Replication of Controlled Experiments in Empirical Software Engineering—A Survey

Johan Per Fredrik Almqvist

Examensarbete för 20 p, Institutionen för datavetenskap,
Naturvetenskapliga fakulteten, Lunds universitet

Thesis for a diploma in computer science, 20 credit points,
Department of Computer Science,
Faculty of Science, Lund University

Abstract

Replication of controlled experiments in empirical software engineering—a survey

This survey studies 51 replicated experiments in empirical software engineering in 20 experiment series. These 51 experiments constitute all replicated experiments identifiable by searches of electronic databases of scientific publications.

To study the phenomenon of replication of experiments in software engineering, the report sets out with a detailed review of existing guidelines for experimentation and replication, both within the field of software engineering and from other disciplines.

Thereafter, the experiments and series of experiments are studied from both a quantitative and a qualitative perspective. The survey identifies the journals and conferences where experiment results have been announced, the researchers and institutions most active in replication, as well as the success rates of various types of replication.

The survey also analyses how researchers describe their experience with replication, with regards to access to original material (lab packages), motivations for replication and discussion about experiment subjects (students *vs.* professionals).

Finally, the conclusions of the survey are discussed, and proposals are made with regards to enhancing guidelines for experimentation and replications, as well as proposals for further work.

Sammanfattning

Replikering av kontrollerade experiment inom empiriska studier av programvaruteknik—en undersökning

I denna undersökning studeras 51 replikerade experiment inom empiriska studier av programvaruteknik i 20 serier av experiment. Dessa 51 experiment är samtliga experiment som kunnat identifieras igenom sökningar i elektroniska databaser över vetenskapliga publikationer.

För att studera företeelesen replikering av experiment inom programvaruteknik inleds rapporten med en detaljerad genomgång av befintliga riktlinjer för experimentering och replikering, såväl inom programvaruteknik som från andra vetenskaper.

Därefter studeras experimenten och experimentserierna både från ett kvantitativ och kvalitativt perspektiv. Undersökningen går in på de tidskrifter och konferenser resultaten av experimenten har beskrivits i, vilka forskare och forskningsenheter som är mest aktiva inom replikering, och slutligen hur framgångsrika olika typer av replikering är.

Undersökningen går också in på hur forskarna beskriver sina erfarenheter av replikering med avseende på tillgång på underlag, motiveringar för att genomföra replikeringar och diskussion av subjeken för experimenten (studenter eller yrkesaktiva).

Slutligen diskuteras undersökningens slutsatser, och förslag på möjliga förbättringar av riktlinjer för experimentering och replikering lämnas, tillsammans med förslag på vidare forskning.

Keywords

Empirical software engineering, replicated experiments, survey, replication, replicate, replica, experiment series, external validity, construct validity, guidelines

Contents

List of Tables

List of Definitions

List of Figures

Foreword

I stumbled upon the topic of software engineering and empirical experiments in particular during my stay as an exchange student at the University of Oslo. I took the course in software engineering (INF3120[1]) during the fall semester of 2004. In this course, Hans Gallis asked if any of the students had a good command of the Swedish language, as he needed to run a pilot test of a controlled software engineering experiment that the Simula Research Lab was planning. Like for any students, the perspective of earning a little money while doing something actually related to my studies was appealing [2].

Participating in this pilot test, I got quite interested in the topic of empirical software engineering, and I contacted Simula to ask if they had any openings for master's theses. It turned out they had, and from a broad selection of aspects of empirical software engineering I chose to look at replication.

Some readers may argue that this is not a computer science thesis in the strict sense—the only code written in the production of this thesis are a few lines of `perl` and the LaTeX code of the thesis itself. However, I believe that I couldn't have written this thesis without a sound background in computer science, and I also believe that this thesis will be a valuable contribution to the field. One might also argue that software engineering is a discipline in itself and distinct from computer science.

The writing of this thesis has been a valuable experience for me, and I hope that it will have taught me as much about scientific research methods in general as it has about replicated experiments in empirical software engineering. Such methods, I have come to learn, are invaluable in almost any field, including higher education policy which is another interest of mine [1, 2].

References

[1] Johan Almqvist, Bastian Baumann, Paulo Fontes, Birgit Lao, Stephan Neetens, and Péter Puskás. Bologna student surveys. ESIB—The National Unions of Students in Europe, Brussels, Belgium, September 2003. Available from `http://www.esib.org/frankfurt/survey/ebss.pdf`

[2] Stefan Bienefeld and Johan Almqvist. Student life and the roles of students in Europe. *European Journal of Education*, 39(4):429–441, December 2004.

[1] `http://www.uio.no/studier/emner/matnat/ifi/INF3120/index-eng.html`

Acknowledgements

First and foremost, I would like to thank my girlfriend Lene Henriksen for all her support during the writing of this thesis, and even more so for giving me the impetus to start—and finish!—this thesis.

I am also deeply thankful to Stiftelsen Nils Flensburgs resestipendiefond as well as the NORDPLUS programme of the Nordic council of ministers for the financial support I have received, which eased my stay in Oslo significantly.

I would also like to thank my parents, Paula and Christer Almqvist, and my sister Olga Almqvist for the support they've given me throughout the years.

I'd like to thank my friends in Oslo who have helped throughout the writing of this thesis, not least by asking me what my thesis was going to be about—thus making me think how I could explain this to 'normal people'.

I wish to thank my friends in Lund and Germany—Daniel Busche, Vivekan Pillay, Bertolt Meyer, Daniel Sorge, Eskil Lundgren and Sven Berthold—for reminding me, whenever I needed reminding, not to forget about my studies. My fellow students Daniel Bonnevier, Martin Hedlund and Martin Wahlén were very helpful in giving me informal advice throughout the process of writing this thesis (during the summer and with my supervisors in other countries and/or on holidays).

Moreover, I'd like to thank people at Lund university, the University of Oslo and at Simula Research Lab. In particular, Amela Karahasanović who supported my thesis and helped me back on the right track more than once, and Gunnar J. Carelius and Rolf Vassdokken with whom I worked at Simula. I also wish to thank the library staff at the department of computer science at the University of Oslo, without whom a survey as the one in this thesis wouldn't have been possible.

I further thank the staff at the computer science libraries of the universities of Hannover and Hamburg who helped me retrieve two older and hard-to-find articles, as well as Murray Wood at the University of Strathclyde and Timothy Korson for taking the time to respond to my querying e-mails.

Ståle Waren proof-read a large part of this thesis at short notice. Thanks!

Oslo, spring 2006

Johan Almqvist
`johan@almqvist.net`
`http://www.almqvist.net/johan/`

Chapter 1

Introduction

> The whole trouble comes from the fact that there is so much tinkering with software. It is not made in a clean fabrication process, which it should be. What we need, is software engineering.
>
> Friedrich L. Bauer

This thesis deals with a scientific method—replication—as applied to the empirical research in a scientific discipline: software engineering. Software engineering is an important field of research, as it is concerned with both the quality and cost of software production. Software affects every aspect of our lives, and thus, software engineering research can potentially have great impact on the most diverse aspects of life.

The Institute of Electrical and Electronics Engineers (IEEE) defines software engineering as 'the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; *i.e.*, the application of engineering to software' [r25]. The scientific discipline of software engineering is concerned with the development and evaluation of such approaches.

Empirical software engineering

Empirical studies, *i.e.* studies based on observation and experience, are a fundamental component of software engineering research and practice [r60]: software development practices and technologies must be investigated and backed up with empirical proof in order to be understood, evaluated, and adequately deployed in industry, training and further research. This stems from the observation that higher software quality and productivity are more likely to

be achieved if efficient, well-understood, well-evaluated, tested and proven practices and technologies are introduced in software development [r20, r55].

In software engineering experiments, subjects (*i.e.* programmers) perform a given task, usually in a laboratory environment. As in any experiment, researchers try to manipulate one or more variables while fixing or monitoring every other variable, *i.e.* the environment. Examples of such variables are tasks and assignments, problem-solving methodologies, algorithms, time, or the experience of the subject. This differentiates controlled experiments from other empirical methods, such as surveys and case studies, where no variables are fixed or monitored [r60, r26].

Validity in software engineering experiments

Experimental studies in software engineering have been carried out for several decades, and a number of studies have identified and discussed the strengths and weaknesses of this method of scientific investigation.

The results of an experiment can of course never be better than the design of the experiment. Experiments are conducted to verify hypotheses, and if the hypothesis is not formulated correctly or isn't applicable, the experiment in itself will be flawed.

Brooks *et al.* [r8] note that 'work can often be regarded as research by advocacy, that is the work is managed to support an idea or tool and lacks the rigour [that is] characteristic of empirical studies in more established areas of modern engineering, scientific, and social science disciplines.'

Empirical design, or experiment design, is another issue: is the experiment actually measuring what it is meant to measure? Are all variables that aren't under scrutiny being monitored or maintained at fixed levels?

The question of whether an experimental result, or rather the conclusion from this experimental result, is 'true', is a pivotal issue in research. This concept is usually referred to as validity, *i.e.* are the conclusions from a specific experiment valid across a greater field that that under immediate study in the experiment?

Replication

Replication is often proposed as one of the major avenues to achieve, or ensure, greater validity in software engineering research. However, the act of replicating software engineering experiments brings with it some problems of its own; problems which this thesis will attempt to systematise and address.

Replication, in the context of this thesis, is the repetition of an experiment, either as closely following the original experiment as possible, or with a de-

liberate change to one or several of the original experiment's parameters. In the general literature on scientific methods, the term is sometimes used to describe a particular way of conducting a single experiment [r29]; this is not the definition used in this thesis (*c.f* defintion 2).

The replication of controlled experiments is considered a 'crucial aspect of the scientific method' [r26, r36]. However, replication is not commonplace in software engineering: for example, Sjøberg *et al.* [r55] report that of the 113 experiments studied in their survey, only 18 percent were replications.

In a recent article, Miller [r41] asks whether replicating software engineering experiments is 'a poisoned chalice or the Holy Grail'? He argues that the traditional view of replication should be broadened to make it an effective tool to support software engineering research. Part of the problem is to make external replication more attractive. This is supported by other researchers, who note that 'replicated studies as ours and others are sometimes disregarded' [r49] and 'it has traditionally been difficult to publish such "unsuccessful" studies' [r3].

Brooks *et al.* [r8] produced a set of guidelines for the replication of software engineering experiments in 1996, in what might be described as a case study on replication of a software engineering experiment. Miller [r41], one of the co-authors of [r8], writes that software engineering experimentalists still rely on frameworks from other disciplines in lack of an established empirical framework within the discipline.

Shull *et al.* [r51] discuss how deficiencies in laboratory packages and documentation are one of the weak points within the discipline that makes it difficult to use replication to advance our knowledge. Besides flaws in the design set-up, they attribute the problems of replication to weaknesses in the documentation of experiments. As a solution, they propose better laboratory packages and the use knowledge sharing mechanisms.

1.1 Objective

The objective of this thesis is to study the use of replication of controlled experiments in empirical software engineering. As shown in the previous section, this is a crucial element of research in empirical software engineering, but a brief look at the existing work reveals that the phenomenon of replicating controlled experiments hasn't been widely studied.

This thesis will address both replication from a practical perspective by looking at reports of actual replications, and from a theoretical perspective, looking at theory on the 'art' of replicating experiments as applied to the field of empirical software engineering.

The theoretical foundations of replication in empirical software engineering are described in some depth in the chapter on related work (chapter 2). I have covered the work on replication that I could identify though systematic searches of electronic literature databases as well as the methodological references used by experimenters.

The main part of the survey, chapters 4 to 6, are a detailed study of all reports of replicated software engineering experiments referred in on-line archives. This survey studies these experiments mainly with regards to the operation and implications of replication. In order to draw a complete picture of the state of experiment replication as a scientific method in the field of empirical software engineering, it has been necessary to look at a variety of other aspects as well.

Finally, I will make some suggestions for further work in the field that can contribute to a more thorough understanding of both the practical operation of replication and to the theoretical background.

1.2 Research method

The research method forming the basis for this thesis is a systematic review, which has been defined by Kitchenham [r33] for the discipline of software engineering. These guidelines were not applicable without modification to the survey described in this thesis; some variation became necessary. In order to ensure a well-defined framework for the study, the elements of Kitchenham's guidelines that were applicable were largely maintained.

The main difference, then, is that the survey described here is not based on a distinct research question: the motivation for this survey is a broad interest in the operation of experiment replication within the field of software engineering. The formulation of research questions or hypotheses would only have limited the scope of the survey.

The survey itself involved studying reports of replicated experiments, identified from a number of sources. In total, several hundred articles were subjected to a cursory review, and a total of 51 instances of experiments in 42 reports were studied. Information about each experiment series and each replication was collected on specially prepared forms and in a bibliographical database.

The information drawn from the reports was then synthesised, with a special focus on how the experimenters describe the operation of the replication and the reasons for replicating the experiment.

As the reader will note, the bibliography of this thesis is split into two parts: one part that lists references I have used in the writing of this thesis, and another part that contains the publications I have surveyed. The underlying reason

for this division is that I received a significant ready-made selection of articles describing replicated experiments from the CONTEXT project described below, and the research group that produced this selection doesn't want this list to be published.

1.3 Research context

This master's thesis is related to the Controlled Experiment Survey (CONTEXT) project that is currently run by the Simula Research Lab [r52]. The primary goal of CONTEXT is to survey every report describing controlled experiments in empirical software engineering, published in eleven leading software engineering journals and conferences, between 1993 and 2002. This material is systematically gathered and analysed by a group of researchers and students at Simula Research Laboratory. The results are documented in a database to enable the production of statistical reports as well as the evaluation and comparison of various controlled experiments.

During this survey, which is reported in a recent article by Sjøberg *et al.* [r55], it became apparent that a separate, in-depth investigation of both replications and the phenomenon of replication would be necessary. This thesis is a first step in that direction.

As this thesis looks at replication, I have chosen to enlarge the scope of the published material so that all experiments that are part of a series are included, even when not all of the reports are covered in the eleven journals and conferences mentioned above (*e.g.* Ph. D. theses such as Daly [r15] (and [**? ?**])).

The Simula Research Lab [r53] is a research and research teaching centre for computer science funded by the Norwegian government as well as through research grants. It is named after the Simula programming language, the first object-oriented language developed by Kristen Nygaard and Ole Johan Dahl some 40 years ago. The centre has three departments: Networks and Distributed Systems, Scientific Computing and Software Engineering. This thesis, and the CONTEXT project, are done at the Software Engineering department. The department is part of the International Software Engineering Research Network (ISERN).

1.4 Contributions

Previous work in the field can be roughly divided into three categories. Firstly, there are publications that discuss the scientific validity and maturity of software engineering research, often arguing that replication is an aspect where

much remains to be done not only in quantitative terms, but also in terms of quality. Secondly, there are publications that are concerned with a specific subject within the discipline of software engineering. In such publication, the results of the replication are in the spotlight, not the operation of a replication.

Finally, there is a category of surveys of empirical studies in software engineering; these tend to look at very many different aspects of the empirical method. However, as replication is not commonplace in the discipline, such surveys are not positioned to study this particular subject in great detail.

Publications reporting actual replications tend to include a brief section that combines the three categories above.

The subject of this thesis is at the junction of those three categories. It distinguishes itself from previous work as it is an empirical look at a large set of replications. Previous work on replication as such, encompassing the entire field of empirical software engineering is generally based on a theoretical foundation, as opposed to the empirical base of this thesis. Previous surveys encompassing the entire field of empirical software engineering, on the other hand, have not been concerned with replication their main topic.

1.5 Structure

The remainder of this thesis is organised as follows:

Chapter 2 **Related work**
page 9 Identification and description of existing guidelines for controlled experiments in software engineering and their replication.

Chapter 3 **Method**
page 19 Discussion of the method of identification and review of articles describing replicated software engineering experiments.

Chapter 4 **Tabular overview**
page 25 Organisation of reports into categories and series.

Chapter 5 **Quantitative analysis**
page 33 Review of the properties of replicated software engineering experiments.

Chapter 6 **Qualitative analysis**
page 41 Reflections on replication of experiments as collected from the reports analysed in chapter 4.

Chapter 7 **Validity**
page 49 A discussion of the internal, external and construct validity of the findings in this thesis.

Chapter 8 **Results and further work**
page 51 Conclusions of the review and synthesis of the related work in order to establish guidelines for the replication of controlled experiments in software engineering.

Appendices

page 65 Survey details
page 107 Search results
page 127 Simula Research Lab
page 125 Laboratory packages

Chapter 2

Related work

Previous work related to this thesis can be grouped into two categories. On the one hand, I will look at previous surveys of controlled experiments in software engineering. This includes both surveys covering the entire discipline of software engineering, and surveys concerned with experiments on specific topics (such as object orientation or testing techniques).

This thesis is limited to replicated experiments, so the surveys in this first group can be expected to cover a significantly larger number of experiments than this thesis, because they will cover a large number of non-replicated experiments. However, my review of these surveys will be concerned with their coverage of replication.

On the other hand, I will look at the theoretical foundations of experiment replication. A number of guidelines have been formulated, both on replication in particular and as broader sets of guidelines for empirical investigation in software engineering in general. Many of these refer to, or make use of, guidelines from other scientific disciplines.

2.1 Identification of related work

Related work has been identified by systematical searches of databases. Thanks to a broad approach to the identification of related work, I was able to identify related work in both of the categories above with the same search strategy; a strategy that also had a large degree of overlap between searches for related work (*i.e.* surveys and other texts on replication as a scientific method in experimental software engineering) on the one hand, and reports of replicated experiments as such on the other hand.

I used the electronic library system at Lund University, the Electronic Library Information Navigator (ELIN@) [r39]. This library system consists of over 300 databases, and includes, for example, IEEE on-line, Science Online and

ACM. Furthermore, the system consists of over 11 000 journals and conference proceedings, including Kluwer, IEEE, IEE, Elsevier, Wiley and others. Table 2.1 shows a list of databases that resulted in search results (sorted by order of frequency, before any selection or elimination). All these searches were performed on June 1, 2005.

Num. of hits	Database	URI
85	IEEE	`http://ieeexplore.ieee.org/`
47	Proquest	`http://proquest.umi.com/login`
26	Kluwer	`http://www.springerlink.com/`
4	Elsevier	`http://www.sciencedirect.com/`
3	Wiley	`http://www.interscience.wiley.com/`
1	arXiv	`http://www.arxive.org/`
1	Catchword	`http://www.ingentaconnect.com/`

Table 2.1: Databases actually accessed by ELIN@

The search terms I used were `"software engineering" and replication`. I also searched for the terms `"software engineering" and "external validity"`, as external validity is the problem that replication is meant to address, but this search didn't turn up any relevant results. Looking deeper into the results, I identified another relevant combination of search terms: `"software engineering" and guidelines and experiments`.

The search for `"software engineering" and replication` turned up 91 hits. However, most of them were concerned with the replication of data or processes in distributed systems, or described the replication of a specific experiment without addressing the issue of experiment replication as such. The number of articles retained was five.

The terms `"software engineering" and guidelines and experiments` returned 15 articles. Searching for `"software engineering" and "external validity"` turned up two articles, one concerned with the use of students as subjects and one describing a particular experiment.

I read the abstracts for all articles found and identified those 8 that were concerned with the scientific implications of the replication of experiments. A large number of articles that described replications of experiments were set aside for processing in a later stage of this thesis.

Another major source of inspiration and related work is to be found in the references of the articles surveyed in the following chapters. These reports of actual replications often refer to overviews of previous experiment series in the same or related topics, and they almost always refer to guidelines for

the operation of experiments, and in some cases, specific guidelines for the replication of experiments.

2.2 Surveys of replicated experiments

As outlined above, there are a number of existing surveys on the topic of experimentation in empirical software engineering. One group of these are surveys across the whole discipline, describing the state of the art in the field, and discussing various general issues and observations on experimentation, statistics and reporting. A second group of surveys describe series or families of experiments within a specific software engineering topic, summarising the results of empirical research within that topic.

General surveys on experimentation in software engineering

The base for this thesis, both in terms of the choice of topic and in terms of the first set of materials to be analysed, is the survey by Sjøberg *et al.* [r55]. This survey reviews all articles and conference contributions describing controlled experiments in software engineering from a well-defined set of journals and conferences.

Section 9 of the survey by Sjøberg *et al.* [r55] gives an account of replications of controlled software engineering experiments. They note that only 18 percent of the experiments surveyed are replications. Out of the experiments studied, 'five were close replications' according to the definition of Lindsay and Ehrenberg [r36], and in all these 'the results of the original study were confirmed'. Among the differentiated replications, they note that the success is much larger—seven out of eight—if the replication is performed by the experimenters of the original experiment than if the replication is done by other researchers: 'Six of these [seven] reported results differing from the original experiment and one partly confirmed the results of the original experiment. [r55]'

The Sjøberg survey also points to four other surveys of experiments in empirical software engineering: Tichy *et al.* [r59], Zelkovitz and Wallace [r62], Glass *et al.* [r21] and Zendler [r63].

Zelkovitz and Wallace [r62] use the term 'replicated experiment' as one of twelve categories in a taxonomy of experimental methods used in software engineering. Interestingly, however, none of the other surveys address replication of software engineering experiments.

As far as I have been able to identify, there have not been any surveys focused on the replication of software engineering experiments in general. The publications that do offer advice on how to perform replications are usually

based on the authors' experience or on more general discussions on the low standards in empirical software engineering, such as the article by Kitchenham *et al.* [r34].

Surveys on experimentation in specific software engineering fields

With regards to the group of surveys that focus on experimentation within the confines of a specific software engineering topic, these reports were not found in my initial search. Instead, they were identified by studying the reports of actual replications, or by studying the reference material of the more general studies discussed above.

The Lund and Blekinge group or groups (Höst, Ohlsson, Regnell, Runeson, Thelin, Wesslén, and Wohlin) offer a survey of the exisiting work and replications within the field of requirements inspection in articles [**? ?**] and their book [r60]. Basili *et al.* [r3] also report on this series. None of these studies elaborate on replication as such; a discussion of their findings on the experiment series can be found in section 5.6.

Hansen [r22] and Liborg [r35] have pointed out three other articles describing the state of play in experimenation in limited fields: Jørgensen [r27], Juristo *et al.* [r31] and Deligiannis *et al.* [r18].

Jørgensen [r27] reviews studies of development effort estimation. The publications under review do not overlap with the publications in this survey, nor does Jørgensen's review discuss replication. The review further combines both experiments, case studies and surveys in its analysis of the field.

The study on testing technique experiments by Juristo *et al.* [r31] points out replication as a cornerstone in confirming research result; successful replication is one of the methods the authors suggest to confirm knowledge obtained from such experiments. Juristo *et al.* [r31] further remark that a the large majority of experimental results studied in their review is 'pending laboratory replication'. The authors also note that 'it would be recommendable to unify the techniques under study in future replications to be able to generalise conclusions'.

The review by Deligiannis *et al.* [r18] covers a greater field: experimental investigations into object-oriented technology. The review makes few comments on replication as such, but seems to put forward the notion that too close replications tend to inherit flaws in experimental design, thereby limiting the value of the replication in confirming the findings of the original experiment. I discuss this issue further in subsection 6.2.

Given that these reviews cover some quite specific aspects of the topic they are concerned with, I will come back to their findings when discussing this series of experiments in section 5.6.

2.3 Guidelines for replicability and replication

The second category of publications that must be regarded as related work to this thesis are guidelines for the replication of experiments. The level of detail of these guidelines varies greatly, and in section 6.4 I will get back to the extent to which the practitioners of replications make reference to these guidelines. Some of the guidelines were identified in my literature search but aren't referenced at all in the literature, while other weren't present in my searches but are routinely referred to in reports of replication.

> **Replication** An attempt to reproduce an empirical study in order to further validate its findings, *or* the successful outcome of such an attempt.

Definition 1: Replication

It may be argued that the analysis of the guidelines that are actually referred to by replication practitioners should be dealt with in a chapter after the actual survey has been reported, *i.e.* after chapter 4. However, I believe it is more fruitful to compare the guidelines that are used in practice to those guidelines the literature search turned up in this chapter. In addition, this will give a more complete overview of existing guidelines.

Specific guidelines for empirical software engineering

In one of the first articles discussing replication of experiments in software engineering, published in 1986, Basili *et al.* [r2] don't actually formulate any guidelines on how to replicate, and there is no distinction yet between internal and external replication. However, this 1986 article is important because it brings the concept of replication from the general domain of scientific methodology as discussed by Box *et al.* [r6] and Cochran and Cox [r12] into the software engineering field by suggesting replication not only as a choice in experiment design, but also as a possible 'next step' to be taken after the original experiment is concluded.

In 1994 and 1995, Pfleeger discusses replication in her series of articles on experimental design and analysis. She makes a clear point that replication is a prerequisite (though not always a practical possibility) for a test to qualify as a formal experiment [r44]. Pfleeger then goes on to describe different possibilities of replications, *i.e.* 'variations in experiments, subjects or state variables' and underlines that a replication means 'repeating an experiment under identical conditions' and not 'repeating measurements on the same experimental unit' [r45].

Pfleeger further argues for meticulous documentation of the replication: 'The experimental design must describe in detail the number and kinds of replications of the experiments. It must identify the conditions under which each experiment is run (including the order of experimentation), and the measures to be made for each replicate. [...] That is, we want to be sure that the experimental results clearly follow from the treatments that were applied, rather than from other variables.' [r45]

Brooks *et al.* [r8] produced a set of guidelines for the replication of software engineering experiments in 1996, while at the same time testing these guidelines in practice. Their arguments for replication are equally strong: 'Without the confirming power of external replications, results in experimental software engineering should only be provisionally accepted, if at all. [...] Every stage of an experiment from background reading through to result interpretation through to the writing of the report and its conclusions is prone to error. Because of the problems above, scientists demand that experimental results are externally reproducable *i.e.* that an independent group of other researchers can repeat the experiment and obtain similar results. External replication is an alleged cornerstone of modern scientific disciplines.'

The work of Brooks *et al.* [r8] is clearly labeled as an extension of Basili *et al.* [r2], 'to differentiate between the various kinds of internal and external replication and their powers of confirmation and to allow a better appreciation of the context of a piece of empirical work'.

A number of problems are identified that make (external) replications difficult: 'methods may be poorly reported so that it is impossible to perform an external replication study. For example, instructions and task materials given to subjects may not be fully reported or may otherwise be unobtainable. Other authors have criticised poor reporting.' [r8]

Brooks *et al.* [r8] go on to discuss why the traditional distinction into internal and external replications is not sufficient. They discuss the pros and cons of various forms of replication, and define three axes of distinction: **method**, **task** and **subject**. Each of these elements can be modified by a researcher planning to perform a replication, and the choice should be consciously made with regards to the objective of the replication.

In 1999, Basili *et al.* [r3] expands the theoretical foundation of replication in experimental software engineering to what they call 'families of experiments'. The reason for this is that it is 'hard to know how to abstract knowledge [from software engineering experiments] without a framework [...] that allows results to be combined an generalised [...] by creating a list of the specific hypotheses investigated in an area'.

In order to achieve this, the authors suggest that each hypothesis is defined in a way that allows specific experimentation, including possible variations of such experiments and the effects of such variations. Also, experiments that have been carried out need to be documented in such a way that they can be easily replicated with various degrees of variation of the experimental design. Lastly, they call for a 'a community of researchers that understand experimentation and the need for replication, and are willing to collaborate and replicate'. [r3]

Basili *et al.* [r3] also proposes a categorisation of different types of replication; however, I found this structure to be imprecise and less useful than the scheme proposed by Brooks *et al.* [r8]

In their book, Wohlin *et al.* [r60] describe replication as a central mechanism to achieve validity, albeit a sometimes expensive one. They use the term 'true replication' of successful, close, external replications. When discussing the reasons why replications are unsuccessful, they have a different angle: 'If we do not get the same results, we have been unable to capture all aspects in the experiment design that affects the result'.

In terms of general experiment set-up, the Wohlin *et al.* [r60] book is frequently used by practitioners. The book also recommends that information and materials about any experiment should be made accessible after any experiment, to enable both deeper insight and replication.

Shull *et al.* [r51] discuss how deficiencies in laboratory packages and documentation are one of the weak points within the discipline that makes it difficult to use replication to advance our knowledge. They attribute the problems of replication to flaws in the design set-up and, in particular, documentation of experiments. Laboratory packages and knowledge sharing mechanisms, based on the work of Nonaka and Konno [r43].

Miller, one of the co-authors of [r8], recently wrote [r41] that software engineering experimentalists still rely on frameworks from other disciplines in lack of an established empirical framework within the discipline. This is one of the aspects that this thesis will look into.

Guidelines for replication from other fields

Three other scientific fields seem to be common reference marks when it comes to the replication of experiments in software engineering. Medical science and physics, with their strong and long traditions, form an important background to theoretical discussions of different aspects of replication. Kitchenham *et al.* [r34], for example, provide numerous examples from the medical sciences (*e.g.* surgery, psychiatry and obstetrics) and go on to synthesise and adapt

guidelines from this field in particular to for their 'Preliminary guidelines for empirical research in software engineering'. Some reference is also made to examples from physics, but they are usually very general, such as Brooks *et al.* [r8] referring to the 'kitchen-sink fusion' experiment that has never been replicated or quotes of Rutherford saying 'If you experiment needs statistics, you should have designed a better experiment'.

The practitioners of replication however (as identified in this survey) mainly refer to references from the domain of social science, with books such as as the ones by Cook and Campbell [r14], Campbell and Stanley [r9], Judd *et al.* [r28, r29] and Shadish *et al.* [r50] or articles like the ones written by Rosnow and Rosenthal [r48], Lindsay and Ehrenberg [r36] or Lucas [r38].

The books that were available to me—Judd *et al.* [r29], Shadish *et al.* [r50]—don't cover replication in the sense it is used in this thesis at all. They do have valuable guidelines for the design, set-up and analysis of experiments in general, but their definition of the term replication is different (*c.f.* subsection 1).

> **Statistical replication** Performing the same treatment combination more than once, typically to improve internal validity or to avoid statistical errors.

Definition 2: Statistical replication

The articles, and Lindsay and Ehrenberg's [r36] in particular, are clearer on this point. Lucas [r38] writes that the traditional conception of what a replication is is rather narrow. In fact, any study based on 'the same theoretical propositions with alternative empirical indicators' is a replication, even though it 'may not employ the same methods and settings as previous tests.'

This is a central element of research: 'When a theoretical principle is supported in diverse replications, we gain confidence in the theory, and each successive test increases external validity. [. . .] No study, taken alone, can produce general knowledge.' [r38]

However, Lucas [r38] argues that the main problem lies in the theoretical foundations of many studies. No amount of replication, be it in the wide sense and indeed maybe unintentional, or in the close sense, can compensate for a poor theoretical underpinning of a study.

Lindsay and Ehrenberg [r36], writing on replication in the field of social science and organisational science in particular, start out by establishing that there is a bias against the publication of replicated studies, alongside a bias against the publication of results from unsuccessful studies. Several researchers in the field of empirical software engineering echo this conclusion [**?**], [r3, r41]. Lindsay and Ehrenberg further attribute this effect to the misguided

'focus on single sets of data', notably in the teaching of statistics. Lindsay and Ehrenberg go on to argue that replications 'need not and should not be mere repetitions'. They discuss three main features of replications:

Firstly, at least one replication is needed for results to be of any interest at all. 'An isolated study remains virtually meaningless and useless in itself.' [r36] Any result from such a study is 'a one-off result', and it is not clear whether the conclusions will hold again; 'without knowing this, why should anyone pay attention? [. . .] The first replication is therefore the most dramatic' because 'it shows whether or not a wider [. . .] generalization is possible'. [r36]

Secondly, Lindsay and Ehrenberg argue that 'replication must always involve some variation in the conditions of the study'. In fact, an identical replication is 'impossible' on the one hand and 'would be [. . .] pointless' on the other hand. 'In general, the more explicit, differentiated, and/or deliberate such variations [are] while still obtaining the same result', the more valuable these replications are.

Thirdly, Lindsay and Ehrenberg introduce the concept of close and differentiated replications that is also applied in this thesis, *c.f.* definitions 3 and 4.

> **Close replication** A replication that attempts to keep almost all the known conditions of the study much the same or at least very similar as they were in the original experiment.

Definition 3: Close replication

Lindsay and Ehrenberg note that even close replications do vary in one respect or another, and argue that is important to try and capture these differences. They will be valuable in the analysis both if the result of the replication matches the result of the original experiment, and even more so if the results do not coincide. Such close replications are 'particularly suitable early in a program of research' to see if the results of a study can be repeated at all.

> **Differentiated replication** A replication that involves deliberate, or at least *known*, variations in fairly major aspects of the conditions of the study.

Definition 4: Differentiated replication

As a research programme evolves, the interest of differentiated replications grows. According to Lindsay and Ehrenberg [r36], there are three main reasons to perform differentiated replications. Firstly, by varying the methods, materials and subjects, researchers can be more confident that the results are 'not just an artefact either of the persons conducting the study [. . .] or of the particular

manner in which the original study had been conducted'. Secondly, differentiated replications will widen the scope of the result; and thirdly, 'differentiated replication is a search for *exceptions*,' cases where the results do not hold.

2.4 Conclusion

As this overview of related work has shown, there doesn't seem to be a common ground on guidelines for the replication of experiments in empirical software engineering. As Miller [r41] has remarked, there are a number of valuable suggestions, but section 6.4 of this survey shows that they are not yet commonly used. It would probably be worth while to integrate such guidelines into works that begin to be regarded as standards in the field, such as the books by Juristo and Moreno [r30] or Wohlin *et al.* [r60].

Chapter 3

Method

In this chapter, I will outline the methods I have used in the collection of publications to be surveyed. The chapter will also describe the specific information that was retrieved for each experiment series, experiment and article, as well as detailing how this information was systematised and categorised.

I have taken a start in the procedures proposed by Kitchenham [r33]. These procedures are an adaptation of the well-established review procedures from medical sciences to the software engineering discipline.

However, Kitchenham's procedures are not a perfect fit for this thesis either, because those procedures are designed with another type of review in mind: reviews focusing on a specific software engineering phenomenon. The present survey, on the other hand, is concerned with a specific theory-of-science phenomenon—replication—in the context of software engineering *research* (and not in the context of software engineering *as such*).

The Kitchenham guidelines are still very useful, because many of the specifics and weaknesses of software engineering research that shape those guidelines also influence the present survey.

The procedures dissect the procedure of a systematic review into three stages: planning, conducting and reporting the review.

The procedures make special mention of those parts that the author considers most crucial for inclusion in Ph. D. theses (as opposed to larger, multi-researcher projects) [r33]. I have taken that list as a guideline for my efforts.

- Developing a protocol.
- Defining the research question.
- Specifying what will be done to address the problem of a single researcher applying inclusion/exclusion criteria and undertaking all the data extraction.
- Defining the search strategy.

- Defining the data to be extracted from each primary study including quality data.
- Maintaining lists of included and excluded studies.
- Using the data synthesis guidelines.
- Using the reporting guidelines.

3.1 Planning the survey

The survey was planned on the basis of a general interest into the use of replication with regards to controlled experiments in experimental software engineering. This is the main difference to the reviews Kitchenham [r33] discusses: the present survey has no distinct research question from the field of software engineering. Instead, the question is: How do researchers in the field of empirical software engineering use replication?

The formulation of more distinct research questions or hypotheses in the context of a master's thesis is not practical. A student will, in general, only have a very sketchy understanding of the field under investigation for the thesis before beginning the survey, and hypotheses would either have to be supplied by a supervisor, or the student would be forced to revisit them repeatedly as the survey proceeds—possibly entailing a need to redo large parts of the survey. Therefore, I chose to keep my options open (*c.f.* section 7.3).

The review protocol for this thesis was put together and discussed with my thesis supervisor in Oslo, Amela Karahasanović; my supervisor in Lund, Göran Fries, also had a look at it. It outlined the basic research questions as well as the possible sources for investigation; mainly existing collections of articles and extensive database searches.

In terms of a me being a single researcher undertaking the entire review from searching, inclusion and exclusion and data extraction, I have made use of both secondary literature and the body of publications provided to me at the outset to minimise the risk of flaws ensuing from any biases I may carry. The issue of validity is discussed in greater depth in chapter 7

3.2 Conducting the survey

Kitchenham divides this phase of the review into five steps. The research to be surveyed must be identified, a selection of relevant results must be made from this mass, the quality must be assessed, data must be extracted and the process monitored, and finally, the data must be synthesised.

Journals
ACM Transactions on Software Engineering Methodology (TOSEM)
Empirical Software Engineering (EMSE)
IEEE Computer
IEEE Software
IEEE Transactions on Software Engineering (TSE)
Information and Software Technology (IST)
Journal of Systems and Software (JSS)
Software Maintenance and Evolution (SME)
Software: Practice and Experience (SP&E)
Proceedings
International Conference on Software Engineering (ICSE)
IEEE International Symposium on Empirical Software Eng. (ISESE)
IEEE International Symposium on Software Metrics (METRICS)

Table 3.1: Journals and proceedings in the Sjøberg *et al.* [r55] survey

Identification of survey material

The identification of material to be surveyed began with the results of the Sjøberg *et al.* [r55] survey. That survey reviewed experiments described in 'a sample of nine journals and three conference proceedings in the decade from 1993 to 2002', listed in table 3.1.

The Sjøberg *et al.* survey had identified 22 articles as relevant ([**? ?**]).

I also reviewed the results of my search in chapter 2.1 to identify further articles that describe replicated experiments, finding another 6 articles [**? ? ? ? ?**]. On June 3, 2005, I complemented my search: I searched for `"software engineering" and replicated` which returned 76 matches, one of which [**?**] was relevant and new. Finally, to ensure full coverage, I searched for the terms `"software engineering" and replicates` with a result of two articles that I already knew about from the previous search, `"software engineering" and replicating` for 7 articles I already had found and one that invited other researchers to replicate the experiment reported, and finally `"software engineering" and replica` which didn't return any relevant articles.

The data for all publications was archived in an annotated database, a database that was later re-used to constitute the reference system for this thesis. The raw search results are presented in appendix B for reference.

Selection of publications

The database searches returned a large number of results. One might argue that this indicates that my search strategy or the keywords/search terms I used

Publication title	Year
Empirical Software Engineering	1996
IEEE Trans. Soft. Eng.	1988
Information and Software Technology	1995
Journal of Systems and Software	1995
Software Quality Journal	1997

Table 3.2: Year of first issue available for on-line searching for selected journals

> **Experiment** A trial that is designed in order to verify a hypothesis defined beforehand and carried out in a controlled setting, in which the most critical factors can be controlled or monitored.

Definition 5: Experiment

were not focussed enough. However, I believe that the terms I chose adequately describe the field I am interested in. They also describe other, limited fields of the software engineering discipline, but there were no cases where there was any doubt as to whether a given article discussed the replication of an experiment on the one hand or the replication of data or other IT artefacts on the other hand.

The articles that were harder to judge were those that described replicated empirical studies other than experiments. I made sure to run all of these (six in number) by my supervisor who confirmed my judgement on all of them.

The total number of articles identified as describing replicated (in both senses of the word) experiments is thus 44. These articles describe 51 experiments in 20 series, and 31 replications. Of the 20 series, 14 series were found by the survey performed by Sjøberg *et al.* [r55] and the remaining 6 were identified by the search described above.

Quality Assessment

The procedures set forth by Kitchenham give a detailed account of how the author proposes that quality assessment in reviews should be done. This is certainly one of the largest merits of those procedures, however that part of the procedures isn't quite applicable to the present survey: The objective of this survey is to give an account of as many replications as possible.

The majority of the reports surveyed have been published in academic journals or have been published in the proceedings of research conferences. They have thus been subject to a peer review process. Some of the experiments are part of Ph. D. theses; they have been assessed in a similar manner. The remainder (3 publications) have been published as technical reports from

Authors/reference:	
Type:	
Motivation:	
Guidelines:	
Materials (other tasks):	
Changes (more time? debrief?):	
Encouragement of further replication:	
Dates:	
Subjects (experience, country, university):	
Tasks & materials (measurement changed how?):	
Hypothesis or research questions changed? added?	
Confirms results:	
Brooks *et al.* classification [r8]:	
Other comments and textual quotes:	

Figure 3.1: Data extraction form

renowned institutions, and at least in terms of the completeness of the reporting, are in no way second to the articles and conference papers.

Hence, no paper has been excluded from my review. Any assessment of quality was made in the later stages of the review, adding to the total picture of how replication is practised in empirical software engineering.

Data extraction

The data extraction process was obviously the most time-consuming. The detailed findings of this process can be found in annex A. The data extraction form I used is shown in figure 3.1. It may be important to note that every instance of an experiment represents one data point, and thus one article may contain more than one data point.

Data synthesis

The categorisation, analysis and synthesis of the data is described in detail in the next two chapters. I have striven to use agreed standards of synthesis, such as the [r8] classification, in my initial approach to the data. A tabular representation of key elements of each experiment studied was produced, *c.f.* table 4.1.

Together with the raw data in articles and reports, the data extraction forms and the bibliographic database, this table allowed me to study the data from different angles.

3.3 Reporting the review

Kitchenham's advice on reporting and publication in the procedures is aimed at Ph. D. level students and qualified researchers, so it is only partially applicable to this thesis. This thesis will be reviewed by my thesis advisors and by the examination panel at Lund university. Time will tell if any other form of publication will ensue.

The full documentation of publications reviewed is constrained by the instructions I have received. Apart from this limitation, I have strived to provide as extensive documentation as possible on each step of the survey.

The procedures also lay out a proposal on how the report should be structured in terms of sections and subsections. I have chosen different titles for the sections of this thesis; however, I have ensured that all relevant sections are present.

With regards to writing the thesis, I was also inspired and supported by some other thoughts on writing, such as the advice of Bailey [r1].

3.4 Conclusion

The guidelines proposed by Kitchenham, as well as a number of other manuals or instructions, were immensely valuable for my study. At a first glance, some of them seemed irrelevant, and I didn't bother to read them thoroughly. As my study progressed, I returned to these guidelines more and more often to make sure that the survey process is well documented and traceable.

Chapter 4

Tabular overview

As described above, out of the 41 articles reviewed in this survey, 23 were part of the survey of Sjøberg *et al.* [r55]. Another four ([**? ? ?**]) were identified from the references of those surveyed articles. The remaining articles were identified by independent searches, *c.f.* appendix B.

The first distinction I applied to the collection of experiment reports was made according to a set of categories that appeared to be well established in the field. However, upon closer scrutiny, it became clear that the same words didn't always describe the same concepts. I thus had to decide which categories I was going to apply.

4.1 Grouping and numbering

I started by ordering the experiments into series; *i.e.* original experiments and replications of that experiment. Within each series, which were marked with a letter from A to T, the replications were ordered chronologically.

Series and experiment number

The experiment series are grouped after their topics. The original intent was to group them by their place in the software engineering process as described by Sommerville [r56]. However, a more flexible interpretation of the concept of topic turned out to be more useful, especially with regards to the fact that the existing literature has grouped them in the same manner (see also section 2.2 and [**? ?**] and [r60] on inspection techniques or Deligiannis *et al.* [r18] on object orientation).

Within each group of topics, the series have been arranged in chronological order of oldest-first, and the same ordering is applied to the individual exper-

Series	Year	Topic	Stud.	Prof.	Con.	Rej.	Type	Brooks[a]	Ref.
Inspection techniques									
A 0	1987	Defect detection	■	■	–	–	– –	**(–)**	[**?**]
1	1995		■	□	■	■	diff. ext.	**(i,a,s)**	[**?**]
2	1997		■	□	■	■	diff. ext.	**(i,a,s)**	[**?**]
B 0	1995	Defects-Based Reading	■	□	–	–	– –	**(–)**	[**?**]
1	1997	(requirements inspection)	■	□	□	■	diff. ext.	**(s,a,a)**	[**?**]
2	1998		■	□	■	■	diff. ext.	**(a,i,s)**	[**?**]
3	1998		■	□	□	■	diff. ext.	**(i,i,s)**	[**?**]
4	1998		□	■	■	□	diff. int.	**(s,s,a)**	[**?**]
C 0	1996	Perspective-Based Reading	□	■	–	–	– –	**(–)**	[**?**]
1	1997	(PBR *vs.* Ad-Hoc)	■	□	■	□	diff. ext.	**(i,s,a)**	[**?**]
2	1997	(PBR *vs.* CBR)	■	□	□	■	diff. ext.	**(a,i,a)**	[**?**]
3	1998		■	■	■	□	diff. int.	**(i,i,a)**	[**?**]
4	2000		■	□	□	■	diff. ext.	**(a,s,a)**	[**?**]
D 0	1997	Defect detection	■	□	–	–	– –	**(–)**	[**?**]
1	1997		■	□	■	□	diff. int.	**(i,i,s)**	[**?**]
E 0	2001	Perspective-Based Reading	□	■	–	–	– –	**(–)**	[**?**]
1	2001	(PBR *vs.* CBR)	□	■	■	□	close int.	**(s,s,s)**	[**?**]
2	2001		□	■	■	□	close int.	**(s,s,s)**	[**?**]
F 0	2003	Usage-Based Reading	■	□	–	–	– –	**(–)**	[**?**]
1	2004		■	□	■	□	close int.	**(s,s,a)**	[**?**]
2	2004		■	□	■	□	close ext.	**(i,i,i)**	[**?**]
Object orientation									
G 0	1995	Layering and encapsulation	■	□	–	–	– –	**(–)**	[**?**]
1	1995		■	□	■	□	close int.	**(i,s,s)**	[**?**]
H 0	1996	Maintainability of OO systems	■	□	–	–	– –	**(–)**	[**?**]
1	1996	(inheritance depth)	■	□	■	□	close int.	**(s,a,i)**	[**?**]
I 0	1996	Maintainability of OO systems	■	□	–	–	– –	**(–)**	[**?**]
1	1998	(inheritance depth)	■	□	□	■	close ext.	**(s,s,s)**	[**?**]
2	2000		■	□	□	■	diff. ext.	**(i,a,s)**	[**?**]
J 0	1997	Quality Guidelines	■	□	–	–	– –	**(–)**	[**?**]
1	2001	(maintainability of OO systems)	■	□	■	□	diff. int.	**(s,i,i)**	[**?**]
K 0	1998	Use Case guidelines	■	□	–	–	– –	**(–)**	[**?**]
1	2000		■	□	■	■	diff. ext.	**(i,s,s)**	[**?**]
L 0	1999	Comprehension of OO models	■	□	–	–	– –	**(–)**	[**?**]
1	1999		■	□	■	□	diff. int.	**(s,a,s)**	[**?**]
M 0	2002	Design Patterns	■	□	–	–	– –	**(–)**	[**?**]
1	2002		■	□	■	□	diff. int.	**(s,a,a)**	[**?**]
N 0	2002	UML class diagrams	■	□	–	–	– –	**(–)**	[**?**]
1	2004	(comprehension, modifiability)	■	□	■	□	diff. int.	**(s,i,a)**	[**?**]
O 0	2002	Design Patterns	□	■	–	–	– –	**(–)**	[**?**]
1	2004		□	■	■	□	diff. ext.	**(i,a,i)**	[**?**]
Software maintenance									
P 0	1986	Software maintenance	■	■	–	–	– –	**(–)**	[**?**]
1	1994	(modularity)	■	■	□	■	close ext.	**(s,i,s)**	[**?**]
Q 0	1999	Maintenance Process	■	□	–	–	– –	**(–)**	[**?**]
1	1999	(quick-fix *vs.* iter. enhanc.)	■	□	■	□	diff. int.	**(s,i,s)**	[**?**]
2	1999		□	■	■	□	diff. int.	**(s,i,i)**	[**?**]
Miscellaneous									
R 0	1991	Visual depiction of	□	■	–	–	– –	**(–)**	[**?**]
1	1997	decision statements	□	■	■	□	close ext.	**(s,s,s)**	[**?**]
S 0	2001	Database referential	■	□	–	–	– –	**(–)**	[**?**]
1	2001	integrity metrics	□	■	■	■	diff. int.	**(i,s,i)**	[**?**]
T 0	2001	Process simulation models	■	□	–	–	– –	**(–)**	[**?**]
1	2003	(learning efficiency of)	■	□	■	□	close int.	**(s,s,s)**	[**?**]

Table 4.1: Overview of experiments[b]

[a] (Method, Tasks & Materials, Subjects)

[b] Please refer to the text for notes regarding this table.

iments. This allows the reader to see which series and which experiments can or could have been informed by the results of previous series or experiments.

Year

The year listed in the table is the year of publication of the result. In many cases, due to the nature of scientific publication with its peer review and editorial processes, the actual experiment may have been performed up to several years earlier—there are examples of delays of up to five years. However, the date of the experiment isn't always presented in the report, and secondly, the date that is of interest to this survey is the date when the research results were published and thus became widely known. The empirical software engineering scene is obviously quite small, so a share of the researches in the field will inevitably have had knowledge of such results before the actual date of publication.

4.2 Categorisation

In the literature reviewed, replications are generally categorised along two main lines: **internal–external** and **close–differentiated**. In the related work studied in the previous chapter, some alternative classification schemes have been proposed, and I will discuss below which schemes I have chosen for my survey.

Topic

The attribution of experiments to topics has been taken primarily from the reports themselves, correlating where applicable the categorisations made in the secondary reports described in section 2.2. I have also grouped the topics into larger categories; in particular, inspection techniques and object orientation. To a certain degree, there is a possible overlap in these categories; for example, some of the object-orientation experiments address maintainability issues in object-oriented software engineering methods.

Subjects

In the overview presented in the table on page 26, the experiment subjects have only been roughly divided into either students or professionals. The selection of subjects is reported for almost every experiment, even though the level of detail varies greatly. A number of experiments used both students and professionals, as well as academics, as subjects.

Results

A first overview is given of whether the replication confirms or rejects the findings of the original experiment. In most cases, the reports make clear statements; the exception being experiments that have several hypotheses, some of which are confirmed and some of which aren't.

The results in table 4.1 have been taken from the reports, but have also been correlated with the secondary literature referred in section 2.2 .

Replication type

The literature suggests two broad categories that replications can be divided into. Both of these categories are of the either—or type, so they're very simple and I have therefore chosen to use them.

Close or differentiated replication

The distinction between close and differentiated, which is proposed by Lindsay and Ehrenberg in [r36], is a rather coarse distinction, *c.f.* definitions 3 and 4.

If one distinguishes only close and differentiated replications, a replication that uses professional subjects instead of students would be in the same category as a replication that uses the same subjects as the original, but with tasks in another programming language. However, the distinction does have its merit as a first, rough categorisation; the meaning of 'differentiated' in such a context would be 'not close'.

Internal or external replication

The distinction between internal and external replications is quite clear: External replications are independent replications performed by researchers different from those who performed the original experiment. Internal replications are replications that are not external. However, there are of course cases when this categorisation is not clean-cut: replication T:1, for example, is described by the authors as being 'external', while I have categorised it as internal—the experiment is carried out by the same researchers, who have travelled to another country for the occasion.

I have designated all replications where any person involved in the original experiment is also involved in the replication, or where the replication has been carried out at the same institution as the original experiment, as a close replication. There are of course always borderline cases, but I believe that the strict application of a simple rule is more transparent than guesswork by the

author would be. It should be noted that the interpretation of close replication I have applied seems to be stricter that what Sjøberg *et al.* [r55] have applied.

Brooks' classification

Brooks *et al.* [r8] refer to compromising the integrity of the replication instead of distinguishing between close and differentiated replications. Experimenters have to decide if they wish to improve and change the experiment set-up; on the other hand, 'it may be more important to confirm first the original results'.

In accordance with the classification proposed by Brooks *et al.* [r8], I have therefore categorised the experiments as **similar**, **alternative**, or **improved** in terms of (1) method, (2) tasks and materials and (3) subjects.

Obviously, a replication that is identical in all three categories is a close replication; otherwise, it is a form of differentiated replication. A successful close replication in that sense will fulfil Brooks' criterion [r8] of 'confirm[ing] [. . .] the original results'—given that the experiment design as such is sound. That issue is best addressed by making sure that the replication is external [r7].

A weakness of this classification system is that any classification is relative to the original experiment. When a second or later replication, for example, uses the same materials as a previous replication, this will not be obvious from the classification. For instance, experiments A:1 and A:2 were carried out with the same material. The reader will find more detail in appendix A.

4.3 Other classifications

The literature suggests a number of other classifications or typologies that I have not used. For the sake of completeness, I will discuss them (and my choice) briefly here.

The survey by Sjøberg *et al.* [r55] studies controlled experiments in quite some detail, and discusses some of these elements deeply. For example, there is a detailed account of different types of tasks ('plan–create–modify–analyse') and their frequence in experiments. However, the interest of this perspective with regards to replicated experiments in particular is limited, and I have therefore not done such an analysis.

Experiments and quasi-experiments

Some researchers make a distinction between experiments and quasi-experiments, *i.e.* experiments where the assignment of subjects to treatments is not random. I have chosen not to consider this distinction, because I think that the concept of replication under different condition, *in vivo* and *in vitro*, are more

adequately described by looking at the tasks (real or toy) and participants (professionals or students).

'True' replication

Wohlin *et al.* [r60] write that a close replication that obtains the same results as the original experiment is a 'true' replication. I haven't used that term because it mixes the set-up and the results.

Same or other researchers

In their survey, Sjøberg *et al.* [r55] chose a slightly different approach when distinguishing between experiments that were carried out (and reported) by the 'same' researchers or 'others'. This approach has its advantages, however it does not quite address partial overlaps in the authorship of reports, something that seems to be quite common in this field.

'Real' or 'toy' tasks

While the distinction between 'real' tasks (*i.e.* tasks and materials taken from real-world industry or academic problems) and 'toy' tasks (often designed to pin-point specific issues that the researchers are interested in studying) may be appealing at a first glance, it is difficult to clearly distinguish between the two, and it is even more .

Another metric, the number of lines of code (LOC), is also used in the literature. However, many of the experiments in the present survey are not adequately measurable in lines of code; this metric is relevant primarily in maintenance and code review tasks, but less so in, for example, design tasks.

Baslili's types

Basili *et al.* [r3] distinguish six types of replication: Strict replications, replications that vary the manner in which the experiment is run, replications that vary variables intrinsic to the object of study (*i.e.*, independent variables), replications that vary variables intrinsic to the focus of the evaluation (*i.e.*, dependent variables), replications that vary context variables in the environment in which the solution is evaluated, and replications that extend the theory. I feel that this use of words isn't ideal for the purposes of this study either, because some of the categories Basili *et al.* propose are mutually exclusive while others aren't.

4.4 References

As outlined in the previous chapter, I have been instructed to keep the references for a large part of the articles in the survey confidential. I have added the references in square brackets to table 4.1 so that an inclined reader can find references to these reports elsewhere in the text.

Chapter 5

Quantitative analysis

This chapter analyses and summarises the features of replicated experiments and experiment series that can be compared; for example, the journals or conferences they were reported in, the nature of the replication and the results.

5.1 Overview

In table 4.1 the reader will find an overview of the reports surveyed and their interrelation of the experiments reported therein. This table is an adaptation and extension of the table found in the survey by Sjøberg *et al.* [r55].

5.2 Publication channels

As shown in table 5.1, a significant number of replicated experiments have been reported in *Empirical Software Engineering*, a journal initiated in 1996 by

Total	Orig.	Repl.	Publication title
12	3	10	Empirical Software Engineering
9	8	5	IEEE Trans. Soft. Eng.
3	–	3	Information and Software Technology
2	1	2	Journal of Systems and Software
2	1	1	Proceedings of METRICS
2	2	–	Proc. of the Worksh. on Emp. St. of Programmers
1	1	–	Software Quality Journal
1	1	–	Lecture Notes in Computer Science
2	–	2	Ph. D. theses
1	–	1	Technical reports
7	6	3	other Conference proceedings

Table 5.1: Publication channels (combined)

Year	Number
1994	1
1995	2
1996	1
1997	6
1998	5
1999	2
2000	3
2001	3
2002	1
2003	1
2004	4

Table 5.2: Reports of replications per year

Year	Confirm	Reject
1994	–	1
1995	1	–
1996	1	–
1997	3	2
1998	4	2
1999	3	–
2000	–	1
2001	3	1
2002	1	–
2003	1	–
2004	4	–

Table 5.3: Replications confirming and rejecting original findings, by year

some of the most prominent researchers in the field [r20]. *IEEE Transactions on Software Engineering* is the second most common journal for such reports. With three reported replications, *Information and Software Technology* is in third place for the reporting of replications.

The other conferences and books containing such reports present no clear trend; it is however worth noting that there are two Ph. D. theses based on reports of replicated experiments.

In terms of where the original experiments were published, the first place is occupied by *IEEE Transactions on Software Engineering*, with *Empirical Software Engineering* ranking second. Two experiments were first presented at the Workshop on Empirical Studies of Programmers. Interestingly, no original experiments that were later replicated were found in the *Information and Software Technology* journal.

These figures must of course be read with some reservation, because many of the experiment have been reported in several ways (journals, conferences, technical reports and Ph. D. theses). In terms of comparing the frequency of different journals to each other (or different conferences to each other), they are however quite useful.

5.3 Time of publication

Looking at table 5.2, it is interesting to note that the majority of publications during the years with the highest activity (1997 and 1998) were made in Empirical Software Engineering [r20], a journal that was first published in March 1997.

As several researchers have noted, both for the field of empirical software engineering and for other scientific fields, there may be a bias against the publication of unsuccessful replications [**?**], [r3, r41, r36]. The nature of the data in this survey does not allow any conclusions on this topic, however it is interesting to note that the number of reports of 'unsuccessful' replications reached a high level in 1997 and 1998. The *Empirical Software Engineering* journal was started in this very period, with an explicit policy to publish such 'unsuccessful' reports Basili *et al.* [r3], Engineering [r20]. The number of such reports decreased after this period, and no such reports were published in the last three years covered by the survey.

5.4 Researchers

The most prolific researchers in replication, each having participated in three replications, are John W. Daly and James Miller (both with the University of Strathclyde) as well as Oliver Laitenberger (with Fraunhofer IESE, Kaiserslautern, Germany). All of these are also reported by Sjøberg *et al.* [r55] as being among the top 20 researchers conducting controlled experiments. [r55] use a weighted scale based on the ordering of authors in reports; I have only computed the number of reports each researchers has authored.

Another set of eight researchers have reported two replications: Andrew Brooks, Marc Roper and Murray Wood (all three with the University of Strathclyde), Thomas Thelin and Per Runeson (both with Lund institute of technology), Mario Piattini and Marcela Genero (University of Castilla—La Mancha) as well as Guiseppe Visaggio (University of Bari).

In terms of organisations, the Fraunhofer IESE laboratory in Kaiserslautern, Germany, is clearly the most prolific actor in replication, having performed 5 replications and hosted another one. Most replications were performed using students from the University of Kaiserslautern as subjects. This laboratory has hosted a number of researchers from other research groups in empirical software engineering over the years.

The University of Strathclyde and the University of Maryland have performed 4 replications each, while three have been performed at the University of Bari. All of these, though in a different order, are also represented in the top 10 organisations performing controlled experiments, as listed by Sjøberg *et al.* [r55].

Several other institutions have performed two replications, for example the University of Bournemouth, where these replications were performed by apparently unrelated researchers (replications I:1 and R:1).

5.5 Topics

The grouping of experiment series I have undertaken shows that two main areas of software engineering attract a lot of experimentation: inspection techniques and object orientation. Sjøberg *et al.* [r55] use a more fine-grained distinction, based on a selection of classification schemes. In general, the observation that 'code inspections and walkthroughs and object-oriented design methods' are 'two prominent technical areas' [r55] for experimentation, are true not only for experimentation in general, but also for replication in particular. It would appear that experiments on object-oriented design methods are even more strongly represented among the replications.

5.6 Particular features of selected experiment series

The details on all series and experiments surveyed are reported in annex A, and it would be a repetitive exercise to report on each series in detail here. However, some series have particular characteristics that are worth discussing.

Series B and C

Series B and C are the two series with the largest number of replications, and the two series are so related that they are often described as one unit. Experiments in these series have been performed in various laboratories in both Europe and the USA: at the University of Maryland, the University of Kaiserslautern, the University of Strathclyde, the University of Bari, Linköping university and the Norwegian University of Technology in Trondheim, as well as with professionals both at Lucent and NASA. The replications in these two series have been analysed to some extent in report C:4 as well as in Wohlin *et al.* [r60, ch. 10]. The experiments are also often referred to after the place where they were performed, *c.f.* annex A.

Series C is an evolution of the experiments in series B, introducing new methods for requirements inspection. The first experiment, carried out at the NASA Space Flight Center, included a specific task in flight dynamics that was omitted from the subsequent replications.

Experiment C:2 is considered a replication [**?**] even though it scrutinises an alternative technique—without success, which prompts later replications to go back to the original technique.

The initials experiment in both series were also subject to statistical replication, but that data hasn't been reported separately and they are therefore shown only as aggregate experiments.

Series E

Experiment series E is interesting as the three experiments in the series were performed at the same time and by the same experimenters, with randomised tasks—*i.e.* the tasks are really neither **alternative** nor **improved**, but their variation is part of the original experiment set-up.

This series of experiments is therefore hard to classify; it has an experimental design different any other series. Due to the fact that the experiments were performed so closely together, the two replications were not literally used to confirm the findings of the first experiment; rather, the findings are the results of the three experiments combined. However, each of the individual results also support these findings.

The three experiments in the series however contained different tasks, so they cannot be qualified as a statistical replication (*c.f.* definition 2), and they aren't pre-tests or pilot tests. All three experiments were carried out with professionals from Bosch as subjects.

Series H and I

Series H consists of an original experiment and an internal, differentiated replication. The experience from this series was used to design experiment I:0.

Series I has two external replications: one close, and one differentiated. Interestingly, neither of these replications confirm the results of the original experiment.

It would appear that the cause of this is not a lack in the documentation of experiment I:0, but different profiles of the subjects in the experiment. It would have been interesting to see results from pre-tests, diagnostic tests or calibration tests in this series.

Series O

Series O is another example of a series of experiments that has only had professionals as subjects. However, in the original experiment, all subjects were employed with the same company and participated on a voluntary basis, whereas in the replication, they came from a variety of consultancies—and these consultancies were paid for the participation of their staff in the experiment.

Series P

Experiment P:0 is the oldest experiment in this survey. It was chosen for replication in P:1 primarily because it was well-documented and the researchers carrying out experiment P:1 were primarily interested in replication as such.

Series Q

This series consists of three experiments; however, the authors themselves write that 'experiment I served to hone the tools used [**?**]', and it had different tasks than experiments two and three. Experiments one and two were conducted using students as subjects, while the third experiment had professional subjects.

Series T

In series T, experiment T:1 was performed by the researchers performing the original experiment T:0, with students from another university as subjects. The experimental set-up was also improved for experiment T:1. However, the researchers describe replication T:1 as an external replication; this does not accord with the definition of an external replication used in this thesis.

5.7 Cross-comparisons

In their survey, Sjøberg *et al.* [r55, sec. 9] conclude that close replications, regardless of whether they were conducted by the original experimenters or other researchers, confirmed the original findings. Differentiated replications, however, seemed to differ in terms of confirming the original findings based on whether they were performed by the original authors or other researchers: six out of seven external, differentiated replications did not confirm the original findings, and the seventh replication only partly confirmed them. Internal, differentiated replications, on the other hand, exposed the opposite characteristics: seven replications confirmed the original findings, while the eigth reported partially different results. Sjøberg *et al.* [r55] do not propose any explanations for this disparity, but suggest that further research is necessary.

As the base of experiments of this survey is larger (50 *vs.* 34 experiments and 41 *vs.* 20 replications, respectively), these observations merit further study.

Similar, external replications

Referring back to table 4.1 and looking only at those replications that are performed by other experimenters than the original experiment and that can be

classified as close replications, we see that out of the four experiments (F:2, I:1, P:1 and R:1), two did not confirm the results of the original experiment. Replication P:1 was performed almost ten years after the original experiment, which could be considered a variation in itself; for replication I:1, the experimenters suggest that the profile of the subjects used might have been different—they were students on a more applied computer science programme that the subjects in experiment I:0.

Improved, internal replications

Another category that is interesting to study is that of replications by the same experimenters under different conditions, in different settings or with modified tasks. The sample here is larger, there are ten experiments that fit this description (B:4, C:3, D:1, J:1, L:1, M:1, N:1, Q:1 and 2, S:1). With exception of the inconclusive S series, these experiments also confirm the original findings.

Similar, internal replications

Not surprisingly, all instances of close, internal replications turned out to confirm the original results. This is consistent with the findings of Sjøberg *et al.* [r55, sec. 9]. In series E, G and H, the original experiment and replication were reported in the same publication, and one would expect a researcher to endeavour such consistency in the publicised review. Experiments F:1 and T:1, on the other hand, are reported in separate, later, publications. The experiments where conducted with a slightly different group of student subjects (F:1) and more time (T:1) respectively. The results 'increase [. . .] confidence [in the] the result of the original experiment **[?]**' which was the objective of replicating the experiment.

Differentiated, external replications

It its in this category, then, that the vast majority of replications do not confirm the original findings completely. This leads to the suspicion that it may be dangerous to vary too many factors at the same time. In fact, experiment C:1 is the only replication in this category that fully reproduced the results of the original study; in this experiment, the variations to the experiment materials and tasks where not very extensive, and the researchers describe that they were in close contact with the original experimenters to ensure a successful replication.

Studying the reports from differentiated, external replications, it becomes clear that there are some experiments where the researchers did not expect to

confirm the original results (such as experiment K:1), or where the experiment was changed to such an extent that it becomes arguable if the new experiment is a replication at all (*e.g.* experiment C:2). However, the share of successful replications in this group remains small even when these instances are removed from the sample.

Chapter 6

Qualitative analysis

This chapter will look at how the researchers who have performed the experiments studied in this survey describe their their motivations for replicating experiments as well as their their description of the procedure of replication.

6.1 Motivation

The need for replications has been discussed extensively in the literature, as described in section 2.3. This sentiment is reflected by researchers in the field: 'Since it is not possible to draw final conclusions from a single experiment, we conducted a replication of the experiment.' [**?**]

Several authors argue that close replications alone are not sufficient to alleviate all the shortcomings of an experiment; 'it is necessary to perform replications of the original experiment in similar or slightly different experimental settings' [**?**]. 'Another problem with experiments relates to the scale and plausibility of the materials [**?**]'.

Another concern that researchers try to address by replication is that of low statistical power, notably because of the small size of the samples 'of convenience [**?**]' often used in these experiments.

Researchers therefore 'encourage the external replication of [the] study in different environments [and] by different researchers [**?**].'

6.2 Validity

One of the main motivations for replication, as has been discussed in chapter 2, is to increase the validity of the findings. Validity is a general term that refers to whether the results of an experiment are valid, *i.e.* whether the conclusions drawn from the experiment follow logically from the experiment setup and its results.

> **Validity** The extent to which a measurement instrument or test accurately measures what it is supposed to measure.

Definition 6: Validity

The reports I have reviewed also refer to validity as an important motivation for performing replications, as well as a key element in encouraging more replications. 'We realized that there are some threats to validity [. . .] Some of the threats might even only be addressed through replication.' [**?**]

To study the validity of given results, it is common to split the concept of validity into several components. One common approach is to refer to internal validity (*c.f.* definition 9), construct validity (*c.f.* definition 8) and external validity (*c.f.* definition 7). However, many researchers prefer to discuss the concept of validity only in terms of internal and external validity.

External validity

As I have described before, the effect on external validity is the main motivation cited for replication. Many researchers are aware of the threats to external validity that experimentation brings with it, for example because it often 'is not practical to use random samples from a population [**?**]'. 'These threats can only be addressed by replicating and reproducing these studies. Each new run reduces the probability that our results can be explained by human variation or experimental error.' [**?**]

> **External validity** The extent to which a finding applies (or can be generalized) to persons, objects, settings, or times other than those that were the subject of study.

Definition 7: External validity

Several researchers discuss external validity by discussing if the results from an experiment can be generalised, *i.e.* if the results apply 'to the population of interest in the hypothesis [**?**]' and 'the real environment in which the technique should be applied [**?**]'. Replication can contribute significantly both to 'results of the original experiment' and to 'generalizing results [**?**]'.

In some instances, the concerns about whether the results of an experiment can be generalised at all have led researchers to replicate their experiment before publishing any results [**? ? ? ?**], or through 'further investigation [**?**]' of 'issues not covered in the previous [experiment] [**?**]'.

Construct validity

It is interesting to note that the discussion on validity in the reports surveyed is centred predominantly on external validity. However, a well-performed replication must also evaluate the methods used to capture data in the original experiment; many experiments do, not least those experiments that have an improved or alternative method in terms of Brooks *et al.* [r8]. Such alterations imply a look at the construct validity of the original experiment; the necessity of this element in replication was underlined by Deligiannis *et al.* [r18].

> **Construct validity** The extent to which a test may be said to measure that which it has been designed to measure.

Definition 8: Construct validity

Some researches do discuss to this element of replication, sometimes as an aspect of internal validity: 'The replications should address changes in the design, for example, use a different domain, and seed more faults into the document under inspection.' [**?**] 'However, we did change the way in which the dependent variable was measured [. . .] By carrying out this kind of replication in which the same hypothesis is studied, but some details of the experiment are changed, our aim is to make the results of the experiment more reliable.' [**?**] 'A replicated study that allows certain variables to vary from previous experiments in carefully controlled ways provides further evidence of which aspects of the experiment interact with the software engineering aspect being evaluated. Such replicated studies give additional confidence in the experimental design.' [**?**].

Another related issue is that of 'scale and plausibility of the materials' [**?**].

Internal validity

Due to the nature of the concept of internal validity, replication (in the sense used in this thesis, *c.f.* subsection 2.3) as such doesn't affect the internal validity of the experiment. However, controlled experiments in general have greater control can therefore 'be used to confirm results obtained in field studies, where control and, therefore, internal validity is usually weaker [**?**]'. External replications also help ruling out the possibility of researcher bias [**?**].

A number of experiments use statistical replication (*c.f.* defintion 2) internally in their experiments, e.g. the experiments in series A. This is often done to increase the internal validity of the experiment.

Internal validity The certainty with which results of an experiment can be attributed to the manipulation of the independent variable rather than to some other, confounding variable.

Definition 9: Internal validity

6.3 Using students as subjects

Another issue that is discussed in a large number of articles is the issue of using students (be they at bachelor's or degree level) in software engineering experiments. This issue has been discussed as an issue in itself in numerous articles [r24, r4, r10, r11]. Many researchers regard the student setting as an effective testing ground: 'Initially we use students rather than professional[s] because cost considerations severely limit our opportunities to conduct studies with professional developers. Therefore we prefer to refine our experimental designs and measurement strategies in the university before using them in industry. This approach also allows us to do a kind of bulk screening of our research hypotheses. That is, we can conduct several studies in the university, but only rerun the most promising ones in industry. Intuitively, we feel that hypotheses that don't hold up in the university setting are unlikely to do so in the industrial setting.' [**?**]

Some researchers regard the student setting to be insufficient [**? ?**], while others argue that the body of knowledge mean researchers should stop 'disregarding studies done with student subjects', even though this doesn't mean 'that studies with professionals are no longer needed [**?**]'. All the same, many researchers wish to replicate their experiments in professional settings [**? ? ? ?**].

6.4 How to replicate

The level of detail with which researchers report their experience with the actual operation of the replication varies greatly. I was mainly interested in three elements in such reports: the guidelines the researchers used to plan and perform the replication; how they chose the the experiment for replication and how they obtained the material, and finally, what factors are decisive for successful replication.

Guidelines

In terms of guidelines referenced, the works of Wohlin *et al.* [r60] and Brooks *et al.* [r7, r8] as well as the statistics coursebook by Judd *et al.* [r28] dominate;

they are both referenced in 5 reports. The framework of Basili *et al.* [r2] has become less widely used.

A number of other guidelines are cited more than once: a book on statistic (Campbell and Stanley [r9]), two texts by researchers in the field (Daly [r15], Basili *et al.* [r3]), a book on experimentation in software engineering (Juristo and Moreno [r30]) as well as an older article by Kaplan and Duchon [r32]. All but the last two have are discussed in chapter 2.

Out of 31 replications studied, only five (*i.e.* one sixth) refer to a common set of references for carrying out these replications. One might add to this observation that the authors of the guidelines and the researchers performing the replications coincide to a large degree. This underpins the assessment that experimentalists still rely on frameworks from other disciplines in lack of an established empirical framework within the discipline [r41], even though proposals for such frameworks within the discipline exist [r2, r37].

Choosing an experiment to replicate

In cases where the researchers managed to obtain enough materials to perform a close replication, there have often been personal or institutional contacts between the original experimenters and the replicators.

In some cases, the materials are only available in another language than English. Some experiments also use programming languages that are no longer widely used—even though the object of study may still be of great relevance.

It has been suggested that the empirical software engineering community should establish a set of ‘ “classical” experiments that could be easily replicated and modified [**?**]’, and which would be suitable for replication by degree students. Others, however, question the value of such replications [r8].

Lab packages

In order to facilitate replication, many research groups have made their materials available as ‘lab packages’, often in the hope to encourage ‘future work on [the] subject not only by the same researchers but also by independent researchers [**?**]’.

I have unfortunately not been able to look deeper into these lab packages. A tentative listing is available in appendix C. From a first impression, the amount of materials available varies greatly, and a number of researchers who have performed external replications a have noted that the materials provided weren’t sufficient to allow for a close replication [**?**].

The content of the lab packages appears to vary quite significantly. One suggestion [**?**] of what they should contain is:

- Motivation for the study
- Detailed description of the technique
- Design of the study
- Results of the study

Furthermore, in order to maximise the potential benefits of lab packages, it is important not only to include the tasks and materials, experiment design and findings, but also the raw data collected, so as to allow for meta-analysis methods to be used [**? ?**]

Series and families of experiments

Many of the experiments in this survey are part of larger research programmes, involving series of experiments, or families of experiments, and researchers from different institutions [**? ?**]. Some of these programmes also involve other empirical methods such as case studies and surveys [**? ? ? ? ? ?**], because 'the weaknesses of one study can be addressed by the strengths of another [**?**]'. Basili *et al.* [r3] in particular have argued for 'families of experiments' to improve the research output from experiments beyond the results from mere replication.

Research networks

Another point worth mentioning is the International Software Engineering Research Network (ISERN), a network of research institutions in the field of empirical software engineering. One of the activities of this network is a common publication service for technical reports, where experiments can be documented in greater detail than what is possible in articles in scientific journals. Some researchers refer to such auxiliary publication in the reports I have studied [**?**], and many of the institutions represented in this study are members of ISERN.

6.5 Reporting the replication

An issue that is crucial both in the publication of original experiments and replication is the reporting of the experiment. For further replication to be possible, the reports must contain a certain amount of information about the experiment.

The guidelines in the book by Wohlin *et al.* [r60] have become quite universally accepted in the reporting of experiment set-up, and are widely used. The framework of Basili *et al.* [r2] has become less widely used, and hence, unfortunately, the extensions concerning replication put forth by Brooks *et al.*

[r8] are not widely used, but give good guidance to reading reports written according to these guidelines.

As noted above, it is quite common to publish additional material necessary for a replication in lab packages or technical reports, as the amount of material doesn't fit into the constraints presented by publication in scientific journals and conferences.

Publication bias

One report makes an explicit allusion to the possibility of a publication bias Basili *et al.* [r3], Miller [r41] against replications: 'While the broad consensus of the sources reviewed agrees that replicating experiments is paramount to the maturation of the software engineering discipline, some authors suggest that replications still are seen as less important as "original work" in the very strict sense [**?**].'

To establish whether such a bias exists is unfortunately outside the possibilities of this thesis; it would require a different form of study, *e.g.* a case study involving interviews with researchers and review panels of academic journals.

6.6 Meta-analysis

Many researchers argue that in order to make progress in the field of empirical software engineering, results from several experiments and indeed from other empirical studies such as surveys and case studies have to be combined. One method of thus combining results is statistical meta-analysis.

By combining data from several studies with meta-analysis tools, the results can be usefully combined 'to achieve significant, reliable and generalisable results [**?**]'. Some researchers argue that statistical data should more extensively shared among researchers; possibly by means of a type of public database [**? ?**].

Chapter 7

Threats to validity

Validity is one of the central issues in this thesis, and as I have outlined in the previous chapters, on of the main drivers for replication. Validity is often divided into several sub-categories. There is no generally agreed set of such subcategories, but the following three are commonly referred to in the literature [r50, r29].

Definitions of core aspects of validity have been given in the previous chapter: internal validity (*c.f.* definition 9), construct validity (*c.f.* definition 8) and external validity (*c.f.* definition 7).

7.1 Internal validity

The classification of experiments in terms of the topic under scrutiny as well as the categorisation along the axes internal–external and in terms of the Brooks' classification have been done by myself and have not been verified by another, independent researcher. This might have been desirable.

For those series that were present in the work of Sjøberg *et al.* [r55], I have reviewed the classifications made for the purpose of that study; these classifications have been subject to review by several researchers. However, in some instances, I have decided to classify differently based on my own review. These decisions are documented in the annex; *e.g.* the experiments in series C.

7.2 External validity

Given the reservations I have made in the previous section on internal validity, I believe that the external validity of the findings contained herewithin is strong. Chapter 3 describes the method and procedure I have used to identify the largest possible number of series of replicated experiments; *i.e.* the totality of such experiments that have been reported in scientific publications.

The search strategy for this thesis, as discussed in chapter 3, was a best-effort strategy based on searches of abstracts and keywords. Other strategies such as manual reviewing of abstracts might have been even more precise. For example, an article by Müller [r42] was found later, but no variant of the word 'replication' was present neither in the keywords nor in the abstract. In-press articles such as Staron *et al.* [r57] unfortunately confirm this trend. The term 'series of experiments' might have been a useful addition to the search word array.

7.3 Construct validity

The concept of construct validity is not directly applicable to this survey: there are no explicit research questions or hypotheses, and hence, no constructs to validate.

The reader may argue that this is the main weakness of this thesis. On the other hand, it was not possible to formulate precise and relevant research questions on the topic of experiment replication in empirical software engineering, because there was no empirical material on the field available. The publications that did exist either described how replication *should* be done in general (*c.f.* section 2.3), how it was done in a particular case (*c.f.* section 6.4) or that more information on the field is needed (*c.f.* section 2.2).

In this sense, I hope that the present thesis can contribute to better studies of the field in the future.

Chapter 8

Conclusions

In the course of this survey, I have analysed a very large number of scientific publications on replication of software engineering results. It has been a very interesting project (even though it took a long time to complete), and I believe I have extracted some valuable ideas on the replication of controlled experiments in empirical software engineering. As I have noted above, this was the first survey on this subject, and therefore the survey was conducted with an open mind and without specific hypotheses to be tested.

8.1 Results

A number of conclusions can be drawn from the data collected in this survey. However, some of these conclusions may need to be subjected to independent verification.

The number of unsuccessful replications has been reduced in the past few years. Unfortunately, it is not clear if this can be ascribed to publication bias, better reporting of the original experiments or better procedures in replication. In either case, there have not been differentiated, external replications in the last few years.

As seen in section 5.7, it appears that differentiated, external replications are a risky affair. This is in accordance with the consideration of Brooks *et al.* [r8]: instead of changing elements in the experimental set-up, 'it may be more important to confirm first the original results'.

The discussion about reporting standards is still going on. I have made some recommendations below on what seems to be a common understanding of good reporting standards; standards that will allow scrutiny of the experiments as well as replication and meta-analysis.

The study also shows that there is no one experiment series that has been replicated in all four possible combinations of close–differentiated and internal–external.

8.2 Recommendations

The main elements of previous work in the area are general guidelines on empirical methods in software engineering research, and guidelines on replications from other areas. During the survey that I performed, I also made some observation on the reporting of replicated experiments that may be valuable for future reports; recommendations that will hopefully make future surveys easier to carry out.

Guidelines for replication in empirical software engineering

It appears from chapter 2 and section 6.4 that very little has been written so far on how to replicate empirical software engineering experiments. The guidelines that are available are either *ad-hoc* write-ups, footnotes to general experimenting guidelines or instructions borrowed from other fields. Even within these categories, experimenters struggle to find useful tools for their work, and there is little consensus in the field regarding which guidelines for replications should be used. In terms of general experimental guidelines, the book by [r60] for example, seems to be widely accepted; therefore one can hope that a set of replication guidelines may in the future be accepted universally. It is obviously out of the scope of this thesis and beyond my experience to propose such guidelines, but I hope that this thesis can be a valuable contribution.

Reporting experiments

The purpose of reporting an experiment is three-fold. A report should allow other researchers to (i) evaluate the experiment as such, (ii) replicate the experiment, and (iii) use the results for meta-analysis. The combination of these perspectives translate into a number of requirements on reports from controlled experiments (and indeed other empirical studies) in software engineering. For example, the reports should be written in a way that accommodates readers with any of the three interest above. Many of the reports I studied were apparently written only for readers interested in the software engineering results, and not for readers interested in the empirical methods used.

Lab packages

Lab packages are obviously a very important element in making sure that experiments are replicable. Whenever possible, lab packages should not only contain the materials used in the experiment itself, but also other relevant material such as training materials, calibration tests and debriefing questionnaires. The previous work and some of the more extensive reports in the survey contain suggestions of how

Statistical data

Raw data from the experiment can also be provided as it enables the replicators to proceed to meta-analysis of the results, as well as allowing interested parties to analyse the data, possibly in new and different ways.

The results of pre-tests or pilot tests, calibration tests and other auxiliary tests should also be reported, as they allow for a better understanding of the data.

Identifying replications

As I have noted in section 7.2, in many cases, the abstracts and keywords used to index reports of experiments did not contain the term replication. As bibliometric and systematic reviews and surveys are powerful secondary research tools, steps should be undertaken to make it easy to identify relevant work in databases and meta-databases such as ELIN@.

Further experiments and replication

In terms of confirming the results of this survey, it would be desirable to perform an experiment series with the following characteristics:

- an original experiment
- a close, internal replication
- a differentiated, internal replication
- a close, external replication
- a differentiated, external replication

No such series of experiments exists today (ideally, the two external replications would be performed by the same researchers). Some of the conclusions I have drawn above could be effectively tested with such a set-up.

8.3 Further work

This section is an overview of possible future research on the use of replication in empirical software engineering. I have not given proposals for new experiments here because of the complexity of such an undertaking; the following areas might be suitable for master's theses in computer science.

Construct validity

An aspect of replication that may warrant further study is the perspective on construct validity as discussed in subsection 6.2. External replications in particular will involve an analysis of the methods and metrics of the original experiment. Why do researchers choose to change the research method, metrics and hypotheses, and how do their results compare to the original results? Can they be combined by using meta-analysis tools? How do series of experiments evolve as the methods are refined?

Meta-analysis

Another possible area of further work is to meta-analyse results from replicated controlled experiments in empirical software engineering.

Initial work on meta-analysis in general has been begun by several researchers, *e.g.* Miller [r40], Pickard *et al.* [r46], Hayes [r23] and within the CONTEXT project, by Dybå *et al.* [r19]. It might be of interest to use such methods to study replicated experiments in particular, for instance the large experiment series labelled B and C in this survey. This could help refine the results and the experimental methods as well as establishing what levels of data reporting are needed.

Easing replication

A systematic review of replicated and non-replicated experiments, respectively, could further help to identify why some experiments are replicated when others are not? Possibly, this is an artificial question—if replications only are caused by personal relations between the experimenters. That, however, would be an interesting result as such, which could allow us to question the 'externality' of external replications.

Replication of other empirical studies

During my search for publications on replicated controlled experiments, I also found a number of other empirical studies in software engineering, such as

surveys and case studies, that have been replicated. It would be of interest to survey these, in particular the case studies, in a manner similar to this thesis and investigate if the findings of this thesis are supported by such a study.

References

The following references are the related work and other bibliography identified during the writing of this thesis.

[r1] Duane A. Bailey. A letter to research students. Letter.

[r2] Victor R. Basili, Richard W. Selby, and David H. Hutchens. Experimentation in software engineering. *IEEE Transactions on Software Engineering*, 12(7):733–743, 1986.

[r3] Victor R. Basili, Forrest J. Shull, and Filippo Lanubile. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*, 25(4):456–473, July/August 1999. doi:10.1109/32.799939.

[r4] Patrik Berander. Using students as subjects in requirements prioritization. In *Proceedings of the International Symposium on Empirical Software Engineering (ISESE'04)*, pages 167–176, 2004. doi:10.1109/ISESE.2004.1334904.

[r5] Pierre Bourque and Vianney Côte. An experiment in software sizing with structured analysis metrics. *Journal of Systems and Software*, 15:159–172, 1991. doi:10.1016/0164-1212(91)90053-9.

[r6] George E. P. Box, William G. Hunter, and J. Stuart Hunter. *Statistics for experimenters*. Wiley, New York, NY, USA, 1978. ISBN 0-471-09315-7.

[r7] Andrew Brooks, John William Daly, James Miller, Marc Roper, and Murray Wood. Replication's role in experimental computer science. Technical Report EFoCS-5-94[1] [RR/94/172], Department of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 1994.

[r8] Andrew Brooks, John William Daly, James Miller, Marc Roper, and Murray Wood. Replication of experimental results in software engineering. Technical Report ISERN-96-10, Department of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 1996.

[r9] Donald Thomas Campbell and Julian Cecil Stanley. *Experimental and Quasi-Experimental Designs for Research*. Houghton Mifflin Co., Boston, MA, USA, 1966. ISBN 0-395-30787-2.

[1] According to the original author, this is "basically the same" as [r8]

[r10] Jeffrey Carver, Letizia Jaccheri, Sandro Morasca, and Forrest J. Shull. Issues in using students in empirical studies in software engineering education. In *Proceedings of the 9th International Software Metrics Symposium (METRICS2003)*, pages 239–249, 2003. doi:10.1109/METRIC.2003.1232471.

[r11] Marcus Ciolkowski, Dirg Muthig, and Jörg Rech. Using academic courses for empirical validation of software development processes. In *Proceedings of the 30th Euromicro Conference*, pages 354–361, 2004. doi:10.1109/EUROMICRO.2004.83.

[r12] William G. Cochran and Gertrude M. Cox. *Experimental designs*. Wiley, New York, NY, USA, 1950. ISBN 0-471-54567-8.

[r13] Samuel D. Conte, Vincent Y. Shen, and Hubert E. Dunsmore. *Software Engineering Metrics and Models*. Benjamin Cummings, Menlo Park, CA, USA, 1986. ISBN 0-8053-2162-4.

[r14] Thomas D. Cook and Donald Thomas Campbell. *Quasi Experimentation: Design and Analysis Issues for Field Settings*. Houghton Mifflin, Boston, MA, USA, 1979. ISBN 0-395-30790-2.

[r15] John William Daly. *Replication and a Multi-Method Approach to Empirical Software Engineering Research*. Ph. D. thesis, Department of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, March 1996. `http://www.cis.strath.ac.uk/research/efocs/papers/jdthesis.pdf`.

[r16] John William Daly, Andrew Brooks, James Miller, Marc Roper, and Murray Wood. Verification of results in software maintenance through external replication. In *Proceedings of the International Conference on Software Maintenance*, pages 50–57. IEEE Comput. Soc. Press, 1994.

[r17] John William Daly, Khaled El Emam, and James Miller. An empirical research method for software process improvement. Technical Report ISERN-97-04, Fraunhofer Institut (IESE), Kaiserslautern, Rheinland-Pfalz, Germany, 1997.

[r18] Ignatios S. Deligiannis, Martin Shepperd, Steve Webster, and Manos Roumeliotis. A review of experimental investigations into object-oriented technology. *Empirical Software Engineering*, 7(3):193–231, 2002. doi:10.1023/A:1016392131540.

[r19] Tore Dybå, Vigdis By Kampenes, and Dag I. K. Sjøberg. A systematic review of statistical power in software engineering experiments. Submitted to *Journal of Information & Software Technology*, in press, 2005. doi:10.1016/j.infsof.2005.08.009.

[r20] Description of Empirical Software Engineering—An International Journal. `http://www.springeronline.com/journal/10664/about`.

[r21] Robert L. Glass, Iris Vessey, and Venkataraman Ramesh. Research in software engineering: an analysis of the literature. *Information and Software Technology*, 44(8):491–506, 2002. doi:10.1016/S0950-5849(02)00049-6.

[r22] Ove Hansen. Survey of controlled software engineering experiments with focus on subjects. Master's thesis, University of Oslo, Oslo, Norway, 2004. http://urn.nb.no/URN:NBN:no-9897.

[r23] Will Hayes. Research synthesis in software engineering: A case for meta-analysis. In *Proceedings of the 6th International Software Metrics Symposium (METRICS1999)*, pages 143–151, 1999. doi:10.1109/METRIC.1999.809735.

[r24] Martin Höst, Björn Regnell, and Claes Wohlin. Using students as subjects–a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5(3):201–214, 2000. doi:10.1023/A:1026586415054.

[r25] IEEE standard glossary of software engineering terminology. IEEE Std 610.12-1990, 1990.

[r26] ISERN basic terminology. http://www.cs.umd.edu/projects/SoftEng/tame/isern/isern.definitions.html.

[r27] Magne Jørgensen. A review of studies on expert estimation of software development effort. *Journal of Systems and Software*, 70(1-2):37–60, 2004. doi:10.1016/S0164-1212(02)00156-5.

[r28] Charles M. Judd, Eliot R. Smith, and Loiuse H. Kidder. *Research Methods in Social Relations*. Holt, Rinehart and Winston, Orlando, FL, USA, 6th edition, 1991. ISBN 0-03-031149-7.

[r29] Charles M. Judd, Eliot R. Smith, and Loiuse H. Kidder. *Research Methods in Social Relations (international edition)*. Harcourt Brace Jovanovich College Publishers, Fort Worth, TX, USA, 6th edition, 1991. ISBN 0-03-032977-9.

[r30] Natalia Juristo Juzgado and Ana María Moreno. *Basics of Software Engineering Experimentation*. Kluwer Academic Publishing, Boston, MA, USA, 2001. ISBN 0-7923-7990-X.

[r31] Natalia Juristo Juzgado, Ana María Moreno, and Sira Vegas. Reviewing 25 years of testing technique experiments. *Empirical Software Engineering*, 9(1-2):7–44, 2004. doi:10.1023/B:EMSE.0000013513.48963.1b.

[r32] Bonnie Kaplan and Dennis Duchon. Combining qualitative and quantitative methods in information systems research: A case study. *MIS Quarterly*, pages 571–586, December 1988.

[r33] Barbara A. Kitchenham. Procedures for performing systematic reviews. Technical Report TR/SE-0401, Department of Computer Science, Keele University, Keele, Staffs, UK, July 2004. Available from http://ease.cs.keele.ac.uk/sreview.doc.

[r34] Barbara A. Kitchenham, Shari Lawrence Pfleeger, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8):721–733, August 2000. doi:10.1109/TSE.2002.1027796.

[r35] Nils-Kristian Liborg. A study of threats to validity in controlled software engineering experiments. Master's thesis, University of Oslo, Oslo, Norway, 2004. http://urn.nb.no/URN:NBN:no-9890.

[r36] R. Murray Lindsay and A. S. C. Ehrenberg. The design of replicated studies. *American Statistician*, 47(3):217–228, August 1993.

[r37] Christopher M. Lott and H. Dieter Rombach. Repeatable software engineering experiments for comparing defect-detection techniques. *Empirical Software Engineering*, 1(3):241–277, 1996. doi:10.1007/BF00127447.

[r38] Jeffrey W. Lucas. Theory-testing, generalization, and the problem of external validity. *Socological Theory*, 21(3):236–253, September 2003. doi:10.1111/1467-9558.00187.

[r39] Information about the ELIN@ system. http://www.lub.lu.se/headoffice/elininfo.shtml.

[r40] James Miller. Applying meta-analytical procedures to software engineering experiments. *Journal of Systems and Software*, 54(1):29–39, 2000. doi:10.1016/S0164-1212(00)00024-8.

[r41] James Miller. Replicating software engineering experiments: a poisoned chalice or the Holy Grail. *Information and Software Technology*, 47:233–244, 2005. doi:10.1016/j.infsof.2004.08.005.

[r42] Matthias M. Müller. Two controlled experiments concerning the comparison of pair programming to peer review. *Journal of Systems and Software*, 78(2):166–179, 2005. doi:10.1016/j.jss.2004.12.019.

[r43] Ikujiro Nonaka and Noboru Konno. The concept of "ba": Building a foundation for knowledge creation. *California Management Review*, 40(3): 40–54, spring 1998.

[r44] Shari Lawrence Pfleeger. Design and analysis in software engineering, part 1: the language of case studies and formal experiments. *ACM SIGSOFT Software Engineering Notes*, 19(4):16–20, 1994. doi:10.1145/190679.190680.

[r45] Shari Lawrence Pfleeger. Experimental design and analysis in software engineering, part 2: How to set up an experiment. *ACM SIGSOFT Software Engineering Notes*, 20(1):22–26, January 1995. doi:10.1145/225907.225910.

[r46] Lesley M. Pickard, Barbara A. Kitchenham, and Peter W. Jones. Combining empirical results in software engineering. *Information and Software Technology*, 40(14):811–821, 1998. doi:10.1016/S0950-5849(98)00101-3.

[r47] Colin Robson. *Real world research*. Blackwell, Oxford, United Kingdom, 1st edition, 1993. ISBN 0-631-17688-8.

[r48] Ralph L. Rosnow and Robert Rosenthal. Statistical procedures and the justification of knowledge in psychological science. *American Psychologist*, 44(10):1276–1284, 1989. doi:10.1037/0003-066X.44.10.1276.

[r49] Kristian Sandahl, Ola Blomkvist, Joachim Karlsson, Christian Krysander, Michael Lindvall, and Niclas Ohlsson. An extended replication of an experiment for assessing methods for software requirements inspection. *Empirical Software Engineering*, 3(4):327–354, 1998. doi:10.1023/A:1009724120285.

[r50] William R. Shadish, Thomas D. Cook, and Donald T. Campbell. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin, Boston, MA, USA, 2002. ISBN 0-395-61556-9.

[r51] Forrest Shull, Manuel G. Mendonça, Victor R. Basili, Jeffrey Carver, José C. Maldonaldo, Sandra Fabbri, Guilherme Horta Travassos, and Maria Cristina Ferreira. Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9:111–137, 2004. doi:10.1023/B:EMSE.0000013516.80487.33.

[r52] Homepage of the CONTEXT project. `http://home.simula.no/project_one.php?project_id=42`.

[r53] Homepage of the Simula Research Lab. `http://www.simula.no/`.

[r54] Dag I. K. Sjøberg, Bente Anda, Erik Arisholm, T. Dybå, Magne Jørgensen, Amela Karahasanović, Espen F. Koren, and Marek Vokáč. Conducting realistic experiments in software engineering. *Proceedings of the International Symposium on Empirical Software Engineering (ISESE'02)*, pages 17–26, 2002. doi:10.1109/ISESE.2002.1166921.

[r55] Dag I. K. Sjøberg, Jo E. Hannay, Ove Hansen, Vigdis By Kampenes, Amela Karahasanović, Nils-Kristian Liborg, and Anette C. Rekdal. A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering*, 31(9):733–753, September 2005. doi:10.1109/TSE.2005.97.

[r56] Ian Sommerville. *Software Engineering*. Pearson Education, Harlow, Herts, UK, 6th edition, 2001. ISBN 0-201-39815-X.

[r57] Miroslaw Staron, Ludwik Kuzniarz, and Claes Wohlin. Empirical assessment of using stereotypes to improve comprehension of UML models: A set of experiments. *Journal of Systems and Software.* In Press, Corrected Proof, 2005. doi:10.1016/j.jss.2005.09.014.

[r58] Thomas Thelin and Per Runesson. Prospects and limitations for cross-study analyses—a study on an experiment series. In *Proceedings of the 2nd Workshop Series on Empirical Software Engineering—The Future of Empirical Studies in Software Engineering*, pages 133–142, 2003.

[r59] Walter F. Tichy, Paul Lukowicz, Lutz Prechelt, and Ernst A. Heinz. Experimental evaluation in computer science: A quantitative study. *Journal of Systems and Software*, 28(1):9–18, 1995. doi:10.1016/0164-1212(94)00111-Y.

[r60] Claes Wohlin, Per Runesson, Martin Höst, Magnus C. Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in Software Engineering: An Introduction*. Kluwer Academic Publishers, Boston, MA, USA, 2000. ISBN 0-7923-8682-5.

[r61] Murray Wood, John William Daly, James Miller, and Marc Roper. Multi-method research: An empirical investigation of object-oriented systems. *Journal of Systems and Software*, 48(1):13–26, 1999. doi:10.1016/S0164-1212(99)00042-4.

[r62] Marvin V. Zelkovitz and Dolores Wallace. Experimental validation in software engineering. *Information and Software Technology*, 39(11):735–743, 1997. doi:10.1016/S0950-5849(97)00025-6.

[r63] Andreas Zendler. A preliminary software engineering theory as investigated by published experiments. *Empirical Software Engineering*, 6(2): 161–180, 2001. doi:10.1023/A:1011489321999.

DOI references (digital object identifiers) may be resolved using the web site of the International DOI Foundation at `http://www.doi.org/`

ISERN technical reports are available from `http://www.iese.fhg.de/network/ISERN/pub/isern_biblio_tech.html`

Appendices

Appendix A

Survey details

Series A

Experiment A:0

Authors/reference:	**[?]**
Type:	original
Motivation:	–
Guidelines:	Basili *et al.* [r2]
Materials (other tasks):	several relatively complex tasks
Changes (more time? debrief?):	–
Encouragement of further replication:	Mentions that they will
Dates:	autumns of 1982, 1983 and 1984
Subjects (experience, country, university):	intermediate-level students at the University of Maryland; professionals at NASA and Computer Sciences corporations
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

This is actually three experiments, but the results have been combined into a common report.

Experiment A:1

Authors/reference:	**[?**]
Type:	replication
Motivation:	validate results
Guidelines:	Daly *et al.* [r16]
Materials (other tasks):	The materials used in the experiment are published as a technical report.
Changes (more time? debrief?):	Materials changed to C, more tasks,
Encouragement of further replication:	Yes, encouraged by materials in the tech report.
Dates:	Summer and autumn 1994
Subjects (experience, country, university):	50 students from the university of Kaiserslautern
Tasks & materials (measurement changed how?):	Materials translated to C. Added a fault isolation task.
Hypothesis or research questions changed? added?	only additions to account for extra task
Confirms results:	yes and no
Brooks *et al.* classification [r8]:	**(improved, alternative, similar)**

Also in this case, the experimenters actually ran three experiments (replications)

"We hope that our efforts will make it possible for the experiment to become a standard exercise that developers will use to evaluate and sharpen their defect-detection skills."

"We found dramatic differences [. . .] for the efficiency of observing failures."

Experiment A:2

Authors/reference:	**[?**]
Type:	replication
Motivation:	validate and generalise results
Guidelines:	Brooks *et al.* [r7]
Materials (other tasks):	from experiment A:1
Changes (more time? debrief?):	Same as A:1 but less strict statistical testing
Encouragement of further replication:	yes, reference to materials from A:1
Dates:	not stated. Probably 1996
Subjects (experience, country, university):	47 intermediate students at the University of Strathclyde
Tasks & materials (measurement changed how?):	See A:1
Hypothesis or research questions changed? added?	only additions
Confirms results:	yes and no
Brooks *et al.* classification [r8]:	**(improved, alternative, similar)**

"Replication is a critical component in providing empirical foundations - it is necessary both to validate earlier results and to 'recipe improve', where the experimental parameters are varied in a controlled manner, perhaps focusing on specific aspects of previous studies or, alternatively, attempting to generalise earlier findings."

"This work also demonstrates the importance of replication. If software engineering is to be based on substantial empirical evidence then replication is necessary to demonstrate that results are reliable and generalisable. The empirical study reported in this paper has now been carried out in various forms at five different sites over 20 years. Evidence from those studies was used to explain and substantiate the current research findings. "

"In the future it is hoped that further replications of this work will be carried out. All the materials necessary to run the experiment are available. They are also all available in electronic form. It was found that running the experiment within a practical software engineering class was a useful educational tool. "

Series B

Experiment B:0

Authors/reference:	[**?**] *Maryland-95*
Type:	original
Motivation:	–
Guidelines:	Judd *et al.* [r28]
Materials (other tasks):	– this is the original
Changes (more time? debrief?):	– no, this is the original
Encouragement of further replication:	strongly encouraged
Dates:	spring 1993, autumn 1993
Subjects (experience, country, university):	24 graduate students x 2
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"Until now, however, there have been no reproductible, quantitative studies comparing alternative detection methods for software inspections. We have conducted such an experiment and our results demonstrate that the choice of fault detection method significantly affects inspection performance. Furthermore, our experimental design may easily be replicated by interested researchers."

"There are several threats to this experiment's external validity. These threats can only be addressed by replicating and reproducing these studies. Each new run reduces the probability that our results can be explained by human variation or experimental error. Consquently, we are creating a laboratory kit (i.e., a package containing all the experimental materials, data, and analysis) to facilitate replication. The kit is available via anonymous ftp [. . .]"

"Finally, we are using the lab kit to reproduce the experiment with other university researchers in Japan, Germany, Italy and Australia and with industrial developers

at AT& T Bell Laboratories[1] and Motorola Inc. These studies will allow us to evaluate our hypotheses with different populations of programmers and different software artefacts."

The report contains a replication but data not presented as two distinct sets.

Experiment B:1

Authors/reference:	[**?**] *Bari*
Type:	replication
Motivation:	improve validity
Guidelines:	Judd *et al.* [r28] Campbell and Stanley [r9]
Materials (other tasks):	"Our strict replication of the Porter, Votta, and Basili experiment was made possible by the availability of the experimental material, prepared by the original experimenters in the form of a laboratory package."
Changes (more time? debrief?):	as little as possible (translation to italian!)
Encouragement of further replication:	yes
Dates:	spring 1995
Subjects (experience, country, university):	30 undergrads
Tasks & materials (measurement changed how?):	transl to italian, minor adjusts
Hypothesis or research questions changed? added?	same as orig
Confirms results:	No.
Brooks *et al.* classification [r8]:	**(similar, alternative, alternative)**

"We found these experimental results [**?**],[2] and their implications on the inspection process, very interesting. However, since it is not possible to draw final conclusions from a single experiment, we conducted a replication of the experiment of Porter, Votta and Basili.

A comprehensive definition of replications is in [r28]

> "Replication means that other researchers in other settings with different samples attempt to reproduce the research as closely as possible. If the results of the replication are consistent with the original research, we have increased confidence in the hypothesis that the original study supported."

Software engineering, as a scientific discipline, needs research whose primary purpose is replication. Such research is especially concerned with external validity, i.e. the extent to which we can generalize the results to the population of interest in the hypothesis. Frequently, in software engineering research we are not able, or it is not practical, to use random samples from a population in order to increase our ability to generalize. Generalization must then be done by running multiple experiments in different settings and times. However, replications conducted by the same researchers are not sufficient because the empirical observations in support of a hypothesis mat be

[1] *Lucent* at the time the replication was done

[2] Reference added by the author

in error or biased by the original researchers. A scientific hypothesis gains increasing acceptance when independent replications conducted by different researchers arrive at the same conclusions."

cannot confirm results

give hints/lessons learned for future replications

"Further replications are needed to understand better under which conditions scenario-based reading is effective." + Campbell + Stanley quote.

Experiment B:2

Authors/reference:	**[?**] *Strathclyde*
Type:	replication
Motivation:	further evaluate hypothesis
Guidelines:	–
Materials (other tasks):	"This experiment has re-used, except as described above, the material from the original experiment, see **[?** , Tech. Report] for a full description of the materials."
Changes (more time? debrief?):	not fractional factorial design
Encouragement of further replication:	"More work is required to finally confirm this conjecture."
Dates:	not clear
Subjects (experience, country, university):	50 undergrads
Tasks & materials (measurement changed how?):	design changed, more time
Hypothesis or research questions changed? added?	No (?)
Confirms results:	Yes.
Brooks *et al.* classification [r8]:	**(alternative, improved, similar)**

"Importantly[,] this paper is not a one-off study, but is part of a large piece of work involving several other researchers investigating the same hypothesis. Multiple independent studies of the same hypothesis are essential if software engineering is going to produce empirically evaluated theories and procedures. The paper attempts to compare its results with the other studies whenever possible."

generally supportive of results

Experiment B:3

Authors/reference:	[**?**] *Linköping*
Type:	replciation
Motivation:	eager to see results, enrich area
Guidelines:	none.
Materials (other tasks):	Lab kit from UMD
Changes (more time? debrief?):	see below
Encouragement of further replication:	yes, w/ pros
Dates:	September 15 and September 19, 1995
Subjects (experience, country, university):	24 undergraduates
Tasks & materials (measurement changed how?):	more defects, -Ad Hoc
Hypothesis or research questions changed? added?	not really (just -Ad Hoc)
Confirms results:	No.
Brooks *et al.* classification [r8]:	**(improved, improved, similar)**

"Replicated studies as ours and others [**?**] are sometimes disregarded but are critical contributions since it may be possible to further validate findings. The originators [**?**][3] observed that their Scenario method was 35% more effective than Ad Hoc and Checklist methods, and we were eager to see if we could replicate the same behaviour in our environment. As scientists[,] we are also motivated by the fact that a replicated study makes more data from the experiment and its instrumentation available, which focus and enrich the discussion in the area of inspection methods."

"We have reported a replication of a controlled experiment using the same instruments but in another educational culture."

"Of course, the most interesting question is what happens when the experiment is replicated using professional subjects. We hope that our work will demonstrate the feasability and utility of replicated experiments to professional organisations in order to obtain the necessary interest and resources."

"For educational purposes it would be great if there were a number of 'classical' experiments that could be easily replicated and modified." (regarding letting master students experiment).

Ext. materials are there.

+more specification defects. -design and analysis different -undergrads: less experience (cf Italy)

[3] Reference addded by the author

Experiment B:4

Authors/reference:	**[?**] *Lucent*
Type:	replication
Motivation:	improve external validity, check if industy experiments are needed?
Guidelines:	none?
Materials (other tasks):	same experimenters
Changes (more time? debrief?):	–
Encouragement of further replication:	not really
Dates:	1996?
Subjects (experience, country, university):	18 professionals at Lucent
Tasks & materials (measurement changed how?):	no
Hypothesis or research questions changed? added?	no
Confirms results:	Yes.
Brooks *et al.* classification [r8]:	**(similar, similar, alternative)**

"Initially we use students rather than professional[s] because cost considerations severely limit our opportunities to conduct studies with professional developers. Therefore we prefer to refine our experimental designs and measurement strategies in the university before using them in industry. This approach also allows us to do a kind of bulk screening of our research hypotheses. That is, we can conduct several studies in the university, but only rerun the most promising ones in industry. Intuitively, we feel that hypotheses that don't hold up in the university setting are unlikely to do so in the industrial setting.

Of course, this reasoning is asymmetrical. It may or may not be true that results derived in the university apply in industry. Therefore, we still need to conduct studies with professional subjects. Consequently, to improve the external validity of our initial results[,] we have replicated the experiment using professional software developers as subjects. We have also compared the performances of the student and professional populations to better understand how generalizable the original results were. This is important because experiments using professional subjects are far more costly than those using student subjects."

"To address this concern[,] we reran the experiment using software development professionals as subjects. One of our major findings is that, although the performances of the student and professional populations were different, all of the hypothesis tests gave the same results. This doesn't imply that studies with professional[s] are no longer needed, but it suggests that student studies shouldn't automatically be discounted. This is very important because studies with professionals are much more expensive than are studies with student subjects."

"These results also call into question the common practice of disregarding studies done with student subjects. The far more important question is clearly[:] when do student subjects provide an adequate model of the professional population."

Some materials appended.

Series C

Experiment C:0

Authors/reference:	[**?**] *NASA*
Type:	original
Motivation:	address threats to external validity.
Guidelines:	Campbell and Stanley [r9]
Materials (other tasks):	Published, well known!
Changes (more time? debrief?):	–
Encouragement of further replication:	already in abstract and at the end
Dates:	1995
Subjects (experience, country, university):	14 (13) professionals from NASA Goddard Space Flight Center (12 in pilot)
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"Throughout the pilot study and the 1995 run we realized that there are some threats to validity. For us it was important to describe and address all of them in detail so that other researchers benefits from the lessons we have learned and can try to avoid the threats while replicating this experiment or developing another one. Some threats have their origin in the fact that this was not an experiment with students but with professionals from industry. Some of the threats might even only be addressed through replication."

OJ! at använda proffs inte alltid bra! Kolla varför. Jf. Simula-artikeln conducting realistic experiments.

Really two runs, one described as a "pilot study for our experimental design."

Experiment C:1

Authors/reference:	**[?**] *Kaiserslautern*
Type:	replication
Motivation:	increase confidence
Guidelines:	Judd *et al.* [r28] – Brooks *et al.* [r7]
Materials (other tasks):	lab package, only 1 part used
Changes (more time? debrief?):	see below
Encouragement of further replication:	yes...
Dates:	1995-1996-1997
Subjects (experience, country, university):	undergraduate students at the university of Kaiserslautern
Tasks & materials (measurement changed how?):	similar (well, not the NASA specific part)
Hypothesis or research questions changed? added?	yes(?)
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, similar, alternative)**

"The number of subjects participating in such experiments is often low, especially if experiments are conducted in an industrial setting. Thus, it is necessary to perform replications of the original experiment in similar or slightly different experimental settings to increase the confidence in the original findings. Furthermore, replication helps generalize the results, especially when they are conducted in different contexts. We distinguish two forms of replication: internal and external replication [r7]. Internal replication is undertaken by the original experimenters; external by independent researchers. Brooks *et al.*[r7] state that external replication is critical for establishing sound results and that it provides either supporting evidence or questions the validity of the original experiment."

"With respect to experimentation, we found the GQM approach a useful vehicle to document our hypothesis and analyses. We encourage other researchers to use it as it makes replication much easier. We will work on improving the maintainability and reusability of GQM plans through formalization. We intend to extend the original lab package and make a more detaield version of this paper available as a report of the International Software Engineering Research Network (ISERN) to facilitate further replication."

Asked direct questions to Forrest Shull

Actually two runs!

GQM = Goal/Question/Metric approach.

Experiment C:2

Authors/reference:	[**?** , chapter 10] *Trondheim*
Type:	replication
Motivation:	new approach
Guidelines:	none mentioned
Materials (other tasks):	lab package from UMD, experimenter spent time there
Changes (more time? debrief?):	materials
Encouragement of further replication:	yes, with pros
Dates:	April 1996
Subjects (experience, country, university):	51 graduate students
Tasks & materials (measurement changed how?):	changed
Hypothesis or research questions changed? added?	different
Confirms results:	no
Brooks *et al.* classification [r8]:	**(alternative, improved, alternative)**

"To continue the investigation of process conformance, a replicated experiment should be carried out in a different setting using subjects that are more equal in terms of individual capabilities and motivation."

"Appendix C Experimental Material includes all documents, forms and slides that were used in the experiment. Together with the information in Chapter 10, this should be sufficient to enable a replication of the experiment. Some of the material is written in Norwegian."

"Thus, in order to validate the work presented here, parts of the experiment on Perspective-Based Reading (PBR) were replicated. There are various reasons for this—an experiment that studied a different process could have been chosen instead, but since we already had some knowledge and a high interest in PBR, this was the choice. The availability of existing material necessary to carry out the experiment also was in favor of this decision."

"The way of applying the approach should also be considered. Through modifications of the deviation vectors, arbitrary processes may be combined and compared. However, if the processes are different, it is questionable whether such operations have any meaning. Currently, the expected application is to compare individual process conformance measurements that are obtained from replicated executions of the same expected process, as typically seen in experiments."

" As has been seen in other experiments using students as subjects [**?**][4], it is difficult to achieve the same results as in replications using subjects from another population such as professional developers. Since these threats are due to the entire context of the experiment, there was little that could be done to reduce their potential effect. The alternative would have been not to do any experimental validation at all."

"In a replication of the experiment, these problems should be avoided as far as possible. A replication in an environment where volunteering professional developers were available would be desirable."

[4]Quoted as unpublished material

"The experimental validation presented in this thesis should be replicated in a context where professional developers could be used as subjects. It should also be carried out under controlled conditions."

"Appendix C includes all the material which was used in the experiment, i.e., slides, which are in Norwegian, forms and documents. Together with the experimental design presented in chapter 10, this material is sufficient to replicate the experiment if desired."

Not really a replication, the task was changed completely.

Experiment C:3

Authors/reference:	**[?]** *Maryland-98*
Type:	replication
Motivation:	MOTIVATION
Guidelines:	Basili *et al.* [r2] ([r28])
Materials (other tasks):	same people
Changes (more time? debrief?):	The method CBR is evaluated against is different
Encouragement of further replication:	yes, provides extensive package
Dates:	1995
Subjects (experience, country, university):	Students from the University of Maryland and professionals from NASA.
Tasks & materials (measurement changed how?):	essentially the same as C:0
Hypothesis or research questions changed? added?	Compared CBR to another method, commonly used at NASA
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, improved, alternative)**

"My approach to packaging experience has been to develop 'lab packages,' accessible via the Internet, which contain the following information:

- Motivation for the study (i.e. what features from the step 1 characterization of the environment are addressed by the technique or by the latest improvements);
- Detailed description of the technique (i.e. a description of the results from step 3, in sufficient detail that it can be critiqued by other researchers or practitioners, or even borrowed for use in different environments);
- Design of the study (i.e. a description of the environment in which the technique was executed, as selected in step 4, and the variables monitored to determine effectiveness, as chosen in step 2);
- Results of the study (i.e. details of the analysis done in step 5 and the conclusions drawn).

Such lab packages can support future work on this subject not only by the same researchers but also by independent researchers. By providing easy access to he experimental materials, I hope to encourage such independent replications. Although the contributions of replicated studies are too often minimized, such studies would allow the following benefits:

- A strengthening of the external validity of the study: By replicating studies in other environments, it becomes less likely that the conclusions of previous studies are the results of characteristics of the experimental environment or

subjects rather than the software engineering aspect being evaluated. Replicated studies help develop an understanding of the limits of experimental results.

- A strengthening of the internal validity of the study: A replicated study that allows certain variables to vary from previous experiments in carefully controlled ways provides further evidence of which aspects of the experiment interact with the software engineering aspect being evaluated. Such replicated studies give additional confidence in the experimental design."

"This question is necessary because it must be recognized that the work contained in this dissertation can be only a starting point for an investigation into building software reading techniques. By studying the effects of the technique in other environments, an understanding of the limits of the technology can be built up. Obviously, much of this effort will depend on other researchers and practitioners who continue to extend this work, but a significant contribution can be made by providing lab packages. Successful lab packages would encourage and aid replicated studies, contributing to the validity of a specific study."

"Finally, checking external validity requires assessing whether the results of studies can be generalized to the real environment in which the technique should be applied. This confidence can be gained in part by having the experiment replicated in other environments, by independent researchers. Such replications help demonstrate that the results observed are not due to features particular to a given environment."

"We provided a lab package to enable replication of this experiment. The experimental design was well-suited to such a package since we had deliberately included two customizable considerations, which we intended to increase the desirability of a replication since it could be easily tailored to be relevant to other environments [. . .]. The first was to use two separate sets of documents, to provide a baseline that could be repeated in different environments (the generics) while also providing direct feedback about PBR in a particular environment (here, the actual NASA/SEL documents). The intention is that the domain-specific documents can be varied between replications to provide experimental results that are directly applicable to a given environment. As we have seen, it also allowed us to notice differences in the way reviewers approach documents from familiar and non-familiar domains."

"This work, and the lab package created for it, has been used in several replications by other researchers. [. . .] The most directly relevant replications took place at the University of Kaiserslautern, in Germany, and the University of Trondheim, in Norway. Two replications were run at Kaiserslautern. A novel approach of these studies was that review teams actually met in order to discuss the defects; the results concerning the team coverage of the document were not simulated, as in our studies [**?**]. Since they achieve a similar result for their analysis of team performance, there is greater confidence that our method of simulation was accurate. They also undertook a statistical technique called meta-analysis to combine the results of their studies and ours. Since their results were very similar to ours for teams, individuals, and perspectives, the meta-analysis allows additional confidence in our results.

The replication undertaken at Trondheim is also interesting as it represents an experiment with a different but related version of the PBR techniques [**?**]. The same underlying abstractions of information and models of the low-level activities were used; however, the abstractions were mapped to procedures in a very different way. In order to be able to gauge the level of process conformance subjects used when asked to apply PBR, Sørumgård gave subjects a detailed technique for building the abstraction rather than allowing them to use their usual technique. Somewhat surprisingly, the result of this study was that the more specific technique did not result in increased process conformance."

"Section 2.1.9 discussed a replication of the original experiment at the University of Trondheim that attempted to use a more specific technique to accomplish these same ends [Sørumgård97]. Because that experiment did not observe the effects it had

hoped to see (viz. increased process conformance), we adopted a different strategy for producing reading techniques at a higher level of detail. The differences between our technique and theirs lie in the underlying models of the reading technique and are discussed in section 3.3 below."

"Even though this is only a draft of the finished product, we consider this to be a promising lab package in that it has already supported replications of the experiment in a number of different environments (in Philadelphia, Italy, Brazil, and Sweden). As the analyses of these experiments are published, they can be used to build hypotheses about the larger area of reading techniques, as the Kaiserslautern and Trondheim replications were used in the original PBR experiment."

Experiment C:4

Authors/reference:	**[?**] *Lund*
Type:	replication
Motivation:	different, but related question
Guidelines:	Cook and Campbell [r14], Wohlin *et al.* [r60]; none on repli. as such
Materials (other tasks):	"based on a lab package provided by the university of Maryland in order to support empirical investigations of scenario-based reading." Only 1 part used?
Changes (more time? debrief?):	
Encouragement of further replication:	Yes. both meta-analysis and new replications.
Dates:	1999? 2000?
Subjects (experience, country, university):	30 graduate students
Tasks & materials (measurement changed how?):	minimal changes
Hypothesis or research questions changed? added?	yes
Confirms results:	no
Brooks *et al.* classification [r8]:	**(alternative, similar, alternative)**

Experiment as such was close, but different questions.
Analysis of previous repl. series.
Stong arguments for meta-analysis.

Series D

Experiment D:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	135 lines of COBOL
Changes (more time? debrief?):	–
Encouragement of further replication:	–
Dates:	June 1995
Subjects (experience, country, university):	101 undergraduate students at the University of New South Wales
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment D:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	confirm, improve, further study of D:0
Guidelines:	Judd *et al.* [r28]
Materials (other tasks):	Same COBOL task
Changes (more time? debrief?):	more realistic, roles assigned
Encouragement of further replication:	yes, with professionals
Dates:	June 1996
Subjects (experience, country, university):	101 students at the University of New South Wales (different from D:O)
Tasks & materials (measurement changed how?):	Tasks and secondary material (forms) improved
Hypothesis or research questions changed? added?	improved to refelect role assignment
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved,improved,similar)**

Series E

Experiment E:0

Authors/reference:	[**?**]
Type:	original
Motivation:	Alleviate low power, increase generalizability
Guidelines:	Daly [r15], Lindsay and Ehrenberg [r36] and several others on statistics
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	none, just new people. (or: different modules? They have different names p. 396 - but they say "close" themselves on p 397 and refer to "same artefacts")
Encouragement of further replication:	Yes.
Dates:	March–July 1998
Subjects (experience, country, university):	professionals at Robert Bosch GmbH
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment E:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	Alleviate low power, increase generalizability
Guidelines:	Daly [r15], Lindsay and Ehrenberg [r36] and several others on statistics
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	none, just new people. (or: different modules? They have different names p. 396 - but they say "close" themselves on p 397 and refer to "same artefacts")
Encouragement of further replication:	Yes.
Dates:	March–July 1998
Subjects (experience, country, university):	professionals at Robert Bosch GmbH
Tasks & materials (measurement changed how?):	all had different! [see **?** , table 2, table 4]
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, similar, similar)**

Experiment E:2

Authors/reference:	[**?**]
Type:	replication
Motivation:	Alleviate low power, increase generalizability
Guidelines:	Daly [r15], Lindsay and Ehrenberg [r36] and several others on statistics
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	none, just new people. (or: different modules? They have different names p. 396 - but they say "close" themselves on p 397 and refer to "same artefacts")
Encouragement of further replication:	Yes.
Dates:	March–July 1998
Subjects (experience, country, university):	professionals at Robert Bosch GmbH
Tasks & materials (measurement changed how?):	all had different! [see **?** , table 2, table 4]
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, similar, similar)**

"Therefore, the objective is to chose a design that minimizes the threats to validity within the prevailing cost constraints."

"One possibility to tackle the problem of low power is to replicate an empirical study and merge the results of the studies using meta-analysis techniques."

"The replication of experimental studies provides a basis for confirming the results of the original experiment [[r15]]. However, replications can also be useful for generalizing results. A framework that distinguishes between close and differentiated replications has been suggested to explain the benefits in terms of generalizing results [[r36]]. Our two replications can be considered close since they were performed by the same investigators, using the same design and reading artifacts, under the same conditions, within the same organization, and during the same period of time. However, there were also some differences that facilitate the generaliztion of the results. First, the subjects were different. Therefore, we can claim that the results hold across subjects at Bosch Telecom GmbH. Second, the modules were varied. Again, if consistent results are obtained, then we can claim that they are applicable across different code modules at Bosch Telecom GmbH. By varying these two elements in the replications, one attempts to find out if the same results occur *despite* these differences [[r36]]."

"To attain such generalisations, it is necessary to replicate the current study under different conditions."

"Therefore, we encourage the external replication of this study in different environments by different researchers. A replication can take many forms, such as controlled experiments or case studies in industrial projects."

"However, replication, in general, raises the question of how to compare and combine the results of the original study and the replications. We found meta-analysis tools a useful tool for this purpose. Other researchers may consider these techniques in their arsenal of analysis approaches. This requires that researchers performing empirical research not only present results from statistical[ly] significant tests in their articles, e.g., p-values, but also compute and include the effect size and the number of subjects in their reporting."

Some (illustration) material in the article.

Series F

Experiment F:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	package (see later)
Changes (more time? debrief?):	–
Encouragement of further replication:	strongly encouraged (see below)
Dates:	spring 2001
Subjects (experience, country, university):	23 graduate students at Lund institute of technology
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"As always, when conducting experiments to increase the body of knowledge, the experiment has to the replicated in different contexts. The replications should address changes in the design, for example, use a different domain, and seed more faults into the document under inspection. The method should also be investigated in a case study in an industrial setting in order to evaluate whether it still provides positive effects. It would be especially interesting to investigate the method with professionals as subjects."

The further work includes enhancement of UBR, either to include checklist items or to investigate the time-based ranking method. Although the results are promising, the method needs to be replicated and compared with, for example, usage-based testing."

Experiment F:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	increase confidence or reject
Guidelines:	Basili *et al.* [r3], meta: Thelin and Runesson [r58], general: Wohlin *et al.* [r60], Juristo and Moreno [r30]
Materials (other tasks):	same people
Changes (more time? debrief?):	no
Encouragement of further replication:	yes, underway + package
Dates:	spring 2004 or autumn 2003
Subjects (experience, country, university):	62 graduate students (more heterogenous than F:0) from Blekinge institute of technology
Tasks & materials (measurement changed how?):	no
Hypothesis or research questions changed? added?	no (but deducts combined results)
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, similar, alternative)**

"It is important to perform new experiments, but also to conduct replications[r3], meta-analyses[r40, r46], and to design methods to compare series of experiment[s] [r58]. Replications of experiments create a body of knowledge, which aids the insight into software engineering practice. The objective of this paper is to contribute to the improvement of reading techniques by replicating an experiment. The purpose of the replication is to increase the confidence or to reject the result of the original experiment.

"Replications of experiments are important in order to understand more about software engineering methods and techniques. In order to facilitate replications using this experimental package, the material is published on `http://serg.telecom.lth.se/research/packages/`"

Experiment F:2

Authors/reference:	**[?]**
Type:	replication
Motivation:	important, esp ext (see below)
Guidelines:	General: Brooks *et al.* [r7], Juristo and Moreno [r30], Wohlin *et al.* [r60] **[? ?]** On ext. replication: Basili *et al.* [r3] **[?]** On metaanalysis:Hayes [r23]
Materials (other tasks):	package is public (not mentionned)
Changes (more time? debrief?):	UBR-ir introduced
Encouragement of further replication:	yes, in general terms but not specifically
Dates:	december 2003
Subjects (experience, country, university):	131, mixed undergraduates and graduates (from the Tech. Univ. of Vienna or the Univ. of Vienna?)
Tasks & materials (measurement changed how?):	a little (UBR-ir) but data kept separate
Hypothesis or research questions changed? added?	addition
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, improved, improved)**

"While an individual empirical study can provide evidence in a specific context, the generalization of empirical findings should be supported with a well-planned familiy of empirical studies, e.g. replicated experiments [r7, r30, r60] **[? ?]**. Especially external replications are important, where the experimenters are different from the researchers, who proposed the method [r3]**[?]**."

"Replication of experiments requires careful planning and preparation to achieve repeatable and comparable results."

Series G

Experiments G:0

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	not expressed clearly. Statistical power?
Guidelines:	None explicitly on replication. Conte *et al.* [r13]
Materials (other tasks):	more complex in J:1, slightly different in J:2
Changes (more time? debrief?):	materials changed
Encouragement of further replication:	internal
Dates:	summer 1991, autumn 1992
Subjects (experience, country, university):	11 graduates and undergradutes from Ohio State University
Tasks & materials (measurement changed how?):	yes
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, similar, similar)**

Experiments G:1

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	not expressed clearly. Statistical power?
Guidelines:	None explicitly on replication. Conte *et al.* [r13]
Materials (other tasks):	more complex in J:1, slightly different in J:2
Changes (more time? debrief?):	materials changed
Encouragement of further replication:	internal
Dates:	summer 1991, autumn 1992
Subjects (experience, country, university):	11 graduates and undergraduates from Ohio State University
Tasks & materials (measurement changed how?):	yes
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, similar, similar)**

Experiments G:2

Authors/reference:	**[?]**
Type:	original and replication
Motivation:	not expressed clearly. Statistical power?
Guidelines:	None explicitly on replication. Conte *et al.* [r13]
Materials (other tasks):	more complex in J:1, slightly different in J:2
Changes (more time? debrief?):	materials changed
Encouragement of further replication:	internal
Dates:	summer 1991, autumn 1992
Subjects (experience, country, university):	11 graduates and undergradutes
Tasks & materials (measurement changed how?):	yes
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved, similar, similar)**

"The second experiment was a replication of the first experiment using a more complex component that encapsulated a 'partial map'."

"The subjects used in our experiments, while mature students many of whom had full-time jobs involving software development, might not be representative of the typical programmer. Generally, they had only a couple of years' experience in commercial software development. Subjects with different backgrounds might perform differently on our experimental tasks; this is a potential avenue for future research."

"We are planning furter experiments to more carefully analyze the defects made in the development of components such as those studied herein. It is important not only to characterize the kinds of defects observed, but also to provide if possible some cognitive explanation of these observations. We are also planning other studies to try and replicate the results reported herein. Studies such as these serve to provide a more sound and scientific basis for using (or not using) various software engineering methods."

Very little mention of replication as a concept.

Series H

Experiment H:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	–
Changes (more time? debrief?):	–
Encouragement of further replication:	see H:1
Dates:	unknown. 1995?
Subjects (experience, country, university):	31 students (from the University of Strathclyde?)
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment H:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	not explicit
Guidelines:	[r16]
Materials (other tasks):	one task not used as subjects were supposed to participate in G:0
Changes (more time? debrief?):	more experienced students, one task skipped
Encouragement of further replication:	yes, multi-method & series I
Dates:	unknown. 1995?
Subjects (experience, country, university):	29 more experienced students (from the U. of Strathclyde?)
Tasks & materials (measurement changed how?):	one task removed
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, alternative, improved)**

Part of a multi-method programme of research that also includes other types of studies.

Series I

Experiment I:0

Authors/reference:	[**?**]
Type:	original
Motivation:	improvement of experiment
Guidelines:	[r16]
Materials (other tasks):	more inheritance depth that H
Changes (more time? debrief?):	–
Encouragement of further replication:	yes, with professionals
Dates:	Not given. 1995?
Subjects (experience, country, university):	31 postgraduates, partly same as H:1, from the University of Strathclyde
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment I:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	see quotes below. increase confidence in findings
Guidelines:	Daly [r15]
Materials (other tasks):	freely available from [**?**]
Changes (more time? debrief?):	no
Encouragement of further replication:	in general, yes
Dates:	unclear. 1996-1997-1998?
Subjects (experience, country, university):	10 undergraduates from Bournemouth University
Tasks & materials (measurement changed how?):	no
Hypothesis or research questions changed? added?	no
Confirms results:	no
Brooks *et al.* classification [r8]:	**(similar, similar, similar)**

"This section describes an experiment conducted at Bornemouth university to independently replicate the work of the Strathclyde research team [**?**][5] To this end, we are indebted to them for making experimental materials and procedures freely available."

[5]Reference added by the author.

"There seem to be two lessons to be learnt. Firstly, one needs to be cautious about uncritically accepting the findings of single experiments, especially where small numbers of student subjects are employed. Replication is important since this allows us to have a far greater degree of confidence in the findings."

"Another problem with experiments relates to the scale and plausibility of the materials. Obviously[,] this is not addressed by faithful replication."

"Thus, there is a pressing need for further empirical research utilising more subjects and dealing with industrial scale tasks."

Experiment I:2

Authors/reference:	[**?**]
Type:	replication
Motivation:	not specified
Guidelines:	no, neither on experimenting nor on replication
Materials (other tasks):	not clear
Changes (more time? debrief?):	see below
Encouragement of further replication:	yes.. see text
Dates:	not clear. 1998-1999?
Subjects (experience, country, university):	48 undergraduates from the University of Southhampton
Tasks & materials (measurement changed how?):	yes—added understandability, modifications in leaf nodes
Hypothesis or research questions changed? added?	some (understandability)
Confirms results:	no
Brooks *et al.* classification [r8]:	**(improved, alternative, similar)**

Materials available as a lab package.

"In this paper, we discuss the results of an experiment which we carried out based on that carried out by Daly *et al.* [**?**], which investigated the modifiability of C++ programs with zero, three and five levels of inheritance and suggested that there was an optimum level of inheritance lying between three and five levels. The experiment described in this paper differed in serveral respects. [. . .]"

"Ideally, this empirical research should be carried out on as many industrial-sized systems as possible, (with subjects of varying experience), supported by well-designed hypotheses. Industrial-strength tools need to be provided to aid the speedy collection of data and dissemination of results. To encourage replication of the experiment contained in this paper by other researchers, experimental materials are publicly available at `http://www.ecs.soton.ak.uk/~rnv95r`"

Series J

Experiment J:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	Kaplan and Duchon [r32], Daly *et al.* [r17], Judd *et al.* [r28], Rosnow and Rosenthal [r48]
Materials (other tasks):	Yes, offered and tips (see below)
Changes (more time? debrief?):	–
Encouragement of further replication:	somehow
Dates:	1997
Subjects (experience, country, university):	undergraduates at the University of Kaiserslautern, number unknown
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"It is also important to point out that weaknesses imposed by these two threats can be addresed if similar results can be obtained by using different empirical techniques—the idea is that the weaknesses of one study can be addressed by the strengths of another; see, e.g., [r17, r32]."

"Two, the various threats to external validity limit our ability to generalise the results—in this instance, we plan to use the results of this student[-]based experiment to facilitate further investigation."

"In software engineering, to answer the type of question we are addressing here, we usually expect to have [to] work with small sample sizes—it is common to work with a sample of convenience, e.g., students in a programming class or with professional programmers during a training session. [. . .] Performing power analysis as well as external replications are necessary to achieve significant, reliable and generalisable results. In addition, it is likely that consistent data will have to be collected from different studies and integrated to allow meta-analyses to be performed [r28, r48] [. . .]"

"To be plausible, collaboration between different research groups is necessary, an objective of research networks such as ISERN (International Software Engineering Network)."

Experiment J:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	further investigation
Guidelines:	Brooks *et al.* [r8], Wood *et al.* [r61], Kaplan and Duchon [r32]
Materials (other tasks):	same researchers
Changes (more time? debrief?):	different tasks and more subjects
Encouragement of further replication:	Yes
Dates:	1997?
Subjects (experience, country, university):	33 graduate students, University of Kaiserslauterns
Tasks & materials (measurement changed how?):	yes, to "further isolate the effects being investigated."
Hypothesis or research questions changed? added?	subset
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, improved, improved)**

"The investigation utilizes and improves upon the object-oriented materials from the original study ([. . .]). In addition, the hypotheses of this study are a subset of the hypotheses from the original study ([. . .]). Consequently, using the framework of Brooks *et al.* as a reference [r8], the study can be classified as an internal replication—that is, a replication conducted by the same set of researchers that performed the original study. The replication framework of Brooks *et al.* [r8] provides a classification scheme for replications along three different dimensions of an experiment **(method, tasks, subjects)**. Accordingly, we would classify this internal replication as **(similar, improved, improved)**. **Method** is similar because it is the same method used in the original study. **Tasks** are classified [as] improved because they were modified to test the hypotheses more thoroughly, [. . .]. **Subjects** are also classified as improved because, although the same subject pool was used, the number of subjects was far greater and the debriefing questionnaire elicited more detailed information."

"Laboratory settings such as this one allow the investigation of a larger number of hypotheses at a lower cost than field studies. The hypotheses that seem to be supported in the laboratory setting can then be tested further in more realistic industrial settings with a better chance of discovering important and interesting findings. Conversely, laboratory experiments can be used to confirm results obtained in field studies, where control and, therefore, internal validity is usually weaker."

"A replication package is available for researchers interested in externally replicating our experiment. Improvements to the experimental procedure might include increasing the task time and improving the time data collection procedures."

Replication package is available, and the article provides hints on what to improve.

Series K

Experiment K:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	–
Changes (more time? debrief?):	–
Encouragement of further replication:	no
Dates:	not given. 1998?
Subjects (experience, country, university):	69 post-graduate students at Université de Paris I—Panthéon Sorbonne
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment K:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	not convinced by results in K:0
Guidelines:	[r47]
Materials (other tasks):	obtained from K:O in the CREWS project
Changes (more time? debrief?):	some modified guidelines
Encouragement of further replication:	not this experiment in particular
Dates:	not given. 2000?
Subjects (experience, country, university):	14 post-graduate students at Bournemouth University
Tasks & materials (measurement changed how?):	improved guidelines
Hypothesis or research questions changed? added?	unchanged
Confirms results:	yes and no
Brooks *et al.* classification [r8]:	**(improved, similar, similar)**

"When attempting to replicate an experiment, it would be ideal to perform an 'exact replica- tion' of that experiment. The goal would be to get the same results as the original experiment from similar conditions or, indeed, to prove the null hypothesis. This is not very easy to do in most software engineering experiments because different people will be involved who have different degrees of knowledge and interpretation

and apply them in different ways. Although 'no replication is ever exact' (Robson, 1993), as close a replication as possible is the next best alternative."

"It is unclear which approach the CREWS experiment took but insufficient information has been given to allow close replication. A number of points interested us about the CREWS experiment. First, we were not convinced by all the CREWS guidelines. Second, we were not convinced about the validity of their hypotheses. Third, we were not convinced that application of the guidelines could produce better use-case descriptions than application of common sense because many of the guidelines appear to be constructs in the English language that are used every day."

"There is a replication issue to consider regarding this study. Since the CREWS experimental procedures are not completely documented, it is not justified to call our experiment an exact replication. However, it has proven a useful exercise nonetheless to implement the guidelines and see what difference they make to use-case descriptions."

"The body of work produced by CREWS is very significant and we welcome this research. The guidelines, as part of that research, are very interesting and we recommend that the guidelines should be considered when authoring use cases, especially aspects of the Style Guidelines. However, our results found, especially with regards the Content Guidelines, only the number of times a guideline was correctly implemented. As such, we are unclear how some of the guidelines necessarily improve use-case descriptions. We think it important that further studies be carried out to implement the CREWS guidelines. "

Series L

Experiment L:0

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	Increase confidence in findings
Guidelines:	Cook and Campbell [r14]
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	Different subjects (but same time, randomised)
Encouragement of further replication:	Different application but same domain. +did a pilot
Dates:	yes, esp. with professionals.
Subjects (experience, country, university):	1997-1998?
71 undergraduates (for both experiment instances) (University of Dayton, Ohio?) Tasks & materials (measurement changed how?):	another task.
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, alternative, similar)**

Experiment L:1

Authors/reference:	**[?]**
Type:	original and replication
Motivation:	Increase confidence in findings
Guidelines:	Cook and Campbell [r14]
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	Different subjects (but same time, randomised)
Encouragement of further replication:	Different application but same domain. +did a pilot
Dates:	yes, esp. with professionals.
Subjects (experience, country, university):	1997-1998?
71 undergraduates (for both experiment instances) (University of Dayton, Ohio?) Tasks & materials (measurement changed how?):	another task.
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, alternative, similar)**

"Furthermore, to increase confidence in the findings, the main experiment was replicated using another task."

"Replication helped addess the criticism of low generalizability leveled against experiments and contributed to the external validity of the study."

All material in the article—possible to replicate quasi-close just from the article.

There was a pilot w/ 18 students also.

Series M

Experiment M:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	Java task
Changes (more time? debrief?):	–
Encouragement of further replication:	–
Dates:	January 1997
Subjects (experience, country, university):	74 German graduate students at the University of Karlsruhe
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment M:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	complement results
Guidelines:	no explicit reference
Materials (other tasks):	C++ task
Changes (more time? debrief?):	different task, less experienced subjects
Encouragement of further replication:	no
Dates:	May 1997
Subjects (experience, country, university):	22 American undergraduates at Washington University St. Louis
Tasks & materials (measurement changed how?):	different task (other programming language)
Hypothesis or research questions changed? added?	same
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar,alternative,alternative)**

Series N

Experiment N:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	–
Changes (more time? debrief?):	–
Encouragement of further replication:	need more metrics, replication a possibility
Dates:	unclear. 2001 or 2002?
Subjects (experience, country, university):	70 undergraduate students at the university of Castilla–La Mancha
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"Several experts [have] suggested the necessity of a public repository of measurement experiences, which we think would be a good step towards the success of all the work done on software measurement."

Experiment N:1

Authors/reference:	[**?**]
Type:	replication, internal
Motivation:	improve some issues not covered
Guidelines:	None on replication as such. Wohlin *et al.* [r60]
Materials (other tasks):	same people
Changes (more time? debrief?):	none
Encouragement of further replication:	no
Dates:	unclear. 2002?
Subjects (experience, country, university):	28 undergraduates from another university, the university of Seville
Tasks & materials (measurement changed how?):	yes
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, improved, alternative)**

"We decided to carry out this experiment trying to improve some issues not covered in the previous one[.]"

"The experiment was replicated with [an]other group of twenty eight students. They were also undergraduate students [. . .] Therefore, the characteristics of the subjects were similar."

It's unclear whether the materials were changed between the original experiment and the replication or between previous experiments and the original in this series. The text tends to suggest the latter.

Series O

Experiment O:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	C++ design pattern tasks
Changes (more time? debrief?):	–
Encouragement of further replication:	not explicitly encouraged
Dates:	November 1997
Subjects (experience, country, university):	29 professionals with 4 years' experience, all from one company, Munich, Germany
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment O:1

Authors/reference:	**[?]**
Type:	replication
Motivation:	improve realism
Guidelines:	[r54, r34], [r36]
Materials (other tasks):	adapted from O:0
Changes (more time? debrief?):	real tasks, not paper-based, performed at a computer
Encouragement of further replication:	FURTHER
Dates:	DATES
Subjects (experience, country, university):	44 professional subjects (paid), Oslo, Norway
Tasks & materials (measurement changed how?):	Real tasks performed at a computer, not paper-based
Hypothesis or research questions changed? added?	same, but much more data collected
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(improved,alternative,improved)**

Series P

Experiment P:0

Authors/reference:	**[?]**
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	–
Changes (more time? debrief?):	–
Encouragement of further replication:	calls for more research but not replication in particular.
Dates:	not given. 1984?
Subjects (experience, country, university):	16 'experienced programmers' (professionals and advanced students)
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

There are actually two runs in this experiment; statistical replication.

Experiment P:1

Authors/reference:	[**?**]
Type:	replication
Motivation:	replication as such, experiment suitable for replication
Guidelines:	Wrote their own: [r7]
Materials (other tasks):	same
Changes (more time? debrief?):	resisted temptation to change in the interest of replication
Encouragement of further replication:	not of this particular research
Dates:	not given. 1993?
Subjects (experience, country, university):	23 (17 successful), various levels of students and researchers
Tasks & materials (measurement changed how?):	same
Hypothesis or research questions changed? added?	inductive analysis
Confirms results:	no
Brooks *et al.* classification [r8]:	**(similar, improved, similar)**

"[P:0] qualified as well-performed and experimentally based."

Series Q

Experiments Q:0

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	"[. . .] Experiment III served to verify the repeatability of the experiment in a production environment and to make a first comparison between the results obtainable in a[n] academic or a production milieu."
Guidelines:	no reference
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	more complex task in 1 and 2, professionals in 2. (obs: orig = 0)
Encouragement of further replication:	No mention
Dates:	1996-1997
Subjects (experience, country, university):	undergraduates
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment 1 almost a pre-test "Experiment I served to hone the tools used".

Experiments Q:1

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	"[. . .] Experiment III served to verify the repeatability of the experiment in a production environment and to make a first comparison between the results obtainable in a[n] academic or a production milieu."
Guidelines:	no reference
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	more complex task in 1 and 2, professionals in 2. (obs: orig = 0)
Encouragement of further replication:	No mention
Dates:	1996-1997
Subjects (experience, country, university):	undergraduates
Tasks & materials (measurement changed how?):	different tasks
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar,improved,similar)**

Experiments Q:2

Authors/reference:	[**?**]
Type:	original and replication
Motivation:	"[. . .] Experiment III served to verify the repeatability of the experiment in a production environment and to make a first comparison between the results obtainable in a[n] academic or a production milieu."
Guidelines:	no reference
Materials (other tasks):	Same experimenters
Changes (more time? debrief?):	more complex task in 1 and 2, professionals in 2. (obs: orig = 0)
Encouragement of further replication:	No mention
Dates:	1996-1997
Subjects (experience, country, university):	professionals
Tasks & materials (measurement changed how?):	same tasks as E:1
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar,improved,improved)**

Material not in article and not 'offered'.

Series R

Experiment R:0

Authors/reference:	**[?]**
Type:	original
Motivation:	[r2]
Guidelines:	–
Materials (other tasks):	LabView emulation in SuperCard
Changes (more time? debrief?):	–
Encouragement of further replication:	not mentioned
Dates:	not stated. 1990?
Subjects (experience, country, university):	5 non-professionals
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

Experiment R:1

Authors/reference:	**[?]**
Type:	replication
Motivation:	interest in the field, replication as such, small number of subjects in S:0
Guidelines:	none given, [r2]
Materials (other tasks):	unobtainable, reconstructed from pieces
Changes (more time? debrief?):	very similar overall
Encouragement of further replication:	no outright mention
Dates:	not stated. 1996
Subjects (experience, country, university):	nine subjects
Tasks & materials (measurement changed how?):	reconstructed but largely identical
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, similar, similar)**

"The small number of subjects is one of the reasons that a replication was important."

"The goals of our investigation are threefold. First, we replicated the study of Green et al.. Thus, we based our experimental design on theirs."

"The subjects in this experiment were all programmers with more than three years of programming experience. None had more than cursory experience with LabVIEW programming. The experimental method used was the same as that of the experiment of Green et al. The 16 screens used as stimuli were identical to those used in that previous experiment. We were unable to obtain a working version of the original SuperCard driving program, but were able to reconstruct it from pieces. The primary differences from this experiment and the study of Green et al. were these: [. . .]"

"The experimental design [and method] of the replication experiment was identical to that of the experiment of Green et al.."

"The validity of our replication may be questioned because our subjects lack of experience with LabVIEW programming, where subject in Green's experiment had at least six months' experience."

The article also contains a report of a further, new experiment.

Series S

Experiment S:0

Authors/reference:	[**?**]
Type:	original
Motivation:	experiments have problems
Guidelines:	Pfleeger [r45], Bourque and Côte [r5]
Materials (other tasks):	Available on-line
Changes (more time? debrief?):	–
Encouragement of further replication:	yes, see below
Dates:	unknown. 1998-1999?
Subjects (experience, country, university):	59 undergraduate students
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"However, the controlled experiments have problems (like the large number of variables that causes differences, dealing with low level issues, microcosms of reality and small set of variables) and limits (do not scale up, are done in class training situations, are made in vitro and face a variety of threats of validity). Therefore, it may be more convenient to run multiple studies, mixing controlled experiments with case studies. For these reasons, a more deep empirical evaluation is under way in collaboration with industrial and public organisations in "real-life" situations.

Experiment S:1

Authors/reference:	**[?]**
Type:	replication
Motivation:	make results of experiment more reliable
Guidelines:	Wohlin *et al.* [r60], Basili *et al.* [r3]
Materials (other tasks):	same people
Changes (more time? debrief?):	measurement changed
Encouragement of further replication:	yes
Dates:	not clear. 1999-2000?
Subjects (experience, country, university):	11 professionals from Cronos S.A:
Tasks & materials (measurement changed how?):	no
Hypothesis or research questions changed? added?	no
Confirms results:	unclear
Brooks *et al.* classification [r8]:	**(improved, similar, improved)**

"Prior to this study, we conducted another controlled experiment with the aim of proving [. . .]

As the controlled experiment presented later in this paper is the replica of this earlier one, most of their characteristics are are similar. [. . .] [W]here necessary, we will point out specific differences between the original and the replica."

There was also a case study in this experiment family.

"As previously indicated, the main goal of this paper is to explain the replica of the previous controlled experiment. The hypothesis did not vary in the replication of the experiment. However, we did change the way in which the dependent variable was measured (we wanted to capture the analyzability in another way to confirm if the previous results were independent of the way the analyzability was captured) and of the subjects (due to the limitations related to the experiments performed by students). By carrying out this kind of replication in which the same hypothesis is studied, but some details of the experiment are changed, our aim is to make the results of the experiment more reliable.

Experiment goals defined in GQM: "*To analyze* the metrics for relational databases *for the purpose* of evaluating if they can be used as a useful mechanisms *with respect of* the relational databases analyzability from the designer *point of view in the context of* professionals in relational databases."

"Replication of the experiments is also necessary because with the isolated results of one experiment only, it is difficult to appreciate how widely applicable the results are, and, thus, to assess to what extent they really contribute to the field [r3]. In this paper, the complete replica of an experiment conducted with the two metrics presented is explained in detail."

"Performing empirical validation with the metrics is fundamental in order to demonstrate their practical utility. In this line, we have summarized two previous empirical studies made with metrics for relational databases: a controlled experiment and a case study."

Series T

Experiment T:0

Authors/reference:	[**?**]
Type:	original
Motivation:	–
Guidelines:	–
Materials (other tasks):	–
Changes (more time? debrief?):	–
Encouragement of further replication:	yes, buth the experiment needs to be evolved.
Dates:	not given. 1999?
Subjects (experience, country, university):	12 graduate students from the university of Kaiserslautern
Tasks & materials (measurement changed how?):	–
Hypothesis or research questions changed? added?	–
Confirms results:	–
Brooks *et al.* classification [r8]:	**(–)**

"In any case, the point should be emphasised that the presented research at its current stage is exploratory of nature and just the first step of a series of experiments, which—after modification of the treatments and stepwise inclusion of subjects with different backgrounds—might yield more generalisable results in the future."

"A closer look at the nature of the applied treatments also proposes an improved experimental design for future replications."

Experiment T:1

Authors/reference:	**[?]**
Type:	replication, internal
Motivation:	not given explicitly
Guidelines:	none (only on statistics in general
Materials (other tasks):	same people
Changes (more time? debrief?):	more time
Encouragement of further replication:	yes
Dates:	2001?
Subjects (experience, country, university):	10 graduate and 2 postgraduate students from the university of Oulu
Tasks & materials (measurement changed how?):	more time
Hypothesis or research questions changed? added?	no
Confirms results:	yes
Brooks *et al.* classification [r8]:	**(similar, similar, similar)**

"This paper presents the results of a controlled experiment and its first external replication [. . .] While the experiment was originally performed at the university of Kaiserslautern, Germany **[?]** its replication took place at the University of Oulu,Finland."

"In any case, the point should be emphasized that the presented research at its current stage is exploratory of nature and just the first step of a series of experiments, which—after modification of the treatments and stepwise inclusion of subjects with different backgrounds—might yield more generalisable results in the future."

"Although the results of the two studies are promising, further replication is required for two reasons. First, a single study even if replicated only provides a starting point for investigation. In this case, the studies were exploratory in nature. Based on the presented results, a further replication should consider the examination of cause/effect relationships. And second, each empirical study exhibits specific threats to validity, which can only be ruled out by replication. Additional replications of this study are currently planned."

Misinterpretation of 'external replication'? Same experimenters, even if in a different setting?

Appendix B

Search results

These reference lists are automatically generated and thus have a poorer quality that the main reference lists of this thesis.

Search results for "software engineering" and replication

[1] J. Miller. Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information and Software Technology*, 47(4):233–244, 2005.

[2] James Miller. Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information and Software Technology*, 47(4):233–244, 2005.

[3] Andrew Taylor. An operations perspective on strategic alliance success factors: An exploratory study of alliance managers in the software industry. *International Journal of Operations & Production Management*, 25(5):469–490, 2005.

[4] Forrest Shull, Manoel G. Mendoncca, Victor Basili, Jeffrey Carver, José C. Maldonado, Sandra Fabbri, Guilherme Horta Travassos, and Maria Cristina Ferreira. Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9(1-2):111–137, 2004.

[5] Fubo Zhang and Erik H. D'Hollander. Using hammock graphs to structure programs. *IEEE Transactions on Software Engineering*, 30(4):231–245, 2004.

[6] J. Miller. Statistical significance testing–a panacea for software technology experiments. *Journal of Systems and Software*, 73(2):183–192, 2004.

[7] Marek Vokác, Walter Tichy, Dag I. K. SjØberg, Erik Arisholm, and Magne Aldrin. A controlled experiment comparing the maintainability of programs designed with and without design patterns-a replication in a real programming environment. *Empirical Software Engineering*, 9(3):149–195, 2004.

[8] M. Genero, M. Piatini, and E. Manso. Finding "early" indicators of uml class diagrams understandability and modifiability. *Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on*, pages 207–216, 2004.

[9] D. Winkler, M. Halling, and S. Biffl. Investigating the effect of expert ranking of use cases for design inspection. *Euromicro Conference, 2004. Proceedings. 30th*, pages 362–371, 2004.

[10] Chen-Liang Fang, Deron Liang, and Fengyi Lin. A nested invocation suppression mechanism for cctive replicated soap systems. *Software Engineering Conference, 2004. 11th Asia-Pacific*, pages 392–399, 2004.

[11] Uzair Ahmad, Mohammad Waseem Hassan, Arshad Ali, Richard McClatchey, and Ian Willers. An integrated approach for extraction of objects from xml and transformation to heterogeneous object oriented databases. Technical report, arXiv, 2004.

[12] F. Zhang and E.H. D'Hollander. Using hammock graphs to structure programs. *IEEE Transactions on Software Engineering*, 30(4):231–245, 2004.

[13] R. Smeikal and K.M. Goeschka. Fault-tolerance in a distributed management system: a case study. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 478–483, 2003.

[14] P. Krause, B. Freimut, and W. Suryn. New directions in measurement for software quality control. In *Software Technology and Engineering Practice, 2002. STEP 2002. Proceedings. 10th International Workshop on*, pages 129–143, 2003.

[15] Dietmar Pfahl, Oliver Laitenberger, Jorg Dorsch, and Günther Ruhe. An externally replicated experiment for evaluating the learning effectiveness of using simulations in software project management education. *Empirical Software Engineering*, 8(4):367–395, 2003.

[16] E. Mendes, N. Mosley, and S. Counsell. A replicated assessment of the use of adaptation rules to improve web cost estimation. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 100–109, 2003.

[17] Chen Shiping and I. Gorton. A predictive performance model to evaluate the contention cost in application servers. In *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pages 435–440, 2002.

[18] Deron Liang, Chen-Liang Fang, Chyouhwa Chen, and Fengyi Lin. A nested invocation suppression mechanism for active replication fault-tolerant corba. In *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pages 117–125, 2002.

[19] S. Frolund and R. Guerraoui. e-transactions: end-to-end reliability for three-tier architectures. *IEEE Transactions on Software Engineering*, 28(4):378–395, 2002.

[20] D. Rodriguez, R. Harrison, M. Satpathy, and J. Dolado. An investigation of prediction models for project management. In *Computer Software and Applications Conference, 2002. COMPSAC 2002. Proceedings. 26th Annual International*, pages 779–784, 2002.

[21] Deron Liang, Chen-Liang Fang, Chyouhwa Chen, and Fengyi Lin. A nested invocation suppression mechanism for active replication fault-tolerant corba. In *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pages 117–125, 2002.

[22] Oliver Laitenberger. Cost-effective detection of software defects through perspective-based inspections. *Empirical Software Engineering*, 6(1):81–84, 2001.

[23] Lionel C. Briand, Jürgen Wüst, and Hakim Lounis. Replicated case studies for investigating quality factors in object-oriented designs. *Empirical Software Engineering*, 6(1):11–58, 2001.

[24] Bjorn Regnell, Per Runeson, and Thomas Thelin. Are the perspectives really different? - further experimentation on scenario-based reading of requirements. *Empirical Software Engineering*, 5(4):331–356, 2000.

[25] Litoiu M., Rolia J., and Serazzi G. Designing process replication and activation: a quantitative approach. *IEEE Transactions on Software Engineering*, 26(12):1168–1178, 2000.

[26] Litoiu Marin, Rolia Jerome, and Serazzi Giuseppe. Designing process replication and activation: A quantitative approach. *IEEE Transactions on Software Engineering*, 26(12):1168–1178, 2000.

[27] Polze A., Schwarz J., Wehner K., and Sha L. Integration of corba services with a dynamic real-time architecture. In *Real-Time Technology and Applicationd Symposium, 2000. RTAS 2000. Proceedings. Sixth IEEE*, pages 198–206, 2000.

[28] Anderson G.E., Graham T.C.N., and Wright T.N. Dragonfly: linking conceptual and implementation architectures of multiuser interactive systems. In *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 252–261, 2000.

[29] To T.-P.J., Koon-Hung Wong, and Chi-Kwong Li. Strategic selection and replication of movies by trend-calibrated movie-demand model. In *Multimedia Software Engineering, 2000. Proceedings. International Symposium on*, pages 97–100, 2000.

[30] Andrew Brooks, Fredrik Utbult, Catherine Mulligan, and Ross Jeffery. Early lifecycle work: Influence of individual characteristics, methodological constraints, and interface constraints. *Empirical Software Engineering*, 5(3):269–285, 2000.

[31] J. Putman. General framework for fault tolerance from iso/itu reference model for open distributed processing (rm-odp. In *Object-Oriented Real-Time Dependable Systems, 1999. WORDS 1999 Fall. Proceedings. Fifth International Workshop on*, pages 111–118, 2000.

[32] Anders Wesslén. A replicated empirical study of the impact of the methods in the psp on individual engineers. *Empirical Software Engineering*, 5(2):93–123, 2000.

[33] Karl Cox and Keith Phalp. Replicating the crews use case authoring guidelines experiment. *Empirical Software Engineering*, 5(3):245–267, 2000.

[34] Karamanolis C.T. and Magee J.N. Client-access protocols for replicated services. *IEEE Transactions on Software Engineering*, 25(1):3–21, 1999.

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

[36] Briand L.C., Wust J., Ikonomovski S.V., and Lounis H. Investigating quality factors in object-oriented designs: an industrial case study. In *Software Engineering, 1999. Proceedings of the 1999 International Conference on*, pages 345–354, 1999.

[37] T Christos and N Jeffrey. Client-access protocols for replicated services. *IEEE Transactions on Software Engineering*, 25(1):3–21, 1999.

[38] Philip M. Johnson and Danu Tjahjono. Does every inspection really need a meeting. *Empirical Software Engineering*, 3(1):9–35, 1998.

[39] San-Yih Hwang and Chi-Ten Yang. Component and data distribution in a distributed workflow. In *Software Engineering Conference, 1998. Proceedings. 1998 Asia Pacific*, pages 244–251, 1998.

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

[41] Adam Porter and Lawrence Votta. Comparing detection methods for software requirements inspections: A replication using professional subjects. *Empirical Software Engineering*, 3(4):355–379, 1998.

[42] Kristian Sandahl, Ola Blomkvist, Joachim Karlsson, Christian Krysander, Mikael Lindvall, and Niclas Ohlsson. An extended replication of an experiment for assessing methods for software requirements inspections. *Empirical Software Engineering*, 3(4):327–354, 1998.

[43] M. Van Steen, S. Van der Zijden, and H.J. Sips. Software engineering for the scalable distributed applications. In *Computer Software and Applications Conference, 1998. COMPSAC '98. Proceedings. The Twenty-Second Annual International*, pages 285–292, 1998.

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

[45] Tai A.T. and Alkalai L. On-board maintenance for long-life systems. In *Application-Specific Software Engineering Technology, 1998. ASSET-98. Proceedings. 1998 IEEE Workshop on*, pages 69–74, 1998.

[46] Aleta Ricciardi, Michael Ogg, and Fabio Previato. Experience with distributed replicated objects: The nile project. *Theory and Practice of Object Systems*, 4(2):107–115, 1998.

[47] M. Roper, M. Wood, and J. Miller. An empirical evaluation of defect detection techniques. *Information and Software Technology*, 39(11):763–775, 1997.

[48] CHYE-LIN CHEE and SEVKI S. ERDOGAN. An installable version control file system for unix. *Software: Practice and Experience*, 27(6):725–746, 1997.

[49] Ogura N., Saisho K., and Fukuda A. Design of protocols in timed csp for highly reliable and available client-server system. In *Software Engineering Conference, 1997. Asia Pacific ... and International Computer Science Conference 1997. APSEC '97 and ICSC '97. Proceedings*, pages 495–502, 1997.

[50] James D. Kiper, Brent Auernheimer, and Charles K. Ames. Visual depiction of decision statements: What is best for programmers and non-programmers. *Empirical Software Engineering*, 2(4):361–379, 1997.

[51] Triantafillou P. and Neilson C. Achieving strong consistency in a distributed file system. *IEEE Transactions on Software Engineering*, 23(1):35–55, 1997.

[52] Triantafillou Peter and Neilson Carl. Achieving strong consistency in a distributed file system. *IEEE Transactions on Software Engineering*, 23(1):35–55, 1997.

[53] Pierfrancesco Fusaro, Filippo Lanubile, and Giuseppe Visaggio. A replicated experiment to assess requirements inspection techniques. *Empirical Software Engineering*, 2(1):39–57, 1997.

[54] Wikstrom C. Implementing distributed real-time control systems in a functional programming language. *Parallel and Distributed Real-Time Systems, 1996. Proceedings of the 4th International Workshop on*, pages 20–26, 1996.

[55] John Daly, Andrew Brooks, James Miller, Marc Roper, and Murray Wood. Evaluating inheritance depth on the maintainability of object-oriented software. *Empirical Software Engineering*, 1(2):109–132, 1996.

[56] Triantafiliou P. Independent recovery in large-scale distributed systems. *IEEE Transactions on Software Engineering*, 22(11):812–826, 1996.

[57] Triantafillou Peter. Independent recovery in large-scale distributed systems. *IEEE Transactions on Software Engineering*, 22(11):812–826, 1996.

[58] Silva A.R., Sousa P., and Marques J.A. Development of distributed applications with separation of concerns. In *Software Engineering Conference, 1995. Proceedings., 1995 Asia Pacific*, pages 168–177, 1995.

[59] Triantafillou Peter and Taylor David J. The location-based paradigm for replication: Achieving efficiency and availability in distributed systems. *IEEE Transactions on Software Engineering*, 21(1):1–19, 1995.

[60] Triantafillou P. and Taylor D.J. The location-based paradigm for replication: Achieving efficiency and availability in distributed systems. *IEEE Transactions on Software Engineering*, 21(1):1–18, 1995.

[61] Sumin Huang. Developing distributed applications by semantics-based automatic replication. In *Software Engineering Conference, 1994. Proceedings., 1994 First Asia-Pacific*, pages 40–49, 1994.

[62] Kiskis D.L. and Shin K.G. Swsl: a synthetic workload specification language for real-time systems. *IEEE Transactions on Software Engineering*, 20(10):798–811, 1994.

[63] Daly J., Brooks A., Miller J., Roper M., and Wood M. Verification of results in software maintenance through external replication. In *Software Maintenance, 1994. Proceedings., International Conference on*, pages 50–57, 1994.

[64] Kiskis Daniel L and Shin Kang G. Swsl: A synthetic workload specification language for real-time systems. *IEEE Transactions on Software Engineering*, 20(10):798–812, 1994.

[65] Cox John. Oracle updates database development tools. *CommunicationsWeek*, page 12, 1994.

[66] Rangarajan Sampath, Jalote Pankaj, and Tripathi Satish K. Capacity of voting systems. *IEEE Transactions on Software Engineering*, 19(7):698–707, 1993.

[67] Rangarajan S., Jalote P., and Tripathi S.K. Capacity of voting systems. *IEEE Transactions on Software Engineering*, 19(7):698–706, 1993.

[68] Payne A. Designing the databases of the intelligent network. In *Software Engineering for Telecommunication Systems and Services, 1992., Eighth International Conference on*, pages 37–41, 1992.

[69] Eichmann D. Supporting multiple domains in a single reuse repository. In *Software Engineering and Knowledge Engineering, 1992. Proceedings., Fourth International Conference on*, pages 164–169, 1992.

[70] *Proceedings of the 12th International Conference on Distributed Computing Systems (Cat No.92CH3175-7)*. 1992.

[71] Ciciani Bruno, Dias Daniel M., and Yu Philip S. Analysis of concurrency-coherency control protocols for distributed transaction processing systems with regional locality. *IEEE Transactions on Software Engineering*, 18(10):899–915, 1992.

[72] Ammarguellat Zahira. A control-flow normalization algorithm and its complexity. *IEEE Transactions on Software Engineering*, 18(3):237–252, 1992.

[73] Ciciani B., Dias D.M., and Yu P.S. Analysis of concurrency-coherency control protocols for distributed transaction processing systems with regional locality. *IEEE Transactions on Software Engineering*, 18(10):899–914, 1992.

[74] Ammarguellat Z. A control-flow normalization algorithm and its complexity. *IEEE Transactions on Software Engineering*, 18(3):237–251, 1992.

[75] Shin Dong-Guk and Irani Keki B. Fragmenting relations horizontally using a knowledge-based approach. *IEEE Transactions on Software Engineering*, 17(9):872–884, 1991.

[76] Gehani N.H. Concurrent c: real-time programming and fault tolerance. *Software Engineering Journal*, 6(3):83–92, 1991.

[77] O'Donovan B. and Grimson J.B. A distributed version control system for wide area networks. *Software Engineering Journal*, 5(5):255–262, 1990.

[78] Hanna Mary Alice. Defining the 'r' words for automated maintenance. *Software Magazine*, 10(6):41–47, 1990.

[79] Levi S.-T., Tripathi S.K., Carson S.D., and Agrawala A.K. The maruti hard real-time operating system. In *Computer Systems and Software Engineering, 1989. Proceedings., Fourth Israel Conference on*, pages 5–15, 1989.

[80] Wojcik B.E. and Wojcik Z.M. Sufficient condition for a communication deadlock and distributed deadlock detection. *IEEE Transactions on Software Engineering*, 15(12):1587–1595, 1989.

[81] Wojcik Barbara E. and Wojcik Zbigniew M. Sufficient condition for a communication deadlock and distributed deadlock detection. *IEEE Transactions on Software Engineering*, 15(12):1587–1596, 1989.

[82] Hac A. A distributed algorithm for performance improvement through file replication, file migration, and process migration. *IEEE Transactions on Software Engineering*, 15(11):1459–1470, 1989.

[83] Ahamad Mustaque and Ammar Mostafa H. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–497, 1989.

[84] Ahamad M. and Ammar M.H. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–496, 1989.

[85] Hac Anna. A distributed algorithm for performance improvement through file replication, file migration, and process migration. *IEEE Transactions on Software Engineering*, 15(11):1459–1471, 1989.

[86] Sang Hyuk Son. Semantic information and consistency in distributed realtime systems. *Information and Software Technology*, 30(7):443–450, 1988.

[87] Pu Calton, Noe Jerre D., and Proudfoot Andrew. Regeneration of replicated objects: A technique and its eden implementation. *IEEE Transactions on Software Engineering*, 14(7):936–946, 1988.

[88] Yu Chee-Fen and Wah Benjamin W. Learning dominance relations in combinatorial search problems. *IEEE Transactions on Software Engineering*, 14(8):1155–1176, 1988.

[89] Pu C., Noe J.D., and Proudfoot A. Regeneration of replicated objects: a technique and its eden implementation. *IEEE Transactions on Software Engineering*, 14(7):936–945, 1988.

[90] Yu C.-F. and Wah B.W. Learning dominance relations in combined search problems. *IEEE Transactions on Software Engineering*, 14(8):1155–1175, 1988.

[91] Andrews Gregory R., Schlichting Richard D., Hayes Roger, and Purdin Titus D. M. The design of the saguaro distributed operating system. *IEEE Transactions on Software Engineering*, SE13(1):104–119, 1987.

Search results for "software engineering" and replicated

[1] Chen-Liang Fang, Deron Liang, and Fengyi Lin. A nested invocation suppression mechanism for cctive replicated soap systems. *Software Engineering Conference, 2004. 11th Asia-Pacific*, pages 392–399, 2004.

[2] J. Munch and O. Armbrust. Using empirical knowledge from replicated experiments for software process simulation: a practical example. *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 18–27, 2003.

[3] E. Mendes, N. Mosley, and S. Counsell. A replicated assessment of the use of adaptation rules to improve web cost estimation. *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 100–109, 2003.

[4] Bobbio Andrea, Franceschinis Giuliana, Gaeta Rossano, and Portinale Luigi. Parametric fault tree for the dependenability analysis of redundant systems and its high-level petri net semantics. *IEEE Transactions on Software Engineering*, 29(3):270–287, 2003.

[5] A. Bobbio, G. Franceschinis, R. Gaeta, and L. Portinale. Parametric fault tree for the dependability analysis of redundant systems and its high-level petri net semantics. *Software Engineering, IEEE Transactions on*, 29(3):270–287, 2003.

[6] Dietmar Pfahl, Oliver Laitenberger, Jorg Dorsch, and Günther Ruhe. An externally replicated experiment for evaluating the learning effectiveness of using simulations in software project management education. *Empirical Software Engineering*, 8(4):367–395, 2003.

[7] J. Diaz-Herrera, M. Murphy, and D. Ramsey. A collaborative program to retrain lockheed martin aero engineers. *IEEE Software*, 19(5):30–34, 2002.

[8] Christof Ebert and Jozef De Man. e-r&d - effectively managing process diversity. *Annals of Software Engineering*, 14(1-4):73–91, 2002.

[9] Deron Liang, Chen-Liang Fang, Chyouhwa Chen, and Fengyi Lin. A nested invocation suppression mechanism for active replication fault-tolerant corba. *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pages 117–125, 2002.

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

[11] N. Kaveh. Model checking distributed objects design. *Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on*, pages 793–794, 2001.

[12] Lionel C. Briand, Jürgen Wüst, and Hakim Lounis. Replicated case studies for investigating quality factors in object-oriented designs. *Empirical Software Engineering*, 6(1):11–58, 2001.

[13] Laitenberger Oliver, Emam Khaled El, and Harbich Thomas G. An internally replicated quasi-experimental comparison of checklist and perspective-based reading of code documents. *IEEE Transactions on Software Engineering*, 27(5):387–421, 2001.

[14] Briand L.C., Langley T., and Wieczorek I. A replicated assessment and comparison of common software cost modeling techniques. *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 377–386, 2000.

[15] Anders Wesslén. A replicated empirical study of the impact of the methods in the psp on individual engineers. *Empirical Software Engineering*, 5(2):93–123, 2000.

[16] Jayawardana C., Hewagamage K.P., and Hirakawa M. Virtual authoring based on the shallow copy technique for a collection of digital documents. *Multimedia Software Engineering, 2000. Proceedings. International Symposium on*, pages 77–84, 2000.

[17] Coppit D. and Sullivan K.J. Multiple mass-market applications as components. *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 273–282, 2000.

[18] Chandra S., Richards B., and Larus J.R. Teapot: a domain-specific language for writing cache coherence protocols. *Software Engineering, IEEE Transactions on*, 25(3):317–333, 1999.

[19] Ing-Ray Chen, Ding-Chau Wang, and Chih-Ping Chu. User-perceived availability and response-time in voting-based replicated systems: a case study. *Application-Specific Systems and Software Engineering and Technology, 1999. ASSET '99. Proceedings. 1999 IEEE Symposium on*, pages 103–110, 1999.

[20] Karamanolis C.T. and Magee J.N. Client-access protocols for replicated services. *Software Engineering, IEEE Transactions on*, 25(1):3–21, 1999.

[21] Finney K., Fenton N., and Fedorec A. Effects of structure on the comprehensibility of formal specifications. *Software, IEE Proceedings-*, 146(4):193–202, 1999.

[22] Briand L.C., Wust J., Ikonomovski S.V., and Lounis H. Investigating quality factors in object-oriented designs: an industrial case study. *Software Engineering, 1999. Proceedings of the 1999 International Conference on*, pages 345–354, 1999.

[23] R James. Teapot: A domain-specific language for writing cache coherence protocols. *IEEE Transactions on Software Engineering*, 25(3):317–333, 1999.

[24] T Christos and N Jeffrey. Client-access protocols for replicated services. *IEEE Transactions on Software Engineering*, 25(1):3–21, 1999.

[25] Adam Porter and Lawrence Votta. Comparing detection methods for software requirements inspections: A replication using professional subjects. *Empirical Software Engineering*, 3(4):355–379, 1998.

[26] Aleta Ricciardi, Michael Ogg, and Fabio Previato. Experience with distributed replicated objects: The nile project. *Theory and Practice of Object Systems*, 4(2):107–115, 1998.

[27] Kristian Sandahl, Ola Blomkvist, Joachim Karlsson, Christian Krysander, Mikael Lindvall, and Niclas Ohlsson. An extended replication of an experiment for assessing methods for software requirements inspections. *Empirical Software Engineering*, 3(4):327–354, 1998.

[28] Stephen G. Eick, Audris Mockus, Todd L. Graves, and Alan F. Karr. A web laboratory for software data analysis. *World Wide Web*, 1(2):55–60, 1998.

[29] Pek Wee Land L., Jeffery R., and Sauer C. Validating the defect detection performance advantage of group designs for software reviews: report of a replicated experiment. *Software Engineering Conference, 1997. Proceedings., Australian*, pages 17–26, 1997.

[30] Triantafillou Peter and Neilson Carl. Achieving strong consistency in a distributed file system. *IEEE Transactions on Software Engineering*, 23(1):35–55, 1997.

[31] Triantafillou P. and Neilson C. Achieving strong consistency in a distributed file system. *Software Engineering, IEEE Transactions on*, 23(1):35–55, 1997.

[32] Hilderman Robert J and Hamilton Howard J. A note on regeneration with virtual copies. *IEEE Transactions on Software Engineering*, 23(1):56–59, 1997.

[33] Pierfrancesco Fusaro, Filippo Lanubile, and Giuseppe Visaggio. A replicated experiment to assess requirements inspection techniques. *Empirical Software Engineering*, 2(1):39–57, 1997.

[34] Hilderman R.J. and Hamilton H.J. A note on regeneration with virtual copies. *Software Engineering, IEEE Transactions on*, 23(1):56–59, 1997.

[35] David L. Coleman and Albert L. Baker. Synthesizing structured analysis and object-based formal specifications. *Annals of Software Engineering*, pages 221–253, 1997 Volym: 3.

[36] Triantafillou Peter. Independent recovery in large-scale distributed systems. *IEEE Transactions on Software Engineering*, 22(11):812–826, 1996.

[37] Devanbu P., Karstu S., Melo W., and Thomas W. Analytical and empirical evaluation of software reuse metrics. *Software Engineering, 1996., Proceedings of the 18th International Conference on*, pages 189–199, 1996.

[38] John Daly, Andrew Brooks, James Miller, Marc Roper, and Murray Wood. Evaluating inheritance depth on the maintainability of object-oriented software. *Empirical Software Engineering*, 1(2):109–132, 1996.

[39] Triantafiliou P. Independent recovery in large-scale distributed systems. *Software Engineering, IEEE Transactions on*, 22(11):812–826, 1996.

[40] Porter Adam A, Votta Lawrence G Jr, and Basili Victor R. Comparing detection methods for software requirements inspections: A replicated experiment. *IEEE Transactions on Software Engineering*, 21(6):563–576, 1995.

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

[42] Sumin Huang. Developing distributed applications by semantics-based automatic replication. *Software Engineering Conference, 1994. Proceedings., 1994 First Asia-Pacific*, pages 40–49, 1994.

[43] Daly J., Brooks A., Miller J., Roper M., and Wood M. Verification of results in software maintenance through external replication. *Software Maintenance, 1994. Proceedings., International Conference on*, pages 50–57, 1994.

[44] Rushby J.M. and von Henke F. Formal verification of algorithms for critical systems. *Software Engineering, IEEE Transactions on*, 19(1):13–23, 1993.

[45] Rangarajan Sampath, Jalote Pankaj, and Tripathi Satish K. Capacity of voting systems. *IEEE Transactions on Software Engineering*, 19(7):698–707, 1993.

[46] Adam Nabil R and Tewari Rajiv. Regeneration with virtual copies for distributed computing systems. *IEEE Transactions on Software Engineering*, 19(6):594–603, 1993.

[47] Rushby John M and von Henke Friedrich. Formal verification of algorithms for critical systems. *IEEE Transactions on Software Engineering*, 19(1):13–24, 1993.

[48] Rangarajan S., Jalote P., and Tripathi S.K. Capacity of voting systems. *Software Engineering, IEEE Transactions on*, 19(7):698–706, 1993.

[49] Adam N.R. and Tewari R. Regeneration with virtual copies for distributed computing systems. *Software Engineering, IEEE Transactions on*, 19(6):594–602, 1993.

[50] Hu P. and Wilbur S. Low storage cost, partition-tolerant dynamic algorithms for replicated file systems. *CompEuro '92 . 'Computer Systems and Software Engineering',Proceedings.*, pages 89–94, 1992.

[51] Ciciani B., Dias D.M., and Yu P.S. Analysis of concurrency-coherency control protocols for distributed transaction processing systems with regional locality. *Software Engineering, IEEE Transactions on*, 18(10):899–914, 1992.

[52] Shin Dong-Guk and Irani Keki B. Fragmenting relations horizontally using a knowledge-based approach. *IEEE Transactions on Software Engineering*, 17(9):872–884, 1991.

[53] Gehani N.H. Concurrent c: real-time programming and fault tolerance. *Software Engineering Journal*, 6(3):83–92, 1991.

[54] Singhal Mukesh. Update transport: A new technique for update synchronization in replicated database systems. *IEEE Transactions on Software Engineering*, 16(12):1325–1337, 1990.

[55] Singhal M. Update transport: a new technique for update synchronization in replicated database systems. *Software Engineering, IEEE Transactions on*, 16(12):1325–1336, 1990.

[56] Davcev Danco. A dynamic voting scheme in distributed systems. *IEEE Transactions on Software Engineering*, 15(1):93–98, 1989.

[57] Bhargava B. and Riedl J. The raid distributed database system. *Software Engineering, IEEE Transactions on*, 15(6):726–736, 1989.

[58] Davcev D. A dynamic voting scheme in distributed systems. *Software Engineering, IEEE Transactions on*, 15(1):93–97, 1989.

[59] Jajodia S. and Mutchler D. A pessimistic consistency control algorithm for replicated files which achieves high availability. *Software Engineering, IEEE Transactions on*, 15(1):39–46, 1989.

[60] Ahamad M. and Ammar M.H. Performance characterization of quorum-consensus algorithms for replicated data. *Software Engineering, IEEE Transactions on*, 15(4):492–496, 1989.

[61] Bechta Dugan J. and Ciardo G. Stochastic petri net analysis of a replicated file system. *Software Engineering, IEEE Transactions on*, 15(4):394–401, 1989.

[62] Dugan Joanne Bechta and Ciardo Gianfranco. Stochastic petri net analysis of a replicated file system. *IEEE Transactions on Software Engineering*, 15(4):394–402, 1989.

[63] Ahamad Mustaque and Ammar Mostafa H. Performance characterization of quorum-consensus algorithms for replicated data. *IEEE Transactions on Software Engineering*, 15(4):492–497, 1989.

[64] Rochlin G. An information model for intelligent network services. *Software Engineering for Telecommunication Switching Systems, 1989. SETSS 89., Seventh International Conference on*, pages 147–153, 1989.

[65] Jajodia Sushil and Mutchler David. A pessimistic consistency control algorithm for replicated files which achieves high availability. *IEEE Transactions on Software Engineering*, 15(1):39–47, 1989.

[66] Bhargava Bharat and Riedl John. The raid distributed database system. *IEEE Transactions on Software Engineering*, 15(6):726–737, 1989.

[67] Pu C., Noe J.D., and Proudfoot A. Regeneration of replicated objects: a technique and its eden implementation. *Software Engineering, IEEE Transactions on*, 14(7):936–945, 1988.

[68] Mukherjee A., Kramer J., and Magee J. A distributed file server for embedded applications. *Software Engineering Journal*, 3(5):142–148, 1988.

[69] Sang Hyuk Son. Semantic information and consistency in distributed realtime systems. *Information and Software Technology*, 30(7):443–450, 1988.

[70] Pu Calton, Noe Jerre D., and Proudfoot Andrew. Regeneration of replicated objects: A technique and its eden implementation. *IEEE Transactions on Software Engineering*, 14(7):936–946, 1988.

[71] Fuchs W. Kent, Wu Kun-Lung, and Abraham Jacob A. Comparison and diagnosis of large replicated files. *IEEE Transactions on Software Engineering*, SE13(1):15–23, 1987.

[72] Sarin Sunil K. and Lynch Nancy A. Discarding obsolete information in a replicated database system. *IEEE Transactions on Software Engineering*, SE13(1):39–48, 1987.

[73] Mancini Luigi. Modular redundancy in a message passing system. *IEEE Transactions on Software Engineering*, SE12(1):79–87, 1986.

[74] Ahamad Mustaque and Bernstein Arthur J. An application of name based addressing to low level distributed algorithms. *IEEE Transactions on Software Engineering*, SE11(1):59–68, 1985.

[75] Yu Clement T., Chang C. C., Templeton Marjorie, Brill David, and Lund Eric. Query processing in a fragmented relational distributed system: Mermaid. *IEEE Transactions on Software Engineering*, SE11(8):795–811, 1985.

[76] Manber Udi. Concurrent maintenance of binary search trees. *IEEE Transactions on Software Engineering*, SE10(6):777–785, 1984.

Search results for "software engineering" and replica

[1] R.Y. Yen and K.W.K. Tsai. Identification of a noiseless nonlinear system by gradient method. In *Multimedia Software Engineering, 2000. Proceedings. International Symposium on*, pages 259–262, 2000.

[2] C.T. Karamanolis and J.N. Magee. Client-access protocols for replicated services. *IEEE Transactions on Software Engineering*, 25(1):3–21, 1999.

[3] P. Mellor. A model of the problem or a problem with the model. *Computing & Control Engineering Journal*, 9(1):8–18, 1998.

[4] Peter Triantafillou and David J. Taylor. The location-based paradigm for replication: Achieving efficiency and availability in distributed systems. *IEEE Transactions on Software Engineering*, 21(1):1–19, 1995.

[5] P. Triantafillou and D. J. Taylor. The location-based paradigm for replication: Achieving efficiency and availability in distributed systems. *Software Engineering, IEEE Transactions on*, 21(1):1–18, 1995.

[6] C. Pu, J.D. Noe, and A. Proudfoot. Regeneration of replicated objects: a technique and its eden implementation. *Software Engineering, IEEE Transactions on*, 14(7):936–945, 1988.

Search results for "software engineering" and replicating

[1] J. Miller. Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information and Software Technology*, 47(4):233–244, 2005.

[2] James Miller. Replicating software engineering experiments: a poisoned chalice or the holy grail. *Information and Software Technology*, 47(4):233–244, 2005.

[3] Zeiad Abdelnabi, G. Cantone, M. Ciolkowski, and D. Rombach. Comparing code reading techniques applied to object-oriented software frameworks with regard to effectiveness and defect detection rate. In *Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on*, pages 239–248, 2004.

[4] Forrest Shull, Manoel G. Mendoncca, Victor Basili, Jeffrey Carver, José C. Maldonado, Sandra Fabbri, Guilherme Horta Travassos, and Maria Cristina Ferreira. Knowledge-sharing issues in experimental software engineering. *Empirical Software Engineering*, 9(1-2):111–137, 2004.

[5] A.A. Terekhov. Re-using software architecture in legacy transformation projects. In *Software Maintenance, 2003. ICSM 2003. Proceedings. International Conference on*, page 462, 2003.

[6] Shull F., Basili V., Carver J., Maldonado J.C., Travassos G.H., Mendonca M., and Fabbri S. Replicating software engineering experiments: addressing the tacit knowledge problem. In *Empirical Software Engineering, 2002. Proceedings. 2002 International Symposium n*, pages 7–16, 2002.

[7] Karl Cox and Keith Phalp. Replicating the crews use case authoring guidelines experiment. *Empirical Software Engineering*, 5(3):245–267, 2000.

[8] James D. Kiper, Brent Auernheimer, and Charles K. Ames. Visual depiction of decision statements: What is best for programmers and non-programmers. *Empirical Software Engineering*, 2(4):361–379, 1997.

Search results for "software engineering" and validity

[1] Giancarlo Succi, Witold Pedrycz, Snezana Djokic, Paolo Zuliani, and Barbara Russo. An empirical exploration of the distributions of the chidamber and kemerer object-oriented metrics suite. *Empirical Software Engineering*, 10(1):81–104, 2005.

[2] Yasuyuki Tsukada. Interactive and probabilistic proof of mobile code safety. *Automated Software Engineering*, 12(2):237–257, 2005.

[3] Taghi M. Khoshgoftaar, Naeem Seliya, and Kehan Gao. Assessment of a new three-group software quality classification technique: An empirical case study. *Empirical Software Engineering*, 10(2):183–218, 2005.

[4] N. Nagappan. Toward a software testing and reliability early warning metric suite. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, pages 60–62, 2004.

[5] J. Carver, J. VanVoorhis, and V. Basili. Understanding the impact of assumptions on experimental validity. In *Empirical Software Engineering, 2004. ISESE '04. Proceedings. 2004 International Symposium on*, pages 251–260, 2004.

[6] K. Sen, A. Vardhan, G. Agha, and G Rosu. Efficient decentralized monitoring of safety in distributed systems. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, pages 418–427, 2004.

[7] B. George and L. Williams. A structured experiment of test-driven development. *Information and Software Technology*, 46(5):337–342, 2004.

[8] David King and Chris Kimble. Uncovering the epistemological and ontological assumptions of software designers. Technical report, arXiv, 2004.

[9] N. Oses, M. Pidd, and R.J. Brooks. Critical issues in the development of component-based discrete simulation. *Simulation Modelling Practice & Theory*, 12(7-8):495–514, 2004.

[10] Magne Jørgensen. Regression models of software development effort estimation accuracy and bias. *Empirical Software Engineering*, 9(4):297–314, 2004.

[11] L. Flores and J. Barata. Object oriented software engineering for programmable logical controllers a successful implementation. In *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, pages 116–120, 2003.

[12] Padmal Vitharana and K. Ramamurthy. Computer-mediated group support, anonymity, and the software inspection process: An empirical investigation. *IEEE Transactions on Software*, 29(2):167–180, 2003.

[13] G. Ruhe and D. Greer. Quantitative studies in software release planning under risk and resource constraints. In *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, page 26, 2003.

[14] D. L. Moody, G. Sindre, T. Brasethvik, and A. Solvberg. Evaluating the quality of information models: empirical testing of a conceptual model quality framework. In *Software Engineering, 2003. Proceedings. 25th International Conference on*, pages 295–305, 2003.

[15] Issues in using students in empirical studies in software engineering education. Carver, j. and jaccheri, l. and morasca, s. and shull, f. In *Software Metrics Symposium, 2003. Proceedings. Ninth International*, pages 239–249, 2003.

[16] P. Vitharana and K. Ramamurthy. Computer-mediated group support, anonymity, and the software inspection process: an empirical investigation. *Software Engineering, IEEE Transactions on*, 29(2):167–180, 2003.

[17] P. Cesar, J. Vierinen, and P Vuorimaa. Open graphical framework for interactive tv. In *Multimedia Software Engineering, 2003. Proceedings. Fifth International Symposium on*, pages 21–28, 2003.

[18] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time markov chains. *Software Engineering, IEEE Transactions on*, 29(6):524–541, 2003.

[19] W. M. Evanco. The confounding effect of class size on the validity of object-oriented metrics. *Software Engineering, IEEE Transactions on*, 29(7):670–673, 2003.

[20] W.M. Evanco. Comments on "the confounding effect of class size on the validity of object-oriented metrics". *Software Engineering, IEEE Transactions on*, 29(7):670–672, 2003.

[21] S. Liu. A rigorous approach to reviewing formal specifications. In *Software Engineering Workshop, 2002. Proceedings. 27th Annual NASA Goddard/IEEE*, pages 75–81, 2003.

[22] T. J. Ellis. Completing the cycle: meaningful course evaluations. In *Frontiers in Education, 2003. FIE 2003 33rd Annual*, volume 1, 2003.

[23] Christel Baier, Boudewijn Haverkort, Holger Hermanns, and Joost-Pieter Katoen. Model-checking algorithms for continuous-time markov chains. *IEEE Transactions on Software Engineering*, 29(6):524, 2003.

[24] Lars Bratthall and Magne Jørgensen. Can you trust a single data source exploratory software engineering case study? *Empirical Software Engineering*, 7(1):9–26, 2002.

[25] T. Katayama. Proposal of a supporting method for diagram generation with the transformation rules in uml. In *Software Engineering Conference, 2002. Ninth Asia-Pacific*, pages 475–484, 2002.

[26] G. Denaro and M. Pezze. An empirical evaluation of fault-proneness models. In *Software Engineering, 2002. ICSE 2002. Proceedings of the 24rd International Conference o*, pages 241–251, 2002.

[27] Hanks K. S., Knight J. C., and Strunk E. A. Erroneous requirements: a linguistic basis for their occurrence and an approach to their reduction. In *Software Engineering Workshop, 2001. Proceedings. 26th Annual NASA Goddard*, pages 115–119, 2002.

[28] The confounding effect of class size on the validity of object-oriented metrics. El emam, khaled and benlarbi, saida and goel, nishith and rai, shesh n. *IEEE Transactions on Software Engineering*, 27(7):630–650, 2001.

[29] P. Abrahamsson. Commitment development in software process improvement: critical misconceptions. In *Software Engineering, 2001. ICSE 2001. Proceedings of the 23rd International Conference on*, pages 71–80, 2001.

[30] K. El Emam, S. Benlarbi, N. Goel, and S.N. Rai. The confounding effect of class size on the validity of object-oriented metrics. *Software Engineering, IEEE Transactions on*, 27(7):630–650, 2001.

[31] Ho-Won Jung, Robin Hunter, Dennis R. Goldenson, and Khaled El-Emam. Findings from phase 2 of the spice trials. *Software Process: Improvement and Practice*, 6(4):205–242, 2001.

[32] J. Dimitrov. Operational semantics for verilog. In *Software Engineering Conference, 2001. APSEC 2001. Eighth Asia-Pacific*, pages 161–168, 2001.

[33] T. Moynihan. 'requirements-uncertainty': should it be a latent, aggregate or profile construct? In *Software Engineering Conference, 2000. Proceedings. 2000 Australian*, pages 181–188, 2000.

[34] Martin Höst, Björn Regnell, and Claes Wohlin. Using students as subjects-a comparative study of students and professionals in lead-time impact assessment. *Empirical Software Engineering*, 5(3):201–214, 2000.

[35] P. M. S. Bueno and M. Jino. Identification of potentially infeasible program paths by monitoring the search for test data. In *Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on*, pages 209–218, 2000.

[36] V. Lotz, V. Kessler, and G.H. Walter. A formal security model for microprocessor hardware. *Software Engineering, IEEE Transactions on*, 26(8):702–712, 2000.

[37] K. El Emam and A Birk. Validating the iso/iec 15504 measure of software requirements analysis process capability. *Software Engineering, IEEE Transactions on*, 26(6):541–566, 2000.

[38] Ho-Won Jung, Marjan Pivka, and Jong-Yoon Kim. An empirical study of complexity metrics in cobol programs. *The Journal of Systems and Software*, 51(2):119–149, 2000.

[39] K. El Emam and A. Birk. Validating the iso/iec 15504 measures of software development process capability. *Journal of Systems and Software*, 51(2):119–149, 2000.

[40] Liu Shaoying, T. Fukuzaki, and K. Miyamoto. A gui and testing tool for sofl. In *Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific*, pages 421–425, 2000.

Search results for "software engineering" and replicates

[1] E. Mendes, N. Mosley, and S. Counsell. A replicated assessment of the use of adaptation rules to improve web cost estimation. *Empirical Software Engineering, 2003. ISESE 2003. Proceedings. 2003 International Symposium on*, pages 100–109, 2003.

[2] Briand L.C., Langley T., and Wieczorek I. A replicated assessment and comparison of common software cost modeling techniques. In *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 377–386, 2000.

Appendix C

Laboratory packages on the Web

Basili *et al.* [r3] has some interesting pointers on what a lab package should contain.

Series B

I wasn't able to find it...

Series C

`http://www.cs.umd.edu/projects/SoftEng/ESEG/manual/pbr_package/manual.html`

Series F

`http://serg.telecom.lth.se/research/packages/`

Series H

`http://www.ecs.soton.ac.uk/~rnv95r/` but seems to have moved.

Series J

I wasn't able to find it...

Series M

`http://wwwipd.ira.uka.de/EIR/`

Series R

`http://alarcos.inf-cr.uclm.es/english/asp/labpackage.asp`

Appendix D

Simula Research Laboratory

The Simula Research Laboratory conducts basic research in the fields of communication technology, scientific computing and software engineering. Our aim is to carry out research of the highest quality, educate graduate university students and create new businesses. The research will focus on fundamental scientific problems with a large potential for important applications in society. Education will be delivered in partnership with the universities in Norway. Simula will actively support, and create the conditions necessary for, the establishment of businesses based on the research it conducts.

Simula was established in 2001 and is located at Fornebu. The Norwegian Government funds Simula through a contract with The Research Council of Norway. In addition, Simula seeks co-operation with industry in order to provide solutions, increase the relevance of the research, and in order to strengthen the funding of basic research. It is a Simula policy to avoid short-term projects. Consulting and technologically oriented projects should preferably be organised as stand alone companies or conducted in co-operation with other partners.

The Simula Research Laboratory was evaluated in the fall of 2004. Five internationally renowned professors assigned by The Research Council of Norway conducted the evaluation. Some of their findings were:

1. The Evaluation Committee is impressed with the progress and level of activity achieved at the Simula Research Laboratory in the comparatively short time since its foundation. The organisation has succeeded in generating a vibrant research culture and is now operating as a highly effective research unit with growing international recognition.
2. The Simula Research Laboratory offers a unique environment that emphasises and promotes basic research while still covering the broader landscape from postgraduate teaching to commercialisation. The organisational and funding framework allows basic research to take centre stage, without any domination by constraints from pursuit of external funding typically found in industrial research institutes, or from the heavier teaching commitments found in the Universities.
3. The Evaluation Committee recommends that the Simula Research Laboratory be funded for the next 5 years. Furthermore, to ensure long- term continuity, the Evaluation Committee recommends that the Simula Research Laboratory be placed on a rolling 5+5 year contractual basis.

The Software Engineering department

The vision of the SE department is to be an international leader in understanding software engineering technologies regarding their impact on human, organisational and technological dimensions of systems development. Acquiring a deep understanding requires proposing and validating theories on the basis of experiments and other empirical studies, primarily conducted in software development organisations. The motivation for the research conducted in the SE department is to support the private and public IT industries in developing better IT systems using fewer resources. Hence, technology and knowledge transfer is an important part of the strategy of SE, and is provided through media, teaching, courses in industry and consultancy through Simula Innovation.

COLOPHON

Typeset in the Palatino type family using LaTeX (pdfTeX) and the `memoir` class on an Apple iBook G4. The bibliography was managed using `BibDesk` and typeset using BibTeX. `perl`, `make` and `cvs` were invaluable tools.

Printed at the Department of Computer Science, Lund university, Sweden.

$Id: main.tex,v 1.170 2006/03/15 07:11:45 johan Exp $
make target print-final-norefs