.4	5	11.9
No effect size	65	70.7	331	77.2	212	72.4	82	87.2	37	88.1
Total	92	100	429	100	293	100	94	100	42	100

The different types of effect size measures are related to types of outcome and thereby to types of statistical test. Table 4 shows that standardized effect sizes were reported most frequently for parametric tests (46+3 of 293, that is, 17 percent), only a few for nonparametric tests (6 percent) and not for any tests of dichotomous variables. The corresponding parametric tests were ANOVA and *t*-tests; the nonparametric tests were Wilcoxon match pair tests. The standardized mean difference was reported for all but one test, for which the point-biserial correlation coefficient was reported (for an ANOVA test).

Unstandardized effect sizes were reported in equal proportions for parametric tests and tests of dichotomous variables (32+3 of 293 and 5 of 42, respectively, that is, 12 percent) and to a lesser extent for nonparametric tests (6 percent) see Table 4. Most of the 46 unstandardized effect sizes were reported as percentage mean difference (21 tests), but reported were also absolute mean difference (nine tests), difference in proportions or percentage (five tests), ratio of mean values (five tests), difference in average rank values (three tests) and confidence interval for the mean difference (three tests).

For most of the 331 primary tests for which no effect size was reported, mean values, frequencies or graphical presentations of results per experimental group were reported.

We compared the extent of effect size reporting according to whether the results were significant or not (as defined by the authors); see Table 5. For standardized effect sizes there was no difference, but unstandardized effect sizes were reported to a greater extent when significant results occurred than when non-significant results occurred (17.9 percent versus 3.7 percent).

Another factor that seems 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 pair wise comparisons of treatments. Only four of these 51 tests reported the unstandardized effect size.

4.1.2 The interpretation of the effect sizes given by the authors of the reviewed experiments

Possible ways of interpreting the standardized effect size was presented in Section 2.1.3. In one of the surveyed experiment, the point-biserial correlation was interpreted as the percentage of explained variance, but the standardized mean difference effect size was not interpreted in terms of distributional overlap for any of the experiments.

Table 5. Reporting of effect size and significance of results

Levels of effect size reporting	Primary test results				
		Significant		Non-significant	
	N	n	%	N	%
Both standardized and unstandardized	3	3	1.42	0	0
Standardized effect size (only)	52	24	11.3	28	12.9
Unstandardized effect size (only)	43	35	16.5	8	3.7
No effect size	331	150	70.8	181	83.4
Total	429	212	100	217	100

One article reported and compared the standardized effect sizes from three related experiments. For the other experiments, standardized effect sizes were not compared with related research. In two experiments, effect sizes were reported to aid future researchers in planning their experiments, but the sizes were not discussed as part of the result. For the other experiments, standardized mean difference effect sizes were compared with Cohen's conventions from behavioral science [5], for example:

> *We intend to discuss all practically significant results and not constrain ourselves to discussing only statistically significant results. For this exploratory study we consider effects where $\gamma \geq 0.6$ to be of practical significance (the unit is one standard deviation). We make this decision on the basis of effect size indices proposed by Cohen (1969).*

This author judged sizes above 0.6 to be of practical importance. Two authors considered sizes above 0.5 to be of practical importance and one author regarded observed sizes of 0.77

as large. The unstandardized effect sizes were reported with no interpretations or references to practical importance, for example, "Procedural roles reduced the loss of only singular defects by about 30%."

4.2 Our computation of standardized effect sizes from information provided in the surveyed experiments

To identify the sizes of treatment effects found in software engineering experiments, we estimated standardized effect sizes for the primary tests in the reviewed experiments.

4.2.1 Extent of information available for effect size estimation

We managed to estimate standardized mean difference effect sizes for a total of 284 primary tests based on information provided in the reviewed articles. These tests were located in 64 (70%) of the 92 reviewed experiments.

Table 6. Extent of effect size estimation per type of statistical test method

Statistical test method	Total number of primary tests	Primary tests comparing two treatment conditions*			Primary tests comparing more than two treatment conditions*				Total number of effect sizes computed
		N	#ES	%	N	N	%	#ES	#ES
Parametric test of continuous dependent variable	293	250	160	64	43	14	33	55	215
ANOVA	116	78	50	64	38	12	32	40	
t-test	79	79	67	85	0				
Paired *t*-test	39	39	35	90	0				
ANCOVA	28	28	0	0	0				
Tukey's pair wise comparisons	18	18	0	0	0				
Repeated ANOVA	8	5	5	100	3	1	33	6	
Poisson regression	3	3	3	100	0				
Duncan posttest ANOVA	1	0			1	0	0		
Repeated MANOVA	1	0			1	1	100	9	
Nonparametric test of continuous dependent variable	94	90	30	33	4	1	25	3	33
Wilcoxon	41	41	22	54	0				
Mann-Whitney	39	39	2	5	0				
Kruskal-Wallis	8	4	0	0	4	1	25	3	
Rank-sum test	6	6	6	100	0				
Dichotomous dependent variable	42	38	30	79	4	1	25	6	36
Chi-square	25	21	16	76	4	1	25	6	
Fisher's exact test	15	15	12	80	0				
Proportion test	2	2	2	100	0				
Total	429	378	220	58%	51	16	31%	64	284

* **N**: total number of primary tests. **n**: number of primary tests for which effect sizes could be estimated for the pair-wise comparisons, for tests comparing more than two treatments. **#ES**: number of effect sizes estimated

The numbers of effect sizes that were estimated for the various statistical tests are shown in Table 6. Tests comparing two treatment conditions had sufficient information for effect size estimation to be reported for 64% of the parametric tests of continuous variables. The results for nonparametric tests and tests of dichotomous variables were 33% and 79%, respectively. The corresponding results for tests comparing more than two treatment conditions were lower; respectively, 33%, 25% and 25%. Hence, when more than two treatment conditions were compared in a test, information for effect size estimation for the corresponding pair-wise tests was, overall, sparsely reported in the reviewed articles.

Table 7 shows the formulas applied in our effect size estimation. Formula 2 was applied for the majority of tests, including 33 nonparametric tests. We considered mean values to be an appropriate measure of distributional location for nonparametric tests, as long as they were

Table 7. The estimation formulas for Hedges' *g* that were applied in this investigation

No	Data needed and definition of terms	Estimation formulas	References	Number of *g* estimated
1	Hedges' *g*	*g* reported in the paper		18
2	Mean values, standard deviations and group sample sizes	$g = \frac{\overline{X}_1 - \overline{X}_2}{s_p}$	[28]	190
3	Independent t-test value and sample size (n) for each group	$g = t\sqrt{\frac{n_1 + n_2}{n_1 n_2}}$	[28]	16
4	F-ratio from two groups, one way ANOVA	$g = \sqrt{\frac{F(n_1 + n_2)}{n_1 n_2}}$	[28]	13
5	*P*-value and sample size/degrees of freedom	Find t-value based on the p-value and sample sizes, and use Formula 1.	[28]	1
6	Repeated measure design. One between-subject factor and one within-subject factor, t is the number of time points, MS_{bse} is the between-subject mean square error and MS_{wse} is the within-subject mean square error.	Formula (2) in the text using the following estimate for standard deviation $S = \sqrt{\frac{MS_{bse} + (t-1)MS_{wse}}{t}}$	[38], where also estimators for MS_{bse} and MS_{wse} are provided.	4
7	Factorial design.	Formula based on means, sample sizes, standard deviations, corrected for the other factors.	[11, 30]	6
8	Dichotomous outcome, 2*2 table of frequencies.	$g = \frac{\ln(odds\ outcome\ A) - \ln(odds\ outcome\ B)}{\pi/\sqrt{3}}$	[15, 28, 38]	36
Total				284

reported in the paper. In those cases where means and standard deviations were not reported, Formulas 3, 4, 5, 6 and 7, which are based on *t*-value, *F*-values, *p*-value, mean square error and/or sample sizes, respectively, were applied for parametric tests. Formula 8 was applied for tests of dichotomous variables when frequencies and sample sizes were reported.

4.2.2 Standardized effect size values

The values for the 284 estimates of Hedges' *g* range from 0 to 3.40 with a median value of 0.60; see Table 8. The cumulative percentages in the table are, for each *g*, the percentage of effect sizes equal to or below that value. For example, 68% of the effect sizes in our review are equal to or below *g*=1.00. For readers who prefer to view standardized effect sizes in terms of correlations, the r_{pb} values are also presented in Table 8. The range of values is (0, 0.87) with a median value of 0.3 and represents effect sizes that can be expected in studies with balanced design. When the design is unbalanced, the effect sizes tend to decrease with

Table 8. Accumulative percentages for estimated values for Hedges' *g* and the point-biserial correlation

Hedge's *g*	Cumulative percentages for 284 *g* effect size estimates in software engineering experiments	Point-biserial correlation	Cumulative percentages for 284 r_{pb} effect size estimates in software engineering experiments
0.00	7	0.00	7
.10	11	0.10	19
.20	19	0.20	35
.30	28	**0.30**	**50 median**
.40	35	0.40	62
.50	42	0.50	70
.60	**50 median**	0.60	84
.70	56	0.70	92
.80	60	0.80	97
.90	64	0.90	100
1.00	68		
1.10	71		
1.20	73		
1.30	77		
1.40	83		
1.50	86		
1.60	88		
1.70	90		
1.80	90		
1.90	93		
2.00	95		
2.30	97		
2.50	97		
3.00	99		
3.40	100		
Mean *g*	0.81	Mean r_{pb}	0.34
Std *g*	0.69	Std r_{pb}	0.23

increased split in experimental group sizes and the researcher should be aware of this when comparing r_{pb} values from different experiments.

We defined size categories of the estimated *g* and r_{pb} values by viewing the lower 33% of the effect sizes, the middle 34%, and the largest 33%. In Table 9, we present these categories, and we let the median value in these categories represent small, medium and large effect sizes.

Table 9. Small, Medium and Large categories for 284 estimated values for Hedges' *g* and the point-biserial correlation

Size category	Hedges' *g*		Point-biserial correlation, r_{pb}	
	Effect sizes	Median	Effect sizes	Median
Small (lower 33%)	0.00 to 0.376	0.17	0.00 to 0.193	0.09
Medium (middle 34%)	0.378 to 1.000	0.60	0.193 to 0.456	0.30
Large (upper 33%)	1.002 to 3.40	1.40	0.456 to 0.868	0.60

5 Discussion

This section discusses the findings, their implications, and the limitations to this review.

5.1 Comparison with research in behavioural science

It is only in the psychological and educational sciences that we have found similar investigations of effect size reporting, and these assessed only the reporting of *standardized* effect sizes. An assessment of 226 articles on educational and psychology research in 17 journals published in 1994-1995 revealed that standardized effect sizes were reported in 16 articles (7.1%) [20]. Both univariate and multivariate tests, analyzed by several different statistical methods, were included in these 226 articles. This is similar to the proportion of articles reporting standardized effect sizes found in our review (7.7%).

A study by Fidler *et al.* [13] investigated 239 articles published in 1993-2001 that reported new empirical data in the Journal of Consulting and Clinical Psychology. They found that standardized effect size was reported to a greater degree in articles that reported ANOVA tests and Chi-square tests, compared with our review; 32% and 13% compared with 3% and 0, respectively; see Table 10. The extent to which standardized effect sizes were reported in articles that reported *t*-tests was similar in our and Fidler *et al.'s* investigation (15% and 16%, respectively).

Table 10. Number of articles reporting effect size. Comparison of published experiments in software engineering and studies in psychology

Source	Type of statistical test method applied *		
	ANOVA	*t*-test	Chi-square
Articles reporting controlled experiments in software engineering (This review)	3% (1 of 32)	16% (5 of 32)	0% (0 of 9)
Articles reporting psychology studies [13]	32% (38 of 120)	15% (16 of 108)	13% (16 of 126)

*In our review,116 ANOVA tests were reported in 32 articles, 118 *t*-tests were reported in 32 articles and 25 chi-square tests were reported in nine articles.

Considering the maturity of psychological and educational research compared with the relative young field of empirical software engineering, the sparse reporting of effect sizes in our field may be expected. It was more surprising to find similar results to those of Keselman *et al.* and Fidler *et al*. Still, this is a poor consolation, because the extent of effect size reporting in the field of psychological and educational research is regarded as too low, [13, 20].

The sparse reporting of standardized effect sizes in software engineering might be due to effect size estimation's being little known. It is not a topic in standard research methods courses, and formulas for the calculation of effect sizes do not appear in many statistical text books (other than those devoted to meta-analysis). This may improve, as recent literature in empirical software engineering recommends the reporting of effect sizes [12, 23, 29].

However, encouragements for the reporting of effect sizes do not seem to suffice. In the behavioural sciences, it has been suggested that changes in editorial policies will be required before reporting effect sizes will become a matter of routine [13, 44]. Trusty *et al.* [42] report that 23 journals in the social sciences now require that effect sizes be reported, and in their paper, they provide practical information for studies submitted to the Journal of Counseling & Development on generating, reporting and interpreting effect sizes for various types of statistical analysis.

We found one study in the behavioural sciences on the aggregation of standardized effect sizes that was comparable with ours; 1766 effect sizes (standardized mean differences) were estimated from 475 psychotherapy studies [10, 40]. This study found the same distribution of effect sizes as we obtained. Hence, the treatment effects observed in software engineering experiments are of the same magnitude as effects found in a large number of psychotherapy studies; the same average and nearly the same spread of values.

As shown in Table 9, we categorized the effect sizes in our review into the 33% smallest, the 34% middle and 33% largest values and let the median values in these categories represent small, medium and large values in the data. In Table 11, we compare the standardized mean difference effect sizes with corresponding results from an aggregation of average effect sizes from meta-analyses of psychological, educational and behavioural treatments effectiveness [27] (including the study of psychology studies by Smith *et al.*) and the conventions for small, medium and large effect sizes in the behavioural sciences [5].

Table 11. Small, medium and large standardized mean difference effect sizes as observed in this review, in an aggregation of meta-analyses in the social sciences and the conventions in the behavioural sciences

Source	N	Standardized mean difference values		
		Small	Medium	Large
Software engineering experiments (this review)*	284 effect sizes	0.17	0.60	1.40
Meta-analyses of psychological, educational and behavioural studies, [27]†	102 average effect sizes	0.15	0.45	0.90
Conventions from the behavioural sciences, [5]	Not empirically based	0.20	0.50	0.80

* The effect sizes were obtained as the median values for the 33% smallest, the 34% medium and the 33% largest values.

† The effect sizes were obtained as the middle point among the 33% smallest, the 34% medium and the 33% largest values.

The medium and large effect sizes in our review are larger than those observed in the meta-analyses and the conventions from the behavioural sciences. (Note that when we considered the median value as appropriate measure of the middle of the categories, the middle point values were even larger: (small: 0.19, medium: 0.69 and large: 2.2). The discrepancies between the aggregated effect sizes on a study level and the aggregated effect sizes on a meta-analysis level can be explained by the fact that the smallest and largest values on a study level disappear in the overview of average values on the meta-level. The standard conventions in the behavioural sciences seek to represent average values, which seems to be confirmed by the results from the aggregation of meta-analyses. Hence, as our results are the same as those from the aggregation of psychology studies, this might indicate that the conventions from the behavioural sciences (i.e. Cohen's definitions) are appropriate comparators for average effect sizes in software engineering experiments as well (when relevant related research is not present). The effect sizes obtained in our review provide

additional information about the range of values in our field for Hedges' *g* and the point-biserial correlation.

5.2 Guidelines for reporting effect sizes

This section offers guidelines on how to report effect sizes.

5.2.1 Always report effect size

We recommend always reporting effect sizes as part of the experimental results, because there is a risk of making poor inferences when effect sizes are not assessed: (A) nonsignificant results might erroneously be judged to be of no practical importance, and (B) statistical significance might be mistaken for practical importance; see Table 12.

Table 12. Potential problems of inference, when the effect size is not reported,, as a function of statistical significance and effect size [35]

Statistical significance	Effect size	
	Acceptably large	Unacceptably small
p-values low enough	No inferential problem	(B) Mistaking statistical significance for practical importance
p-values too high	(A) Failure to perceive practical importance of "non-significant" results	No inferential problem

The advantage of assessing both effect sizes and statistical significance when making inferences is illustrated by one of the reviewed experiments in which object-oriented design was compared with structured design with respect to the percentage of task-related questions that were answered correctly. The results of statistical tests were nonsignificant at the 0.1 level. The standardized effect size was reported as 0.7, which was regarded as practically important according to Cohen's definitions. The sample size was 13, whereas 56 subjects were needed to achieve a power of 80% at the 0.1 level of significance. If only statistical significance had been reported, the result would have seemed less important than the effect size suggested it to be.

5.2.2 Discuss practical importance

The evaluation of effect sizes based on average values or standard conventions is a first step on the road to assessing the practical importance of the result. For a complete evaluation of practical importance, the effect sizes must be judged in context. Since judging the practical importance of one's experiment is nearly impossible without the relevant situational context

and since the experimental results may be applicable in a wide range of contexts, it may be unrealistic to expect researchers to grade their results in terms of practical importance in their research papers. Nevertheless, we believe that the relevance of software engineering studies would be increased if researchers discussed this issue, possibly through illustrative examples.

Moreover, when an appropriate effect size is reported, the reader can assess practical importance by applying it in their context-specific cost-benefit analysis, as also suggested by [36].

5.2.3 Report both standardized and unstandardized effect size

We recommend reporting both standardized and unstandardized effect sizes, because these two types are supplementary. A standardized effect size includes the variability in the data and gives a complete "average" based on all the data in the sample. There are several approaches to interpreting standardized effect sizes as described in Section 2.1.3. Apply each of them if they bring more information to bear regarding discussion of the result. Moreover, reporting standardized effect sizes aids researchers in planning new experiments (power analysis) and enables comparisons with their own findings.

An unstandardized effect size is easier to interpret than a standardized one and serves as a good basis for discussing practical importance. We place particular emphasis on the value of measures in percentages, which makes the measure applicable to larger-scale projects.

5.2.4 Use the tool box of effect size measures

Many types of standardized effect size measures have been developed, 40 of which are presented in Kirk [21]. However, only two types were reported in the reviewed experiments: the standardized mean difference and the point-biserial correlation. Both of these are parametric. No standardized nonparametric effect size measures were used for the 22% of tests that were analysed by nonparametric methods, neither were any unstandardized effect size measures based on median values used.

When reporting experimental results, we will urge researchers to apply the effect size measure that best suites the data, e.g., nonparametric effect size measures for observations that cannot be assumed to have any known distribution. When aggregating results from different measurement scales, the choices are limited; the standardized mean difference effect size and the point-biserial correlation are most commonly used, because they provide good approximation formulas for variables that are not continuous.

5.2.5 Report confidence intervals

When reporting an effect size, the accuracy of the estimate, measured in terms of a confidence interval, should be reported as well. Although the exact calculation of confidence intervals for a standardized effect size is complicated, good approximations exist for small effect sizes and sample sizes that exceed 10 per group. Descriptions of both exact methods and approximations are found in [14, 17, 24]. Calculating a confidence interval for an unstandardized effect size is simpler and is provided by most statistical reporting tools.

5.2.6 Report descriptive statistics

We recommend always reporting, for each experimental group, results as mean values, standard deviations, frequencies and sample sizes. When performing analysis of variance, report standard ANOVA table results. Such information enables the reader to estimate effect sizes. Even if you report the effect size measure you find most appropriate, the reader might wish to compute a different one, to aggregate results or for purposes of comparison. For factorial design, there might be different views on how to include the effect of different factors; hence, descriptive statistics for subgroups might be useful.

5.3 Implication for power analysis

For statistical power analysis, Dybå *et al*. [12] recommend applying a medium effect size, as defined by Cohen, (for example, g=0.5) when no other information about the population standardized effect size is available. Table 8 can be used as a guide to assess the likelihood of obtaining specific values for Hedges' g and the point-biserial correlation. For example, there is a likelihood of 58% (100% - 42%) that Hedges' g will be larger then 0.5 in software engineering experiments.

If only large effects are interesting to detect, a large effect size is appropriate to apply in the power-analysis. Moreover, if sufficient power is seen as difficult to achieve, we recommend abstaining from hypothesis testing, and recommend instead reporting effect sizes and confidence intervals when investigating hypotheses. Note that confidence intervals contain all the information to be found in significance tests and much more [8].

5.4 Limitations of this study

The main limitations to this investigation are selection bias regarding articles and tests, and possible inaccuracy in data extraction. The limitations regarding selection of articles and tests are described in, respectively, [39] and [12].

The coding of effect size reporting has two limitations: it was performed by one person only, and the quantitative categorization represents a simplification of the complex matter of reporting experimental results. Important nuances might have been lost and some experiments treated “unfairly”. However, the categorization was checked, rechecked and discussed among all authors.

The effect size calculations were also performed by one person only. Moreover, those tests for which an effect size was not calculated, due to lack of sufficient information reported in the article, represent a limitation to the completeness of the presentation of effect sizes. Possible effect size calculation formulas and data that may have been used for effect size calculation might have been overlooked in the reported experiments. Finally, the calculated effects might be biased by any methodological inadequacies of the original studies.

6 Conclusion

This review investigated the extent of effect size reporting in selected journals and conference proceedings in the decade 1993-2002, the interpretation of the effect sizes given by the authors of the reviewed experiments, the extent to which experimental results are reported in such a way that standardized effect sizes can be estimated, and the standardized effect sizes detected in software engineering experiments.

We found that effect sizes were sparsely reported in the reviewed experiments. Only 29% of the 92 experiments reported at least one standardized and/or unstandardized effect size, and only two experiments reported both. The extent to which standardized effect size was reported was equal to or below what is observed in research in psychology.

The standardized effect sizes were compared mainly with the standard conventions for small, medium and large values defined by Jacob Cohen for the behavioural sciences. The practical importance of the effect size in context was not discussed in any of the experiments.

We found sufficient information in the reviewed experiments to compute standardized effect sizes for 25% to 79% of the primary tests, depending on the type of test.

The effect sizes computed in this investigation were similar to what is observed in individual studies in research in psychology. These values are slightly larger than the standard conventions for small, medium and large effect sizes in the behavioural sciences.

Based on our experiences with working with this review, we have three main recommendations to make regarding effect size reporting. (1) Always report effect size in addition to statistical significance, to avoid erroneous inferences. (2) Avoid allowing the

effect size interpretation to become rigorous and a matter of routine. Apply the unstandardized effect size for discussions of practical importance in context. (3) Always report basic descriptive statistics, such as means, standard deviations, frequencies and sample size, for each experimental group. This will enable researchers to estimate their own choice of effect sizes.

Acknowledgements

This research is funded by the Research Council of Norway through the INCO project. The authors are grateful to Lionel Briand, Magne Jørgensen and Amela Karahasanovic for useful discussions and to the anonymous referees for valuable comments. Thanks to Chris Wright for proofreading the paper.

References

[1] 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.

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

[3] 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.

[4] J. Cohen, Some statistical issues in psychological research, in: B.B. Wolman (Ed.), *Handbook of Cinical Psychology*, Academic Press, New York, (1965) 95-121.

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

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

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

[8] J. Cohen, The earth is round ($p < .05$), *American Psychologist* 49 (12) (1994) 997-1003.

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

[10] D.S. Cordray and R.G. Orwin, Improving the quality of evidence: Interconnections among primary evaluations, secondary analysis, and quantitative synthesis, *Evaluation Studies Review Annual* 8 (1983) 91-119.

[11] J.M. Cortina and H. Nouri, *Effect Size for Anova Designs*, Sage, Thousand Oaks, CA, 2000.

[12] T. Dybå, V.B. Kampenes, and D.I.K. Sjøberg, A systematic review of statistical power in software engineering experiments, *Information and Software Technology* 48 (8) (2006) 745-755.

[13] F. Fidler, G. Cumming, N. Thomason, D. Pannuzzo, J. Smith, P. Fyffe, H. Edmonds, C. Harrington, and R. Schmitt, Toward improved statistical reporting in the journal of consulting and clinical psychology, *American Psychological Association* 73 (1) (2005) 136-143.

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

[15] V. Hasselblad and L.V. Hedges, Meta-analysis of screening and diagnostic tests, *Psychological Bulletin* 117 (1995) 167-178.

[16] L.V. Hedges and I. Olkin, Nonparametric estimators of effect size in meta-analysis, *Psychological Bulletin* 96 (3) (1984) 573-580.

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

[18] 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.

[19] J.E. Hunter and F.L. Smith, *Methods for Meta-Analysis*, Sage, Thousand Oaks , CA, 2004. 2nd.

[20] 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.

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

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

[23] 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.

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

[25] H.C. Kraemer and G. Andrews, A nonparametric technique for meta-analysis effect size calculation, *Psychological Bulletin* 91 (2) (1982) 404-412.

[26] J. Krauth, Nonparametric effect size estimation: A comment on Kraemer and Andrews, *Psychological Bulletin* 94 (1) (1983) 190-192.

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

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

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

[30] H. Nouri and R.H. Greenberg, Meta-analytic procedures for estimation of effect sizes in experiments using complex analysis of variance, *Journal of Management* 21 (4) (1995) 801-812.

[31] S. Olejnik and J. Algina, Measures of effect size for comparative studies: application, interpretations, and limitations, *Contemporary Educational Psychology* 25 (2000) 241-286.

[32] 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.

[33] 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.

[34] R. Rosenthal and M.R. DiMatteo, Meta-analysis: Recent development in quantitative methods for literature reviews, *Annual Review of Psychology* 52 (2001) 59-82.

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

[36] 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)

[37] W.R. Shadish, L. Robinson, and C. Lu, *ES: A computer program for effect size calculation*, Assessment System Corporation, St. Paul, USA, 1999.

[38] W.R. Shadish, L. Robinson, and C. Lu, *Manual for ES: A computer program for effect size calculation*, Assessment System Corporation, St. Paul, USA, 1999.

[39] D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A.C. Rekdal, A survey of controlled experiments in software engineering, *IEEE Transactions on Software Engineering* 31 (9) (2005) 733-753.

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

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

[42] 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.

[43] T. Vacha-Haase and B. Thompson, How to estimate and interpret various effect sizes, *Journal of Counseling Psychology* 51 (4) (2004) 473-481.

[44] T. Vacha-Haase, J.E. Nilsson, D.R. Reetz, T.S. Lance, and B. Thompson, Reporting practices and APA editorial policies regarding statistical significance and effect size, *Theory & Psychology* 10 (3) (2000) 413-425.

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

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.

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 113 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.

Keywords: quasi-experiments, randomization, field experiments, empirical software engineering, selection bias, effect size.

1 Introduction

In an experiment, an intervention is introduced deliberately to observe its effects. This is the control that essentially allows the observation of treatment-outcome relations in experiments. Internal validity pertains to the validity of inferring causal relationships from these observations, that is, "whether observed co-variation between *A* (the presumed treatment) and *B* (the presumed outcome) reflects a causal relationship from *A* to *B* as those variables were manipulated or measured" [39]. A challenge in this respect is that changes in *B* may have causes other than the manipulation of *A*. One technique to help avoid such alternative causes is *randomization*, that is, the random assignment of study units (e.g., people) to experimental groups, including *blocked or stratified randomization,* which seeks to balance the experimental groups according to the characteristics of the participants.

However, randomization is not always desirable or possible. For example, in software engineering (SE), the costs of teaching the experimental subjects all the treatment conditions (different technologies) 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.

Laitenberger and Rombach [23] claim that quasi-experiments (in which study units are assigned to experimental groups nonrandomly) represent a promising approach to increasing the amount of empirical studies in the SE industry, and Kitchenham [21] suggests that researchers in SE need to become more familiar with the variety of quasi-experimental designs, because they offer opportunities to improve the rigour of large-scale industrial studies.

Different nonrandom assignment procedures produce different potential alternative causes for observed treatment effects. Hence, in order to support internal validity in quasi-experiments, these potential alternative causes must be identified and ruled out. This is done in the design and analysis of the experiment, for example, by measuring a pretest score and adjusting for initial group differences in the statistical analysis. According to Shadish [37], the theory of quasi-experimentation [4, 5, 8] provides (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 (the validity typology). The theory was developed for research in social science and has also been recognized in other fields of research, such as medical informatics [14], environmental research [24], and economics [29].

Even though the theory of quasi-experiments asserts that quasi-experimentation can yield plausible inferences about causal relationships [37], it seems that in many disciplines there is little awareness of the fact that proper inferences from quasi-experiments require methods different from those used for randomized experiments. Shadish *et al*. [39] claim that the most frequently used quasi-experimental designs typically lead to causal conclusions that are ambiguous, and empirical results from research in medical science and psychology indicate that randomized experiments and quasi-experiments provide different results [6, 7, 15, 38]. The purpose of this article is to report the state of practice in SE on these matters. This is done by a systematic review of the 113 experiments reported in the decade from 1993-2002 in 12 leading journals and conference proceedings in SE [44]. We investigate the extent of quasi-experimentation in SE, the types of quasi-experiments that are performed, how the quasi-experiments are designed and analyzed, how threats to validity are reported, and whether different results are reported for quasi-experiments and randomized experiments.

The remainder of this article is organized as follows. Section 2 presents the concepts used in this investigation. Section 3 describes the research method applied. Section 4 reports the results of this review. Section 5 discusses the findings and limitations of this review. Section 6 concludes.

2 Background

In this article, we use the vocabulary of experiments defined by Shadish *et al.* [39], Table 1. Quasi-experiments are similar to randomized experiments, apart from the fact that they lack a random assignment of study units to experimental groups (randomization[13]). In a between-subject design, there is exactly one experimental group for each treatment condition, and the assignment procedure then assigns each subject to exactly one treatment. In a within-subject design, experimental groups contain multiple treatments, possibly in different orders, and in this case, the assignment procedure assigns each subject to one of these multiple treatment sequences. We use the following operational definition of a controlled experiment defined by Sjøberg *et al*. [44]:

> *A controlled experiment in software engineering is a randomized or quasi-experiment, in which individuals or teams (the study units) conduct one or more software*

[13] The *random assignment* of study units to treatment conditions should not be confused with the *random selection* of study units from the study population to form the study sample, which is also referred to as *random sampling*.

engineering tasks for the sake of comparing different populations, processes, methods, techniques, languages or tools (the treatments).

For simplicity, whenever we use the term "experiment" in the following, we use it in the above-mentioned sense of "controlled experiment." Moreover, the notion *to apply a treatment* will be used, even if the participant's level of SE skill also can constitute a treatment

Table 1. Vocabulary of experiments, from [39]

Experiment: A study in which an intervention is deliberately introduced to observe its effects.
Randomized Experiment: An experiment in which units are assigned to receive the treatment or an alternative condition by a random process, such as the toss of a coin or a table of random numbers.
Quasi-Experiment: An experiment in which units are not assigned to conditions randomly.

2.1 Methods of randomization

Several types of method for random assignment are described in [39]. The two types most relevant for this study are simple random assignment (also called complete randomization) and random assignments from blocks (matches) or strata, which represent a restriction on the randomization.

In simple randomization, the participants are divided into each experimental group by a random procedure, that is; the probability of being assigned to a given group is the same for all the participants. Simple randomization does not guarantee equal experimental groups in a single experiment, but because differences are only created by chance, the various participant characteristics will be divided equally among the treatment conditions in the long run, over several experiments. In order to avoid large differences occurring by chance in a single experiment, blocking or stratifying can be used, in which study units with similar scores on the variables of interest are divided into blocks or strata and then assigned randomly to experimental groups from each block or stratum. When blocking, the participants are divided into pairs when there are two treatment conditions, into groups of three if there are three conditions, etc. When stratifying, the participants are divided into strata that are larger than the number of treatment conditions, for example, the 10 persons with the greatest number of years of programming experience in one stratum, and the 10 persons with the fewest number of years experience in another stratum. The use of blocks and strata in statistical analysis is

described in most statistical textbooks. Determining the optimum number of blocks for a given research setting is discussed in [12] and [31].

Randomization methods span from flipping a coin to using a random number computer generator. The latter procedure is recommended in guidelines for statistical methods in psychology [49], because it enables the supply of a random number seed or a starting number that other researchers can use to check the methods later.

2.2 Selection bias, the problem with quasi-experimentation

Selection bias is a threat to internal validity. It is defined by Shadish *et al.* [39] to be "s*ystematic differences over conditions in respondent characteristics that could also cause the observed effect.*" When a selection is biased, treatment effects are confounded with differences in the study population. Selection bias is presumed to be pervasive in quasi-experiments. Hence, the assignment procedures used in quasi-experiments may lead to pre-experimental differences that in turn may constitute alternative causes for the observed effect. There may also be interactions between selection bias and other threats to internal validity. For example, the participants in one quasi-experimental group might drop out from the experiment (attrition) more often than participants from another experimental group, not because of the treatment, but because they have characteristics that participants in the other group do not have.

Different types of nonrandom assignment procedures might induce different types of causes for selection bias. 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 skills 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 cause other differences of importance for the experimental outcome as well.

When the nonrandom assignment procedure has no known bias, it is called *haphazard assignment*. This might be a good approximation to randomization if, for example, participants are assigned to experimental groups from a sorted list on an alternating basis. However, when haphazard assignment is possible, randomization is often possible as well.

2.3 Design of quasi-experiments

Experimental designs are built from design elements, which can be categorized into four types: assignment methods, measurements, comparison groups, and scheduling of treatments. Corrin *et al.* [9] and Shadish *et al.* [39] show how quasi-experimental designs can be strengthened by adding thoughtfully chosen design elements in order to reduce the number and plausibility of internal validity threats. Among these, we have chosen to describe those that we regard as particularly relevant for reducing selection threats in SE experiments: pretest measures, nonequivalent dependent variables, several experimental groups, and within-subject design (see Table 2).

A *pretest measure* is either taken from a real pretest, i.e., from a task identical to the experimental task, but without any treatment, or it is a measure that is assumed to be correlated with the dependent variable, for example, a similar task (calibration task or training task) [3], exam score, or years of experience. The two latter examples are indicators of the performance of human subjects, which include skill, abilities, knowledge, experience, etc. A challenge is to define which of these characteristics are most relevant in the given experimental setting and to find good operationalizations of those indicators. Pretest scores are used when analyzing the final results to check, or adjust for, pre-experimental differences between the experimental groups. In haphazard assignment, a pretest score can also be used in the assignment procedure (similar to blocked or stratified randomization) to prevent initial differences between the experimental groups.

The *nonequivalent dependent variable* is an additional dependent variable that is expected not to be influenced by the treatments and is used to falsify the hypothesis of alternative explanations for treatment effects or lack of effect. For example, when the outcome is measured in terms of answers to a questionnaire, the nonequivalent dependent variables are questions, the answers to which are assumed not to be influenced by the treatment, but are related to the participants' performance. If the answers from the outcome differ among the experimental groups, whereas the answers from the nonequivalent dependent variables do *not* differ among the groups, the belief that there are no other explanations for the results than the effect of the treatment is strengthened. If both the outcome and the nonequivalent variables differ among the experimental groups, there is an indication that treatment effects might be confounded with group effect. See [42] for an example of use of this kind of nonequivalent dependent variable.

Applying *several experimental groups* allows control of how the quasi-experimental groups influence the results. If the same result is observed for several experimental groups using the same treatment, it confirms the belief that the result is due to treatment and not group characteristics. This is a kind of replication within the single experiment.

Table 2. Techniques for handling threats to selection bias

Techniques	Examples
Pretest scores for controlling for pre-experimental differences between experimental groups	Results from pre-treatment tasks or measures of indicators of subject performance, such as exam scores or years of experience.
A nonequivalent dependent variable for falsifying the hypothesis of alternative explanations for observed effect or lack of effect.	Time used to perform a task if the technology used can be assumed not to influence performance time.
Several experimental groups for some or each treatment condition in order to allow comparison of effect of different types of groups.	Each treatment condition is applied in two companies.
Within-subject design for enabling each subject to be its own control. Note that this design requires control with possible learning effects.	Cross-over design: Two programming languages are compared and half the participants apply first one language and then the other. The order of language is reversed in the other group.

The *within-subject design* is a method for compensating for initial experimental group differences, as each subject or team serves as its own control. The challenge with within-subject designs is that a learning effect might be confounded with a treatment effect. If learning effects cannot be controlled, a within-subject design is inappropriate; see the discussion in [22]. Ways of controlling learning effects are several replications of treatment conditions, as in Design (*a*) in Table 3, or organising the cross-over-design in such a way that it is possible to estimate and compare all learning effects, as in Design (b) in Table 3. An example of a cross-over quasi-experiment is given by Laitenberger and Rombach [23].

Table 3. Two types of strong quasi-experimental designs.

a)	Within-subject design, where the participants are exposed to all treatments several times and in the same order.	G: X_1O X_2O X_1O X_2O, etc.
b)	2*2 cross-over design, where treatments are exposed to participants in opposite order in the two experimental groups.	G1: X_1O X_2O G2: X_2O X_1O

Note: G_i refers to experimental group *i*, *O* refer to an observation/measurement and *X* refers to the use of a treatment.

Strategies for ruling out threats to selection bias are also presented by Reichardt [35]. These strategies mainly involve hypothesis formulations and constructions of comparison groups and are called relabelling, substitution, and elaboration:

- In *relabelling*, the researcher rephrases the research question or hypothesis of the treatment effect to include the joint effect from treatment and the effect of the selection differences among the groups. The relabelling method can always be applied, but is probably the least desirable method to use because the hypothesis of joint effect is often not as interesting to investigate as the treatment effect alone.
- *Substitution* implies that the comparison is substituted by another comparison or by a pair of other comparisons to control for the possible threats. For example, instead of making one comparison in which the selection threat is difficult to rule out, a pair of comparisons is made, in which one is constructed in such a way that the threat is expected to have a positive effect, and the other one in such a way that the threat is expected to have a negative effect. If the results of both comparisons are in the same direction, then the researcher can conclude that the threat has been taken into account.
- *Elaboration* can be described as the "opposite" of *substitution*. The researcher retains the original comparison for which the selection threats are difficult to rule out, but does not replace it with a pair of comparisons as in substitution. Instead, he or she adds other comparisons by, for example, measuring a nonequivalent dependent variable or using several comparison groups, as described in Table 2.

2.4 Analysis of quasi-experiments

Cook and Campbell [8] give the following general advice when analysing quasi-experiments: (1) plan the design carefully, so as to have available as much information that is required for the analysis as possible, (2) use multiple and open-minded analyses, and (3) use an explicit appraisal of the validity of the findings and the plausibility of alternative explanations.

An open-minded analysis means to be prepared to not necessarily use standard procedures for analysis. An example is an investigation of two methods for software cost estimation accuracy [13]. Nineteen projects were used and each project self selected which estimation method to apply. The researchers observed that project characteristics (based on pretests scores) seemed to overrule the effect of the estimation method. Hence, they analysed the projects within blocks of similar projects.

A pretest score may be applied in the analysis of continuous outcomes either (i) in an analysis of pretest-posttest differences (gain-score), (ii) by creating blocks or strata (retrospectively) within each experimental group on the basis of the pretest scores and including the blocking variable in the analysis (ANOVA with blocking or stratifying), or (iii) by applying the pretest as a covariate in the analysis (ANCOVA) [8]. These methods are described and compared in [8]. Among other things, a convincing illustration of how the use of a simple ANOVA yields an incorrect inference compared with using ANCOVA when the experimental groups differ at pretest. An example of the use of ANCOVA is reported in [3]. In that study, a calibration task was used to measure pretest scores (applied as a covariate in an ANCOVA), which affected the overall conclusion. Further improvement to an ANCOVA by making a reliability adjustment is suggested by Trochim [47].

Scepticism regarding the use of traditional statistical methods, such as ANCOVA, to adjust for selection bias is discussed by Lipsey and Cordray [27]. The major problem is the sensitivity of the results to the violation of model assumptions for such methods. Lipsey and Cordray recommend two groups of methods that have evolved over the last decade: The first is a sequential assessment, in which the first step is to analyse whether certain assumptions regarding the application of the treatment have been met. If the assumptions have been met, the outcome is analysed. The second method is growing program evaluation modelling, which focuses on repeated measures of the individual study unit as the base upon which to construct any other analysis of interest.

The use of Bayesian statistics is suggested by Novich [32]. He argues that statistical analyses involve much more than textbook tests of hypotheses and suggests applying Bayesian statistics because this method allows background information to be incorporated into the analysis. However, according to Rubin [36], sensitivity to inference of the assignment mechanism in nonrandomized studies is the dominant issue, and this cannot be avoided simply by changing the modes of inference to Bayesian methods.

3 Research Method

This section describes how the experiments and tests reviewed in this article were identified and how the data was gathered.

3.1 Identification of experiments

The 103 papers on experiments (of a total of 5,453 papers), identified by Sjøberg *et al*. (2005), are assessed in this review. Table 4 shows the actual journals and conference proceedings, which were chosen because they were considered to be representative of empirical SE research. The 103 articles reported 113 experiments. The process for selecting articles was determined from predefined criteria, as suggested in (Kitchenham, 2004); see (Sjøberg *et al.*, 2005) for full details.

3.2 Information extracted

Each of the 113 experiments was categorized as *randomized experiment*, *quasi-experiment* or *unknown* with respect to the assignment procedure. Since one experiment could comprise several tests for which some were exposed to randomization and some were not, we based our categorization on the primary tests when these could be identified. In total, 429 primary tests were identified in 92 experiments in a multi-review process; see [11] for details. We defined the primary tests to be what the experiments were designed to evaluate, as indicated in the descriptions of the hypotheses or research questions. If no hypothesis or research question was stated, we classified as *primary* those tests that were described to address the main incentive of the investigation. *Secondary tests* comprised all other tests.

Table 4. Distribution of articles describing controlled experiments in the period Jan. 1993 – Dec. 2002

Journal/Conference Proceeding[14]	Number	%
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 (TOSEM)	1	1.0
Software: Practice and Experience (SP&E)	-	-
IEEE Computer	-	-
TOTAL:	103	100%

[14] 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.

The assignment procedure was not always described clearly in the articles. An experiment was categorized as randomized if it was stated explicitly that randomization was used for all the primary tests. An experiment was categorized as a quasi-experiment when a nonrandom procedure was reported explicitly for at least one primary test and when the experimental design or the experimental conduct was such that randomization was obviously impossible for at least one primary test. In other cases, the experiment was categorized as unknown. An e-mail request was sent to the authors of the experiments with an unknown assignment procedure. Answers were received for 20 experiments, for which eight apparently employed randomization and are categorized as such in this review.

In 14 of the experiments, no statistical testing was performed. In seven experiments, it was impossible to track which result answered which hypothesis or research question. For these 21 experiments (which are included in the review), no primary tests were identified and hence, the assignment procedure was determined from the description of assignment to the experimental groups. When teams were used as the study unit, we regarded the assignment procedure to be the assignment of teams to experimental groups. We regarded the forming of the teams as being part of the sampling procedure.

In addition to the categorization of each experiment as *randomized experiment*, *quasi-experiment* or *unknown* with respect to the assignment procedure, the following attributes were registered per primary test:

- study unit
- assignment method for randomized experiments and assignment procedure for quasi-experiments

Moreover, the following attributes were registered per experiment:

- whether pretest scores were measured and used in the assignment procedure, descriptive analysis and/or statistical analysis of outcome for at least one primary test
- whether between-subject or within-subject design was used
- whether techniques, other than using a pretest, were used for ruling out threats to selection bias for at least one primary test
- whether the cross-over experiments assessed the results for differences in learning effects for at least one primary test
- whether internal validity was addressed for at least one primary test
- whether threats to selection were reported for at least one primary test

- Whether professionals was used as study unit
- Whether commercial applications were used
- standardized mean difference effect size for each primary test

Regarding the last five bulleted points, data on internal validity, threats to selection, the use of professionals, and the use of commercial applications were gathered by Sjøberg *et al.* [44] and effect size was estimated by Kampenes *et al.* [19]. This data is presented separately for quasi-experiments and randomized experiments in this article.

Although attributes for data collection should ideally be determined prior to a review [20], our experience is that the determination of which attributes to use and their appropriate wording often needs revision during data collection. We therefore conducted a dual-reviewer (VBK, JEH) pilot on approximately 30 % of the articles in order to stabilize (1) the comprehension of description of study unit and experimental design and (2) the categorization of each experiment as *randomized experiment*, *quasi-experiment* or *unknown.*

4 Results

This section presents the extent of randomization observed in the reviewed experiments and how the quasi-experiments were designed and analyzed compared with randomized experiments.

4.1 Extent of quasi-experiments

Of the 113 surveyed SE experiments, 40 (35%) were quasi-experiments (Table 5), although the term "quasi-experiment" was used for only four experiments. There were 66 (61%) randomized experiments. For seven experiments, randomization or nonrandomization was neither explicitly stated nor obvious from the experimental design and clarifications were not obtained from correspondence by email. Examples of phrases from these seven articles are: "subjects were divided into two groups" and "subjects were assigned to groups A and B so that both had subjects of equal ability." For seven experiments, randomization was performed for some of the tests or to some of the experimental groups, but not completely. We categorized these as quasi-experiments. Only three experiments described the randomization method applied: drawing a letter from a hat, drawing a number from a hat, and drawing lots.

4.2 Design of quasi-experiments

We present the design of quasi-experiments in terms of the extent of use of pretest scores, which assignment procedures that were used, the extent of field experiments, and the use of teams as the study unit.

4.2.1 The use of pretest scores

Only 45 % of the quasi-experiments applied a pretest measure (Table 5). This was slightly more than for the randomized experiments. The majority of the pretest measures were applied in the assignment procedure (in 13 of 18 quasi-experiments and in 18 of 26 randomized experiments (blocked or stratified randomization)). The pretest scores were mainly skill indicators, such as exam scores, years of experience, or number of lines of code written. However, for three experiments, a pre-treatment task was performed and a real pretest score measured. Two of these experiments collected data through a questionnaire that was completed by the participants both before and after the treatment was applied. For one experiment, which investigated the effect of using design patterns, SE maintenance tasks were performed both before and after the participants attended a course in design patterns.

Table 5. The extent of randomization and use of pre-test

Type of experiment	Total number of experiments		Use of pretest scores							
			Total		In assignment		In descriptive analysis		In statistical analysis	
	N	%	N	%*	N	%*	N	%*	N	%*
Quasi-experiments	40	35.4	18	45.0	13†	32.5	3	7.5	2	5.0
Randomized experiments	66	58.4	26	39.4	18	27.3	8	12.1	3	4.5
Unknown	7	6.2	3	42.9	3	42.9	0	0	0	0
Total	113	100	47	41.6	34	30.1	11	9.7	5	4.4

* Percentage of the total number of experiments for that particular type of experiment.
† In addition to the twelve experiments using a pretest based assignment, one experiment, categorized as some randomization, used blocked randomization.

4.2.2 Assignment procedures

We found four main types of nonrandom assignment procedures. The characteristics of these types are shown in Table 6.

1 *Assignment to nonequivalent experimental groups.* There were four types of nonequivalent group designs:

a) Five experiments were designed to investigate the effect of indicators of subject performance, such as experience and skill. The experimental groups were formed to be unequal regarding these indicators. The groups were also nonequivalent with respect to other types of experience or skill, due to the nonrandom assignment procedure. Subjects were assigned on the basis of either questionnaire results or the sampling of subjects from different populations.
b) For one of the experiments, subjects were assigned to experimental groups by including subjects with specific knowledge of the technology (treatment) used.
c) Three experiments included subjects from different classes, projects, or universities.
d) Six experiments assigned participants to experimental groups on the basis of their availability.

2 *Haphazard assignment.* Four experiments applied a pretest-based formula or procedure in the assignment, which was not formally random but seemed to be a good approximation; for example, assignment on an alternating basis from a ranked list of examination scores. For eight experiments, a more judgmental approach was used to assign participants to experimental groups, based on pretest scores and previous knowledge about the participants. For eleven of the twelve experiments that used haphazard assignment, the assignment procedure was not described clearly in the article but information was obtained through mail communication.

3 *Some randomization.* For seven of the experiments, randomization was performed for some, but not all, of the experimental groups or the primary tests. Hence, a nonrandom assignment procedure was used as well.

4 *Within-subject experiments in which all participants apply the treatment conditions in the same order.* For six experiments, all the participants were assigned to the same experimental groups, applying both technologies in the same order.

Assignment to *nonequivalent experimental groups*, *haphazard assignment* and *some randomization* were applied for both between-subject designs and cross-over designs for quasi-experiments, see Table 7. All the cross-over designs observed compared two treatments. The randomized experiments with *other within-subject designs* compared more than two treatment conditions and scheduled the treatments in such a way that a true cross-over was not

Table 6. Quasi-experiments detected in this review (number of experiments)

1) Nonequivalent experimental groups (15)

a) Investigation of skill, experience, etc. as treatment (5)

Assignment, for already included participants, based on answers to a questionnaire

- C++ experience
- Database knowledge

Inclusion of subjects from different skill populations

- Students versus professional
- Programming knowledge
- PSP (personal software process) knowledge

b) Assignment based on knowledge of the technology (1)

- Subjects with knowledge of formal methods versus those without such knowledge were used in a comparison of formal methods versus no formal analysis

c) Experimental groups created from similar groups (classes or projects) at different times (3)

- Student classes from two succeeding years were used as experimental groups (2)
- Development courses at a company from two succeeding years were used as experimental groups (1)

d) A natural assemblage of participants into experimental groups (6)

- Two sections of a student class were used as experimental groups (2)
- Availability and schedule played a role in the assignment of subjects to experimental groups (4)

2) Haphazard assignment (12)

a) Formula-based (4)

Assignment method:

- On an alternating basis from a ranked list of previous marks (2)
- An algorithm was used on a ranked list of previous marks (2)

b) Assignment based on the researcher's subjective judgement (8)

The judgement was based on:

- Knowledge of the subjects' skills (1)
- Background information collected from the subjects (2)
- Combination of experiences with the subjects' skills and background information (3)
- Grade point average (2)

3) Some randomization (7)

a) Randomization and nonequivalent group design (4)

- Experimental groups created partly from different physical locations (1)
  In a three-group experiment, one experimental group was selected from one university, while the two others were selected from a different university and assigned randomly to two groups
- Assignment based partly on knowledge of the technology (1)
  In a three-group experiment, one experimental group was formed by subjects who already understood the component before assignment, while other subjects were assigned randomly to the two other groups in a study of reusable components
- Randomization and skill assessment in a factorial design (2)

b) Randomization for individuals, but not for teams, both being study units (1)

c) Randomization for three experimental groups (1). A fourth group was created by using the participants from one of the other groups.

d) Randomization for two experimental groups (1). Some primary tests compared the pre- and post-treatment scores within the groups, i.e. a nonrandomized comparison

4) Within-subject experiments in which all participants applied the treatment conditions in the same order (6)

a) In an inspection experiment, first the usual technique was applied; then the participants underwent training in a new technique followed by applying the new technique in the experiment (3)

b) In an assessment of the effectiveness of inspection team meetings, individual results were compared with team results, individual inspection being performed first by all participants (1)

c) All participants first performed a paper-based inspection, followed by using a web tool (1)

d) All participants applied estimation methods in the same order (1)

Table 7. Experimental designs detected in this review

Experimental design and assignment procedure	Experiments	
	N	%
Quasi-experiments	40	35.4
Between-subject design, nonequivalent experimental groups (10), haphazard assignment (10), and some randomization (2)	22	55.0
Within-subject design	18	45.0
Cross-over design, nonequivalent experimental groups (5), haphazard assignment (2), and some randomization (1)	8	20.0
All participants applied the treatment conditions in the same order	6	15.0
Other design – some randomization	4	10.0
Randomized experiments	66	58.4
Between-subject design	32	48.5
Within-subject design	34	51.5
Cross-over design	19	28.8
Other Within-subject designs	15	22.7
Experiments with unknown assignment procedure	7	6.2
Total	113	100

obtained. Within-subject design is regarded as one way of reducing selection bias when applying a nonrandom assignment procedure. Still, the extent of within-subject designs was smaller for the quasi-experiments than for the randomized ones (45% versus 52%).

4.2.3 Field experiments

The percentage of experiments applying professionals as the study unit was roughly equal for quasi-experiments and randomized experiments (20% versus 18%; see Table 8). Commercial applications were used in 13 % of the experiments, slightly more in randomized experiments. However, the professionals worked with commercial applications in six of the quasi-experiments (13%) and in four of the randomized experiments (6%). Hence, on the basis of

Table 8. Number of randomized and quasi-experiments in the reviewed articles, by type of study unit

Type of experiment	Total	Median sample size‡	Professionals as study unit*		Commercial applications†		Teams as study unit	
			N	%	N	%	N	%
Quasi-experiments	40	42.0	8	20.0	5	12.5	16	40.0
Randomized experiments	66	34.5	12	18.2	10	15.2	11	16.7
Unknown	7	13.5	1	14.3	0	0	2	28.6
Total:	113	36.0	21	18.6	15	13.3	29	25.7

* Students only were used in 82 experiments and a mix of subjects in nine.

† Other types of applications in the experiments were constructed applications (81), student applications (5), unclear (9) and other (3).

‡ Based on the comparison with the largest number of data-points per experiment for the 92 experiments in which this was reported.

type of study unit and application, a greater industrial focus was seen for quasi-experiments than for randomized experiments. In addition, the quasi-experiments had slightly larger sample sizes than the randomized experiments; see Table 8.

4.2.4 Teams

SE tasks are often performed in teams, and the team was the study unit in 26% of the experiments, more often in quasi-experiments (40%) than in randomized experiments (17%); see Table 8.

For eight of the 16 quasi-experiments with teams, the teams were reported as having been formed as follows: by random assignment (4), by random assignment within experimental groups (1), by the participants themselves (2), or on the basis of the researcher's judgment for creating equal teams based on the participants' C++ marks (1). For eight of the 16 cases, the method was not reported. In all eleven randomized experiments with teams, the teams were formed by assigning individuals by a random process.

A pretest score was used for 36 of the 84 (43%) experiments using individuals and for 11 of the 29 (38%) experiments using teams. For all these experiments, the pretest was a measure of the individual skill level, not of the overall team level.

One experiment reported that cost and time were constraints that hindered the use of teams, even if teams would have been a more realistic study unit than individuals for that particular experiment.

4.2.5 Analysis of quasi-experiments

Only two of the 40 quasi-experiments applied a pretest score in the analysis of results in order to adjust for pre-experimental differences in the participants' characteristics and only three compared pretest-scores in a descriptive analysis, Table 5. In the randomized experiments, slightly more (3 and 8) used pretest scores to adjust for pre-experimental differences happening by chance.

The sparse use of pretest scores is one indication that researchers are, in general, unaware of the potential selection bias in quasi-experiments and how the problem can be handled in the analysis of the results. Another indication of this is that internal validity issues were discussed to a lesser extent for quasi-experiments than for randomized experiments (60% versus 70%); see Table 9, i.e., it is addressed less where it is needed more. Moreover, in most

Table 9. Threats to internal validity, as reported in the surveyed experiments

Type of experiment		Internal validity awareness		At least one internal validity threat present		Threats to selection bias present	
	Total	N	%	N	%	N	%
Quasi-experiments	40	24	60.0	10	25.0	3	7.5
Randomized experiments	66	46	69.7	16	24.4	7	10.6
Unknown	7	1	14.3	0	0	0	0
Total	113	71	62.8	26	23.0	10	8.8

Note: the results are presented as number of experiments.

cases when internal validity was addressed, no threat was claimed to be present. The presence of at least one threat was reported to an equal extent for quasi- and randomized experiments. Threats to selection bias were reported for only three of the quasi-experiments. There seems to be some confusion regarding the term *selection bias*, because among the randomized experiments, 11% reported threats to selection bias, probably referring to differences that occurred by chance. In addition, it seems as though some experimenters referred to selection bias when they meant lack of sampling representativeness.

The effect of the assignment procedure is reduced in within-subject designs because the participants apply several treatment conditions. To be able to draw valid inferences, the possible learning effects or carry-over effects must be equal for the different treatment conditions and this must be controlled or tested for in the analysis. This was controlled for in 63% of the quasi-experiments and 32% of the randomized experiments that had a cross-over design, and for 40% of the randomized experiments that had within-subject designs other than cross-over; see Table 10.

Table 10. The extent of analysis of learning effects (cross-over) for within-subject designs

Design		Total	Experiments analyzing learning effects	
			N	%
Cross-over	Quasi-experiments	8	5	62.5
	Randomized experiments	19	6	31.6
Other within-subject designs	Quasi-experiments	6	0	0
	Randomized experiments	15	6	40.0

We attempted to measure whether selection bias influenced the results from the quasi-experiments in this review. There was sufficient information for the effect size to be estimated for 284 primary tests in 64 experiments; see [19] for details. None of these experiments adjusted the results by pretest scores to control for selection bias. Overall, the randomized experiments had higher average and median effect sizes than had the quasi-experiments; see Table 11. However, the result was ambiguous across types of design; the quasi-experimental cross-over designs had effect size values in the same range as the randomized experiments.

Table 11. Experimental results in terms of standardized mean difference effect size

Experimental design		Effect size results from the primary tests				Number of experiments
		Mean	median	std	Number of tests	
Quasi-experiments	between-subject design	0.53	0.39	0.50	31	11
	Cross-over design	0.83	0.81	0.50	19	6
	Same order of treatments	0.51	0.38	0.51	26	6
	Total quasi-experiments	0.61	0.50	0.52	76	23
Randomized experiments	Between-subject	0.83	0.69	0.69	104	24
	Cross-over design	0.99	0.63	0.91	31	12
	Other within-subject designs	0.87	0.77	0.69	61	8
	Total randomized experiments	0.86	0.68	0.73	196	44
Unknown		1.25	1.32	0.85	12	3
	Overall	0.81	0.60	0.69	284	70*

*Some experiments had tests in different categories. A total of 64 unique experiments were represented in this table.

5 Discussion

5.1 Extent of quasi-experimentation

Compared with the extent of quasi-experiments observed in other research areas (range 10%-81%), SE places itself in the middle (39%), see Table 12. Fewer quasi-experiments than randomized ones are conducted in research on medical science and psychology, whereas in experimental criminology, more quasi-experiments than randomized ones are conducted.[15]

[15] For simplicity, we use the terms "quasi-experiments" and "randomized experiments" even if these terms are not always used in other research areas for comparative studies (trials) that use nonrandom and random assignment procedures.

Table 12. Proportion of quasi-experiments

Study	Inclusion criteria	No of exp.	Quasi-experiments	
			N	%
Meta-analysis of psychology studies [40]	• published reports in *Psychological Abstracts* 1975-1979 • at least three comparison groups • between-subject design • information for effect size estimation available	143	-	10
Review of methods in clinical trials [10]	• comparative clinical trials published in one of four medical journals in July – December 1979	67	-	16
Review of controlled clinical trials within surgery [30]	• published controlled clinical trials in six medical journals in 1983 • minimum total sample size:10 (five for cross-over studies)	96	15	16
Review of controlled clinical trials of acute myocardial infarction [6]	• studies published in 1946-1981reporting on a comparison of a treatment to a control	145	43	30
Review of controlled clinical trials within medicine [7]	• published controlled clinical trials in a sample of medical journals in 1980 • minimum total sample size:10 (five for cross-over studies)	114	49	43
Review of experiments in criminology [48]	• all available comparative studies within seven areas of criminal justice	204	158	77
Meta-analysis of experiments within school-based prevention of problem behavioural [51]	• all available reported comparisons from published in journals (80%), other publications (10%) and unpublished reports (10%) • 165 studies included, the results reported on comparison level, not study level	216	174	81
This study	• controlled experiments within SE published in nine journals and three conference proceedings in 1993-2002	113	38	39

Guidelines and textbooks on research in medical science and psychology typically favour randomized experiments for cause-effect investigations, because of their potential to control for bias [2, 18, 34, 49]. This might explain the relatively large extent of randomization in these areas of research. In addition, especially in medical research, randomization is made possible by patients easily enrolling themselves to randomization procedures at hospitals, health care centres and medical doctors.

In contrast, sparse use of randomized experiments is reported in criminology. Many kinds of intervention pertaining to criminal justice do not lend themselves readily to randomized designs [25], because practical, ethical, financial and scientific factors play a role [41]. Hence, it seems that experiments in criminology have mostly been performed in field settings, where randomization is not feasible. In SE, even if 39% of the experiments were quasi-experiments, only 13% (six) of them were field experiments in the sense that the subjects were

professionals working with commercial systems. So, most of the quasi-experimentation in SE consists of research other than field experiments, even though the running of field experiments is regarded as the main incentive for running quasi-experiments in SE [21, 23]. The sparse use of field experiments may be explained by practical constraints, such as costs for the industry, and methodological challenges, such as the level of experimental control that can be achieved in a practical setting [23]. Whereas these constraints seem to lead to a large amount of quasi-experiments being conducted in criminology, the same constraints seem to lead SE researchers to use students as subjects and run randomized experiments rather than quasi-experiments.

In addition to its use in field experiments, we observed the use of quasi-experimental design in the following: investigations of how subject-performance indicators influence the results; comparisons of students from different classes, years, universities, or with treatment-specific knowledge; investigations that make assignments on the basis of the participant's availability; investigations of both teams and individuals for which randomization for both are difficult; within-subject designs for which all participants apply all treatments once and in the same order; and quasi-experiments using haphazard assignment. Except for haphazard assignment, these quasi-experiments represent settings for which randomization is not feasible, but where participants are available and the investigation of cause-effect relationships is possible through a quasi-experimental design. For experiments that use haphazard assignment, blocked or stratified randomization would probably have been possible instead. The use of blocked or stratified randomization for these experiments would have reduced the extent of quasi-experiments from 39% to 23%.

5.2 Results from quasi-experiments compared with randomized experiments

We found that, on average, effect sizes were larger for the randomized experiments than for the quasi-experiments. This might indicate that selection bias in the quasi-experiments influenced the results. There is probably no single explanation for the observed direction of difference. Selection bias in one nonrandomized comparison might be offset by an opposite bias in another such comparison. Hence, it might act more as a random error than a systematic bias that is due to a cause. This will reduce the confidence in the findings, but effect sizes will be consistently neither over- nor underestimated [50]. The small number of quasi-experiments in our review also gives us reason to view with caution the observed differences in effect sizes

from randomized experiments and quasi-experiments. Nevertheless, we should take note of the results, because the hypothesis that selection bias might influence the results from quasi-experiments has a theoretical foundation and is also empirically supported in other research fields. Meta-analyses in psychology, medical research, cognitive behavioural research and criminology found treatment differences partly in favour of randomized experiments [40, 45, 51], partly in favour of quasi-experiments [6, 7, 30, 48], and some found no difference [26, 33][16]. In these investigations, the observed differences were all explained by the potential bias in the quasi-experiments.

The theory of quasi-experimentation suggests how to control for selection bias. Researchers have attempted to assess these suggested precautions empirically. We found that the quasi-experimental 2*2 cross-over design resulted in effect sizes equal to those of randomized experiments. Hence, the cross-over design seems to be effective in avoiding selection bias. However, these results were based on only six experiments. We did not have sufficient data to evaluate any other techniques for handling selection bias. However, researchers in psychology have found that by avoiding self-selection of experimental groups as the assignment method and/or adjusting for pre-experimental differences, selection bias could be eliminated completely [1], or at least to some extent [15, 16, 28, 38] by using a pretest score.

5.3 Indicators of subject performance

Pretest scores are useful for controlling and adjusting for undesirable pre-experimental differences between experimental groups. Among the 49 experiments that measured a pretest score, subject-performance indicators (measured as exam score, years of experience, and number of lines of code written) were used in all but three experiments. This shows that subject-performance indicators are much more commonly used as pretest scores than measures from real pretest tasks.

Moreover, over half of the quasi-experiments did not apply a pretest score to control for selection bias. We believe that even if this is partly due to lack of awareness of its importance, it is also partly due to the fact that a relevant subject-performance indicator score is often difficult to measure. Hence, we conclude that the SE community needs to conduct more research on how to measure different concepts such as skill, ability, knowledge, experience,

[16] The review of 74 meta-analyses in [25] revealed that some of the meta-analyses found treatment differences in favour of randomized experiments, some in favour of quasi-experiments, and some found no difference. Overall, no significant difference was found.

motivation, etc. and how these concepts interact with different types of technologies [43]. In our review, all the investigations of subject-performance indicators were quasi-experimental. We believe that including participants with certain skills in a quasi-experiment is often more relevant than teaching some kind of knowledge as part of a randomized experiment.

5.4 Quality of reporting

There was incomplete reporting of several of the variables that were investigated in this review: type and rationale for assignment procedure, randomization method, threats to internal validity, and information used for effect size estimation. In our experience, this makes it difficult both to understand and evaluate experiments and to conduct systematic reviews and meta-analyses. For 1/4 of the experiments, the assignment procedure was not described in the articles. Only three of the randomized experiments reported the randomization method. Sparse reporting of the method is also found in medical research; in four studies on clinical trials, the randomization method was reported in, respectively, 0.8, 4, 19 and 51% [10, 17, 33, 46].

Even though some of the articles in our review provided excellent descriptions of experimental design issues, in general, justification for the choice of assignment method was lacking. Moreover, internal validity was addressed in only 55% of the experiments and there was sufficient descriptive information for effect size to be estimated for only 64 of the 92 experiments that reported significance testing; see [19] for details.

5.5 Ways to improve quasi-experimental designs in SE

We detected four main types of quasi-experiment. We will here suggest how these experimental designs could be strengthened by using the design elements described in Section 2.

5.5.1 Nonequivalent experimental group design

The main question to ask when the experimental groups are nonequivalent is: which factors could cause these groups to differ before treatment is administered? The answer depends on the assignment procedure. We observed four types of assignment procedures for nonequivalent group designs; see Table 6 (1a-d).

(a) When investigating skill, the experimental groups differ deliberately regarding this skill. In addition, the groups might differ with respect to other relevant types of skills or with respect to other factors that differ between the populations for which the participants are

sampled. The ways of controlling this are to (1) use pretest measures, for example examination score from a common course that concern types of skills other than treatment skill, (2) nonequivalent dependent variables that are assumed not to be influenced by the treatment skill, and (3) several comparison groups that differ with regards to other factors that may influence the results. If possible, we will recommend including participants from different populations because this enables a balanced design. The alternative, which we do not recommend, is to divide already included participants into skill groups on the basis of, for example, a questionnaire.

(b) The same recommendations as above apply for quasi-experiments that include subjects with knowledge of the technology under investigation, i.e., participants with different knowledge in the different experimental groups. The experimental groups might differ with respect to skills other than knowledge of the particular technology. This potential difference must be controlled.

(c-d) When the experimental groups are formed from different student classes, projects or universities, and when participants are included in experimental groups distant in time, or based on availability, the potential factors that could cause the groups to differ are to be found in the characteristics of the groups from which the participants are sampled. Do the students from the different courses have the same curriculum history? Do the project participants have the same amount of experience? What is the reason for their availability at certain time points? Mainly pretest measures and nonequivalent dependent variables are used to control for differences between the experimental groups. However, within-subject design and several comparison groups are also useful if the experimental constraints allow it.

5.5.2 Haphazard assignment

Haphazard assignment might be a good approximation to randomization, especially when the assignment procedure is formula based, which is the case for two of the reviewed experiments. However, little is known about the consequences of haphazard assignment, whereas the statistical consequences of randomization procedures have been well researched [39]. In addition, haphazard assignment that is based on the researcher's subjective judgment, which was seen in four of the experiments, is difficult to report and recheck. The haphazard assignment procedures observed in the reviewed experiments all used a pretest score in the assignment. In general, we recommend using blocked randomization for such experiments.

5.5.3 Some randomization

For seven of the experiments, the design was partly randomized and partly quasi-experimental. Our recommendation for such experiments is to make this mix of design explicit in the article and control threats to selection bias in the quasi-experimental part of the experiment. Ways of controlling threats to selection bias depend on the actual nonrandom assignment method; see Section 5.5.1.

5.5.4 Within-subject design in which all participants apply the treatments in the same order

When the treatments are applied only once, this is a weak quasi-experimental design, because it does not allow proper control of how learning effects may influence the second technology. Still, it was used in six of the reviewed experiments. One explanation given was that the assumed larger learning effect from one of the technologies prevented a cross-over design and that there were too few participants available to achieve sufficient power in a between-subject design. We recommend avoiding such designs and rather using a between-subject design that is analyzed by confidence intervals and effect size measures, thus avoiding the power problem.

5.5.5 Limitations of this review

Limitations regarding the selection of articles and tests are described in, respectively, [44] and [11]. 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.

Another threat to this review is possible inaccuracy in data extraction. The data was extracted by one person (the first author, VBK). However, we conducted a dual-reviewer pilot (VBK, JEH) on approximately 30 % of the articles in order to stabilize such attributes as study unit, experimental design and the categorization of randomized experiment and quasi-experiments, prior to the full review. Moreover, data for the attributes that were perceived to be potential sources of inaccuracy were checked by one of the other authors (JEH). No disagreements were found.

Effect sizes were not calculated for all the tests, due to the lack of sufficient information reported in the articles. In addition, there were few experiments in each quasi-experimental group. These are limitations to the comparison of effect size values between quasi-experiments and randomized experiments. Another limitation to this comparison is that the experiments differ in respects other than the assignment procedure, for example, methodological quality, topic of investigation, and type of outcome measured.

6 Conclusion

The purpose of this systematic review of the literature was to investigate the extent of randomization and quasi-experimentation in SE, how the quasi-experiments were designed and analyzed, how threats to validity were reported, and whether different results were reported for quasi-experiments and randomized experiments.

One third of all the experiments investigated were quasi-experiments. Of these, four main types were observed: (1) Nonequivalent experimental group designs, (2) experiments using haphazard assignments, (3) experiments using some random and some nonrandom methods of assignment, and (4) experiments in which all participants were assigned to the same experimental groups in a within-subject design.

Reports of threats to selection bias were conspicuous by their absence. Pretest scores were measured in nearly half of the quasi-experiments and cross-over designs were used in eight quasi-experiments. Still, for nearly half the quasi-experiments, no effort to handle selection bias was reported. Overall, the randomized experiments had higher average and median effect sizes than had the quasi-experiments. However, the quasi-experimental cross-over designs had effect size values in the same range as the randomized experiments. This result is based on few quasi-experiments, but is in line with quasi-experimental theory and findings in other fields of research: quasi-experiments might lead to results other than those of randomized experiments unless they are well designed and analyzed to control for selection bias.

To conclude, there seems to be little awareness of how to design and analyze quasi-experiments in SE to obtain valid inferences, for example, by carefully controlling for selection bias. Nevertheless, several of the reviewed quasi-experiments were very well performed and reported, and contributed to the recommendations given in this article on how to improve the general conducting of quasi-experiments. We hope that this article will contribute to an increased understanding of when quasi-experiments in SE are useful and an increased awareness of how to design and analyse such experiments.

Acknowledgement

We thank the Research Council of Norway for financing this work through the INCO project, Gunnar Bergersen for useful discussions, and Chris Wright for proofreading.

References

[1] 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.

[2] 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.

[3] E. Arisholm, H. Gallis, T. Dybå, and D.I.K. Sjøberg, Evaluating pair programming with respect to system complexity and programmers expertise, *IEEE Transactions on Software Engineering* 33 (2) (2007) 65-86.

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

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

[6] 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).

[7] 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.

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

[9] W.J. Corrin and T.D. Cook, Design elements of quasi-experiments, *Advances in Educational Productivity* 7 (1998) 35-37.

[10] R. DerSimonian, L.J. Charette, B. McPeek, and f. Mosteller, Reporting on methods in clinical trials, *The New England Journal of Medicine* 306 (22) (1982) 1332-7.

[11] T. Dybå, V.B. Kampenes, and D.I.K. Sjøberg, A systematic review of statistical power in software engineering experiments, *Information and Software Technology* 48 (8) (2006) 745-755.

[12] L.S. Feldt, A comparison of the precision of three experimental designs employing a concomitant variable, *Psychometrika* 23 (1958) 335-353.

[13] S. Grimstad and M. Jørgensen, A framework for the analysis of software cost estimation accuracy, *ISESE 2006*, ACM Press (2006) 58-65.

[14] 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.

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

[16] 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.

[17] M. Hotopf, G. Lewis, and C. Normand, Putting trials on trial - costs and consequences of small trials in depression: A systematic review of methodology, *Journal of Epidemiology and Community Health* 51 (1997) 354-358.

[18] ICH. *Statistical principles for clinical trials*. 1998 [cited 2007 13 June]; Available from: http://www.ich.org/cache/compo/276-254-1.html.

[19] V.B. Kampenes, T. Dybå, J.E. Hannay, and D.I.K. Sjøberg, A systematic review of effect size in software engineering experiments, *Information and Software Technology - in press* (2007).

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

[21] B. Kitchenham, Empirical paradigm - the role of experiments, in: V.R. Basili, et al. (Ed.), *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) 25-32.

[22] B. Kitchenham, J. Fry, and S. Linkman, The case against cross-over designs in software engineering, *Eleventh Annual International Workshop on Software Technology and Engineering Practice (STEP'04)*, (2004).

[23] 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.

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

[25] M.W. Lipsey, Improving the evaluation of anticrime programs: There's work to be done, *Journal of Experimental Criminology* 2 (2006) 517-527.

[26] M.W. Lipsey and D.B. Wilson, The efficacy of psychological, educational and behavioral treatment, *American Psychologist* 48 (12) (1993) 1181-1209.

[27] M.W. Lipsey and D.S. Cordray, Evaluation methods for social intervention, *Annual Review of Psychology* 51 (2000) 345-375.

[28] 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.

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

[30] 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.

[31] J.L. Myers, *Fundamentals of Experimental Design*, Allyn and Bacon, Boston, 1972.

[32] M.R. Novick, Data analysis in the absence of randomization, in: R.F. Boruch, P.M. Wortman, and D.S. Cordray (Ed.), *Reanalyzing program evaluations: Policies and practices for secondary analysis of social and educational programs*, Jossey-Bass, San Francisco, (1981)

[33] K. Ottenbacher, Impact of random assignment on study outcome: An empirical examination, *Controlled Clinical Trials* 13 (1992) 50-61.

[34] S.J. Pocock, *Clinical Trials. A Practical Approach*, John Wiley & Sons Ltd., 1983.

[35] C.S. Reichardt, A typology of strategies for ruling out threats to validity, in: L. Bickman (Ed.), *Research Design, Donald Campbell's Legacy*, Sage Publications, Inc., (2000) 89-115.

[36] D.B. Rubin, Practical Implications of modes of statistical inference for causal effects and the critical role of the assignment mechanism, *Biometrics* 47 (1991) 1213-1234.

[37] 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.

[38] 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.

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

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

[41] J.P. Shepherd, Explaining feast and famine in randomized field trials, *Evaluation Review* 27 (3) (2003) 290-315.

[42] D.I. Simester, J.R. Hauser, B. Wernerfelt, and R.T. Rust, Implementing quality improvement programs designed to enhance customer satisfaction: Quasi-experiments in the United States and Spain, *Journal of Marketing Research* February (2000) 102-112.

[43] 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.

[44] D.I.K. Sjøberg, J.E. Hannay, O. Hansen, V.B. Kampenes, A. Karahasanovic, N.-K. Liborg, and A.C. Rekdal, A survey of controlled experiments in software engineering, *IEEE Transactions on Software Engineering* 31 (9) (2005) 733-753.

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

[46] B. Thornley and C. Adams, Content and quality of 2000 controlled trials in schizophrenia over 50 years, *British Medical Journal* 317 (1998) 1181-1184.

[47] W.M.K. Trochim, *The Research Methods Knowledge Base*, Atomic Dog Publishing, 2001.

[48] 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.

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

[50] 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.

[51] D.B. Wilson, D.C. Gottfredson, and S.S. Najaka, Scool-based prevention of problem behaviors: A meta-analysis, *Journal of Quantitative Criminology* 17 (3) (2001) 247-272.